

Received November 30, 2021, accepted December 29, 2021, date of publication December 30, 2021, date of current version January 7, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3139850

Detection of Non-Stationary GW Signals in High Noise From Cohen's Class of Time-Frequency Representations Using Deep Learning

NIKOLA LOPAC^{1,2}, FRANKO HRŽIĆ^{2,3}, IRENA PETRIJEVČANIN VUKSANOVIĆ^{2,4}, AND JONATAN LERGA^{1,2,3}

¹Faculty of Maritime Studies, University of Rijeka, 51000 Rijeka, Croatia

²Center for Artificial Intelligence and Cybersecurity, University of Rijeka, 51000 Rijeka, Croatia

³Faculty of Engineering, University of Rijeka, 51000 Rijeka, Croatia

⁴Ministry of the Interior, 10000 Zagreb, Croatia

Corresponding author: Jonatan Lerga (jlerga@riteh.hr)

This work was supported in part by the Croatian Science Foundation under Project IP-2018-01-3739, in part by the Istraživanje, razvoj i inovacije (IRI2) Project "ABsistemDCiCloud" (KK.01.2.1.02.0179), in part by the University of Rijeka under Project uniri-tehnic-18-17 and Project uniri-tehnic-18-15, and in part by the European Cooperation in Science and Technology (COST) Project CA17137.

ABSTRACT Analysis of non-stationary signals in a noisy environment is a challenging research topic in many fields often requiring simultaneous signal decomposition in the time and frequency domain. This paper proposes a method for the classification of noisy non-stationary time-series signals based on Cohen's class of their time-frequency representations (TFRs) and deep learning algorithms. We demonstrated the proposed approach on the example of detecting gravitational-wave (GW) signals in intensive real-life, non-stationary, non-white, and non-Gaussian noise. For this purpose, we prepared a dataset based on the actual data from the Laser Interferometer Gravitational-Wave Observatory (LIGO) detector and the synthetic GW signals obtained by realistic simulations. Next, 12 different TFRs from Cohen's class were calculated from the original noisy time-series data and used to train three state-of-the-art convolutional neural network (CNN) architectures: ResNet-101, Xception, and EfficientNet. The obtained classification results are compared to those achieved by the base model trained on the original time series. Analysis of the results suggests that the proposed approach combining deep CNN architectures with Cohen's class TFRs yields high values of performance metrics and significantly improves the classification performance compared to the base model. The TFR-CNN models achieve the values of the classification accuracy of up to 97.10%, the area under the receiver operating characteristic curve (ROC AUC) of up to 0.9885, the recall of up to 95.87%, the precision of up to 99.51%, the F1 score of up to 97.03%, and the area under the precision-recall curve (PR AUC) of up to 0.9920. This classification performance is obtained on the dataset in which the signal-to-noise ratio (SNR) values of the raw, noisy time-series signals range from -123.46 to -2.27 dB. Therefore, this study suggests that using alternative TFRs of Cohen's class can improve the deep learning-based detection of non-stationary GW signals in an intensive noise environment. Moreover, the proposed approach can also be a viable solution for deep learning-based analysis of numerous other noisy non-stationary signals in different practical applications.

INDEX TERMS Non-stationary signals, noisy signals, time-frequency signal analysis, convolutional neural networks, deep learning, gravitational waves.

I. INTRODUCTION

Non-stationary signals are characterized by a time-varying frequency spectrum. In addition to raw time-series data,

The associate editor coordinating the review of this manuscript and approving it for publication was Siddhartha Bhattacharyya.

analysis of such signals is often performed on their Fourier transform spectrograms. Gravitational waves (GWs) are an example of non-stationary signals. Their discovery in 2015 [1] by the Advanced Laser Interferometer Gravitational-Wave Observatory (LIGO) detectors [2], followed by Nobel Prize in 2017, initiated intensive research in GW data

analysis. The low strain amplitudes of these astrophysical phenomena require high sensitivity measurements, which makes GW observations susceptible to the influence of various instrumental and environmental noise sources [3], [4], despite state-of-the-art measuring and noise reduction equipment. Thus, GW measurements are non-stationary and contain non-white, non-Gaussian, and non-stationary noise. Therefore, detecting such non-stationary signals in an intensive noisy environment is one of the most important research tasks in analyzing the ever-increasing amounts of GW data.

There are several approaches to GW detection, with specific algorithms being developed for different signal types. Compact binary coalescence (CBC) signals, including those from the binary black hole (BBH) and binary neutron star (BNS) mergers, are primarily detected by matched filtering [5], [6]. This technique consists of correlating the noisy strain data to a bank of waveform templates. However, this technique is impractical for detecting continuous and burst GWs, as their simulation models are very computationally demanding. Moreover, matched filtering can be considered as an optimal technique only for signals corrupted by Gaussian noise. Another approach involves denoising techniques that do not require information about the astrophysical properties of the underlying signals. These techniques, including the total-variation-based technique [7] and the technique based on the combination of the local polynomial approximation (LPA) and the relative intersection of confidence intervals (RICI) [8], have recently attracted increasing interest, with applications to denoising of BBH and core-collapse supernova (CCSN) signals. Although these techniques are efficient in noise reduction, they have to be coupled with other algorithm pipelines to enable GW events detection.

In parallel with emerging applications in other fields [9]–[13], machine learning (ML) has recently received increased attention in GW astronomy [14]. ML-based techniques have been used for GW data denoising [15]–[17], parameters estimation [18]–[20], and detector's glitches classification [21]–[23], as well as for GW detection. Although there have been some studies on the detection of other types of GW signals, including classifications based on the time-series strain data [24]–[27] and classifications based on the spectrograms [28]–[30], the research has primarily focused on the BBH signals due to the fact that these signals, in addition to BNS signals, are the only GW signals that the LIGO detectors have successfully observed up to date.

The studies on ML-based BBH signal detection have mainly focused on time-series data classification [31]–[33]. A deep learning-based approach was introduced to the GW data analysis research in [34] by using a one-dimensional (1D) deep convolutional neural network (CNN) for the detection of BBH signals embedded in simulated Gaussian noise. The approach was focused on distinguishing between time-series data containing noisy BBH signals and those containing only noise. The method was further extended in [35] and applied to the time-series data with real-life noise from the LIGO detectors. The authors in [36] report an upgrade

of the method proposed in [34] for use with a GW detector network. Another study on the application of the CNN for the detection of BBH signals in synthetic Gaussian noise using time-series data as input is presented in [37]. The authors in [38] provide an in-depth analysis of the CNN-based binary classification approach in the BBH time-series signal detection and extend its application to trigger generation in the real-life LIGO data.

However, the application of deep learning to two-dimensional (2D) transformations of BBH signals has not been adequately addressed in the literature to the best of our knowledge. Namely, we found only one recent paper reporting a preliminary study on the use of CNN-based classification of Q-transforms for the CBC [39]. More than that, after an extensive literature review, we have not found any approach on GW detection using deep learning and Cohen's class of time-frequency signal representations (TFRs).

Moreover, time series classification represents an important research topic in many other fields and practical applications. Various algorithms for time series classification using different approaches have been developed recently, including those based on shapelets. In [40], a Pruning Shapelets with Key Points (PSKP) method for shapelet discovery was proposed by extracting the shapelet candidates with the key points found based on the calculation of the standard deviation of the time series and then classifying the time series by a decision tree constructed based on the optimal shapelet. Furthermore, an Improved Early Distinctive Shapelet Classification (IEDSC) approach was presented in [41]. This approach included a method for measuring the similarity between time series using relative trend information, a pruning method for reducing the number of shapelets based on the starting positions of the shapelets with good quality, and a method for shapelet selection. In addition, a technique for time series classification based on multi-feature dictionary representation and ensemble learning was proposed in [42].

The conventional pattern recognition algorithms for time series classification are capable of achieving high accuracy. Nevertheless, the application of these algorithms is often time-consuming and requires hand-crafted heuristic features based on domain-specific knowledge. In contrast, deep learning algorithms enable automatic feature extraction while providing high accuracy and scalability. Several studies have recently considered the application of deep learning algorithms for time series classification [43], [44]. The applications include multi-layer perceptrons (MLPs) [45], deep CNNs [45], [46], graph-based extreme learning machine (G-ELM) [47], and long short-term memory fully convolutional networks (LSTM-FCNs) [48], [49].

On the other hand, the utilization of 2D signal transformations for deep learning-based classification of time-series signals is mainly limited to spectrogram representations [50]–[53]. However, the utilization of alternative TFRs from Cohen's class has not been adequately addressed in the literature.

Hence, in this paper, we investigate the potentials of the TFRs from Cohen's class to provide more information content than the original time series when they are used for the deep learning-based classification of noisy, non-stationary signals. To assess the impact of different TFRs on deep learning-based classification algorithms, we apply this approach to the problem of GW detection in intensive noise. First, we have prepared a dataset based on the real-life noise data from the LIGO detectors and physical simulations of GW signals from the BBH mergers, obtaining the signal-to-noise ratio (SNR) values of the raw time-series data in the range between -123.46 and -2.27 dB. Next, 12 different TFRs from Cohen's class were calculated for each time series and used as an input for training three state-of-the-art deep CNNs (ResNet-101, Xception, and EfficientNet). The obtained classification results are compared to those achieved by the base model using deep learning applied to the original time series, where the base model represents the state-of-the-art method for the deep learning-based GW detection proposed in [35]. The analysis of the results indicates that using Cohen's class TFRs provides high classification accuracy and significantly improves the GW detection performance compared to the base model.

The main contributions of this paper can be summarized as follows:

- We propose a method for the classification of noisy non-stationary signals based on their Cohen's class TFRs and state-of-the-art deep learning algorithms. To our best knowledge, based on the extensive literature review, existing research has primarily focused on the deep learning-based classification using time-series signals or only spectrograms as input (not quadratic TFRs from Cohen's class).
- We show that alternative TFRs from Cohen's class can be efficiently utilized for deep learning-based classification of noisy non-stationary signals, achieving improved classification performances.
- The proposed method is successfully applied to the challenging problem of detecting GW signals in intensive real-life, non-stationary, non-white, and non-Gaussian noise, using the dataset whose SNR values range from -123.46 to -2.27 dB. Moreover, the proposed method significantly outperforms the base model representing the state-of-the-art method for deep learning-based detection of GW signals using their original time-series form.
- In addition to the application in GW detection, the proposed method has great potentials to be implemented in practical applications in other fields dealing with the classification of noisy non-stationary signals, such as seismic, speech, EEG, and ECG signals, to name a few.

The rest of this paper is structured as follows. Section II describes the experimental setup used in this study, including data preparation, TFRs, and implementation of the deep learning classification algorithms. The obtained results are

presented and elaborated on in Section III. Finally, the concluding remarks are summarized in Section IV.

II. MATERIALS AND METHODS

A. EXPERIMENTAL SETUP

The experiment involves three main phases shown in Fig. 1: (1) data preparation, (2) training of deep learning models, and (3) model performance evaluation. The data preparation consists of retrieving the real-life noise data from the LIGO detectors and generating synthetic GW signals by performing extensive simulations that realistically model the BBH merger waveforms. Namely, the analyzed dataset consists of 100 000 time series (half containing GWs in intensive noise, and the other half just real-life LIGO noise). After preprocessing, we obtain the time-series input data and apply 12 different Cohen's class TFRs (resulting in 1.2 million time-frequency distributions). Next, we apply the deep learning techniques on TFRs to perform the classification, distinguishing the GW signals embedded in the background noise and those containing only the noise. Namely, we trained three distinct state-of-the-art deep CNN architectures (ResNet-101, Xception, and EfficientNet) on 1.2 million TFRs (12 different types of TFRs, thus obtaining a total of 36 different TFR-CNN combinations). We evaluate the performance of every TFR-CNN combination and compare it to that of the base model [35] trained on the original time-series data. Finally, we perform a series of statistical tests to verify the statistical significance of the obtained results.

B. DATA PREPARATION

Training and evaluation of deep learning models require extensive datasets representative of the underlying phenomena. To this end, we have collected GW data that realistically model the physics of these events. The data collection process involves two main steps: retrieving the actual LIGO recordings and generating the synthetic GW signals by simulations of the BBH merger waveforms. We have followed a similar approach to data generation as in [38], [54].

As background noise, we have employed the real-life data from the LIGO detectors. Using the actual LIGO recordings is a more realistic approach to modeling GW background data than the approach found in many studies on this topic using simulated Gaussian noise since background data, in that case, would not contain characteristic detector glitches. We have acquired the LIGO recordings from the second observing run (O2). The O2 ran from 30th of November 2016 to 25th of August 2017, and included the detection of GWs from seven BBH mergers and one BNS merger [55]. The O2 data [56] are publicly available from the Gravitational Wave Open Science Center (GWOSC) [57].

In this work, we constrained selected O2 data by several criteria imposed on the retrieved data segments. First, both the Hanford (H1) and Livingston (L1) LIGO detectors had to be operating and have data available during the considered time interval. In addition, the data must meet all specified

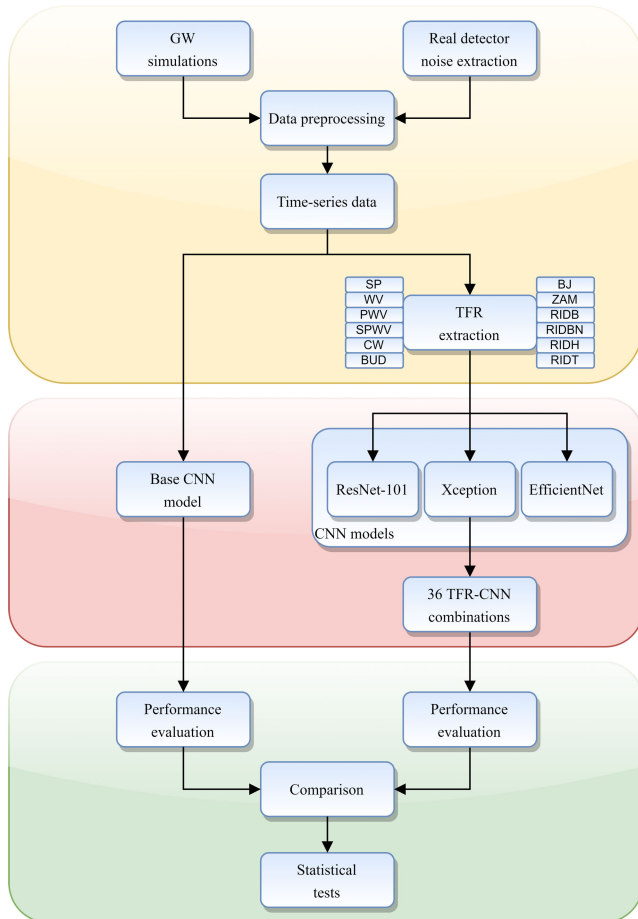


FIGURE 1. Flowchart of the main phases of the experiment.

quality requirements for CBC searches (minimum data quality level CBC-CAT3, as defined by the GWOSC). Finally, we excluded all data segments with the confirmed GW events and those containing hardware injections representing simulated signals injected into the LIGO measurements for testing purposes.

The data collection process begins by randomly choosing a GPS time between the start and end of the O2 run. We then check whether the symmetric time interval of 16 s around the chosen GPS time meets the defined criteria for the background data. If the selected data segment meets these criteria, it is extracted and downsampled from the original sampling frequency of 4096 Hz to 2048 Hz. Downsampling is performed to reduce the computational cost. The resulting sampling frequency of 2048 Hz allows the reconstruction of signals with frequencies up to 1024 Hz, as defined by the Nyquist-Shannon sampling theorem. Since signals from BBH mergers are generally expected in the range of up to a few hundred Hertz, the chosen Nyquist frequency of 1024 Hz is sufficient to detect them.

The whole process of data preparation is parallelized to reduce the execution time. Thus, in parallel with the extraction of the LIGO data, the simulations of the BBH waveforms are performed using the LIGO Algorithm

Library (LALSuite) [58], and PyCBC software package [59]. The waveform simulations are based on a state-of-the-art effective-one-body (EOB) model SEOBNRv4 in the time domain [60], suitable for simulating spinning non-precessing BBHs. The parameter values for the waveform simulations are independently and uniformly sampled at random from the defined parameter space. The parameters describe the waveform source (the masses and the z-components of the spins of the merging black holes), its position and orientation in the sky (polarization, right ascension, declination, coalescence phase angle, and inclination), and its distance, which is modeled by the injection SNR representing the desired network optimal matched-filter SNR (NOMF-SNR).

The masses of the two black holes are sampled from the range of 10 – 80 solar masses, while the z-components of their spins are chosen between 0 and 0.998. The polarization angle is taken from the range 0 – 2π. The right ascension and declination are sampled jointly from a uniform distribution over the sky. The coalescence phase and inclination are sampled together from a uniform distribution over a sphere. These angles define the location in the sky of the detector, as seen from the source reference frame. The desired injection SNR is randomly chosen between 8 and 30 dB since this range corresponds to the network matched-filter SNR (NMF-SNR) values of the real-life GW events observed so far in the GW detectors.

A one-sided Tukey window is applied to the simulated waveforms to suppress potential amplitude discontinuities. The simulated waveforms consist of two time-series signals h_+ and $h_×$, representing the + (plus) and the × (cross) tensor polarization mode of the GW, respectively. The h_+ and $h_×$ signals are then projected onto the antenna patterns of the LIGO detectors using the PyCBC functions. The obtained waveforms represent the noise-free detector signals, which are then injected into the selected background noise. The data examples containing only noise are obtained using only the background data, without the signal injections.

Prior to the injection, the simulated GW signals are scaled to get the desired injection SNR. The injection SNR is adjusted by calculating the matched-filter SNR (MF-SNR), used as a metric in all GW data analyses. The measured GW strain $s(t)$ is defined as:

$$s(t) = h_{GW}(t) + n(t), \quad (1)$$

where $h_{GW}(t)$ represents the GW signal and $n(t)$ the GW detector noise.

Thus, the matched-filter output $z(t)$ for the measured GW strain $s(t)$ and the matched-filter template $h_t(t)$ is calculated as [61]:

$$z(t) = 4 \int_0^\infty \frac{S(f)H_t^*(f)}{S_n(f)} e^{j2\pi ft} df, \quad (2)$$

where $S(f)$ represents the Fourier transform of $s(t)$, $H_t(f)$ the Fourier transform of $h_t(t)$, and $S_n(f)$ the estimated power spectral density (PSD) of the detector noise.

The MF-SNR $\rho(t)$ is defined as [61]:

$$\rho(t) = \frac{|z(t)|}{\sigma_{mf}}, \quad (3)$$

where σ_{mf} is the normalization constant defined for each template as [61]:

$$\sigma_{mf}^2 = 4 \int_0^\infty \frac{|H_t(f)|^2}{S_n(f)} df. \quad (4)$$

The NMF-SNR ρ_n is then calculated based on the MF-SNR values of each GW detector in the network [62]:

$$\rho_n = \sqrt{\sum_i \rho_i^2}. \quad (5)$$

When the time-inverted GW signal and the filter template are identical, the maximal MF-SNR, called the optimal matched-filter SNR (OMF-SNR), is obtained [62], [63]. The NOMF-SNR is then computed according to (5), based on the detectors' OMF-SNR values.

The injection SNR is adjusted in the data generation procedure by first adding together the simulated GW signal and the selected noise segment and then calculating the NOMF-SNR based on the OMF-SNR values obtained for both LIGO detectors. The ratio of the desired injection SNR and the calculated NOMF-SNR is then used to scale the GW signal. Thus scaled GW signal is injected into the noise, obtaining the data example with the desired NOMF-SNR. The 8 – 30 dB range of the desired NOMF-SNR resulted in the 0.10 – 30.46 dB range of the OMF-SNR for the data examples of the LIGO Livingston detector, which were used to generate the dataset further utilized in this study.

The generated data examples are then whitened by dividing their Fourier transforms by the local estimate of the detector noise amplitude spectral density (ASD), calculated as the square root of the PSD [64]. The PSD estimate is obtained by the Welch method [65] applied to the 16 s long data examples, with the Fourier transforms computed on the 4 s long overlapping windowed segments.

Next, the data examples are transformed back to the time domain and high-pass filtered at 20 Hz with the finite impulse response (FIR) filter. The high-pass filter is applied after whitening to remove the low-frequency artifacts possibly generated by simulations. In addition, LIGO data with frequencies below 20 Hz are not used in most studies because of the high noise level.

The whitening procedure involves the computation of the Fourier transform, which distorts the edges of the data examples at both ends. These edges are truncated to obtain data examples of 0.5 s length. This particular data example length was chosen to reduce the computational load of training the CNN models. Since BBH signals are usually of short duration, the most prominent part of the signal, i.e., the one representing coalescence, is expected to be contained within the chosen example length. The maximum of each BBH signal is also randomly positioned within the 0.1 – 0.4 s

interval of the data example to avoid possible overfitting of the deep learning models based on the positions of the signals.

The data generation procedure finally yielded 100 000 examples of length 0.5 s. Half of the examples contain GW signals embedded in the noise, while the other half contains only the noise. According to the standard SNR definition (the ratio of the signal power to the noise power), the 0.5 s long raw time-series data examples with injected GW signals, i.e., the time-series data examples before whitening and filtering, have the SNR values in the range from –123.46 to –2.27 dB.

We use the time-series data obtained by the above-described procedure directly for training and evaluating the base model and calculation of Cohen's class TFRs, which are then used as input to the deep CNN models.

C. COHEN'S CLASS TIME-FREQUENCY REPRESENTATIONS (TFRs)

1) TIME-FREQUENCY DISTRIBUTIONS

Real-life non-stationary signals, such as GWs, are often multi-component and corrupted by noise, which calls for advanced tools for their analysis beyond time-domain or Fourier frequency-domain analysis. Fourier transform-based spectral analysis provides information on the frequency components contained in the signal. However, the Fourier transform does not preserve their time-localization. In order to provide information on both frequency content and its time support, tools for simultaneous signal analysis in both time and frequency domain are designed, leading to the 2D TFRs [66], [67].

The first step towards the joint distribution in the time and frequency domains is the short-time Fourier transform (STFT), where the Fourier transform is performed on a signal $x(\tau)$ localized by a sliding window $h(\tau)$. Thus, the STFT provides an insight in a signal spectrum as a function of time [68], [69]:

$$STFT_x(t, f) = \int_{-\infty}^{\infty} x(\tau) h(\tau - t) e^{-j2\pi f \tau} d\tau. \quad (6)$$

The STFT is a linear TFR with elementary components that are well localized in time and frequency. An alternative approach to time-frequency signal analysis involves quadratic signal transforms representing energy time-frequency distributions. The spectrogram (SP) is a transitional form between these two groups of TFRs, obtained as the squared modulus of the STFT [70], [71]:

$$SP_x(t, f) = \left| \int_{-\infty}^{\infty} x(\tau) h(\tau - t) e^{-j2\pi f \tau} d\tau \right|^2. \quad (7)$$

Although the SP provides a signal representation with low interference terms, it has a poor resolution property that does not allow simultaneously good resolution in time and frequency. This trade-off between time and frequency resolution is due to using the fixed-size window – shorter windows provide better time resolution but lower frequency resolution. In contrast, longer windows increase frequency resolution

and decrease time resolution. In this study, we have used Hamming windows for all considered TFRs.

Given the limitations of the SP, alternative TFRs are investigated. The quadratic TFRs considered in this paper belong to Cohen's class characterized by time and frequency covariance. These TFRs use the concept of the analytic signal. This signal is obtained from the real-valued signal by removing the spectral content for the negative frequencies in order to reduce the number of cross-terms of quadratic distributions. The complex-valued analytic associate $y(t)$ of the real-valued signal $x(t)$ is defined using the Hilbert transform \mathcal{H} as [67]:

$$y(t) = x(t) + j\mathcal{H}\{x(t)\}. \quad (8)$$

The Wigner-Ville distribution (WV) [72] is an another important time-frequency energy distribution defined as a Fourier transform of the instantaneous autocorrelation function of the signal $y(t)$ [73], [74]:

$$WV_y(t, f) = \int_{-\infty}^{+\infty} y\left(t + \frac{\tau}{2}\right) y^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f \tau} d\tau, \quad (9)$$

where $*$ denotes the complex conjugate.

The WV is a fundamental TFR of Cohen's class satisfying many desirable mathematical properties. The WV provides high time-frequency resolution of the true signal components (auto-terms), but its inherent quadratic nature introduces interference terms (cross-terms) between those components. The interference terms can make a visual interpretation of the TFR difficult. Therefore, to suppress the unwanted interference terms, it is necessary to smooth the WV appropriately with the proper kernel. The kernels are often conveniently designed in the 2D Doppler-lag (ν, τ) domain [67]. The Doppler (ν) variable is obtained by the Fourier transform of the time (t) variable. It represents a frequency shift, similarly as τ represents a time shift. The Doppler-lag domain allows the application of kernels to the WV to be perceived as the filtering operation by defining the ambiguity function in the ambiguity (ν, τ) domain [67]. The TFRs obtained by kernel smoothing of the WV are often referred to as reduced-interference distributions (RIDs).

The pseudo Wigner-Ville distribution (PWV) is obtained similarly to (9) but using the instantaneous autocorrelation function windowed by $h(t)$ [75]. Time windowing is equivalent to frequency smoothing of the WV and leads to attenuation of the spurious terms oscillating in the frequency direction. Unfortunately, it also causes a decrease in the frequency resolution of the signal auto-terms. The PWV is defined as [76]:

$$PWV_y(t, f) = \int_{-\infty}^{+\infty} h(\tau) y\left(t + \frac{\tau}{2}\right) y^*\left(t - \frac{\tau}{2}\right) e^{-j2\pi f \tau} d\tau. \quad (10)$$

The cross-terms oscillating in the time direction are not attenuated in the PWV. This limitation is overcome by the smoothed pseudo Wigner-Ville distribution (SPWV), which additionally smooths the PWV in the time direction by a

time-smoothing window $g(t)$ [77]. The SPWV allows the smoothing of the WV to be controlled independently in both the time and frequency domain by selecting the lengths of the windows $h(t)$ and $g(t)$. However, there is a trade-off between the level of the interference terms and the time-frequency resolution. The SPWV is obtained as [78]:

$$SPWV_y(t, f) = \int_{-\infty}^{+\infty} h(\tau) \int_{-\infty}^{+\infty} g(u - t) y\left(u + \frac{\tau}{2}\right) \cdot y^*\left(u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \quad (11)$$

The RIDs are a group of TFRs designed with specific kernel functions designed to reduce the interference terms. The Choi-Williams distribution (CW) [79] uses an exponential kernel of width σ . The choice of σ allows control of the trade-off between resolution and suppression of interference terms. Namely, larger values of σ lead to an increase in auto-term resolution, while smaller values help reduce cross-terms. However, the CW does not provide independent control of time and frequency smoothings. This TFR shows higher interference levels when used for signals with components whose time or frequency supports overlap [80]. The CW is calculated as [81]:

$$CW_y(t, f) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\sqrt{\sigma}}{2\sqrt{\pi}|\tau|} e^{-\frac{u^2\sigma}{16\tau^2}} \cdot y\left(t + u + \frac{\tau}{2}\right) y^*\left(t + u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \quad (12)$$

The Butterworth distribution (BUD) is an extension of the CW that simultaneously provides better suppression of low-frequency interference terms and better preservation of auto-terms resolution. This is achieved by a kernel acting as a 2D low-pass filter in the ambiguity domain with adjustable passband and transition region parameters [82], [83]:

$$BUD_y(t, f) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \frac{\sqrt{\sigma}}{2|\tau|} e^{-\frac{|u|\sqrt{\sigma}}{|\tau|}} y\left(t + u + \frac{\tau}{2}\right) \cdot y^*\left(t + u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \quad (13)$$

The Born-Jordan distribution (BJ) is another RID that preserves time and frequency supports [84]. The use of a narrowband *sinc* kernel in the ambiguity domain provides good suppression of cross-terms at the cost of reduced auto-term resolution [85], [86]:

$$BJ_y(t, f) = \int_{-\infty}^{+\infty} \frac{1}{|\tau|} \int_{t-\frac{|\tau|}{2}}^{t+\frac{|\tau|}{2}} y\left(u + \frac{\tau}{2}\right) \cdot y^*\left(u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \quad (14)$$

The Zhao-Atlas-Marks distribution uses a cone-shaped kernel that simultaneously provides good resolution in time and frequency while reducing interference terms. The ZAM can be obtained by smoothing the BJ along the frequency

dimension [87], [88]:

$$ZAM_y(t, f) = \int_{-\infty}^{+\infty} h(\tau) \int_{t-\frac{|\tau|}{2}}^{t+\frac{|\tau|}{2}} y\left(u + \frac{\tau}{2}\right) \cdot y^*\left(u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \quad (15)$$

The RID with a kernel based on the first kind Bessel function of order one (RIDB) efficiently suppresses the interference terms while maintaining a high time-frequency resolution. The RIDB is defined as [83], [89]:

$$\begin{aligned} RIDB_y(t, f) &= \int_{-\infty}^{+\infty} h(\tau) \int_{t-|\tau|}^{t+|\tau|} \frac{2g(u)}{\pi |\tau|} \\ &\cdot \sqrt{1 - \left(\frac{u-t}{\tau}\right)^2} y\left(u + \frac{\tau}{2}\right) y^*\left(u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \end{aligned} \quad (16)$$

On the other hand, the RID with a kernel based on binomial coefficients (RIDBN) is calculated as [67], [90]:

$$\begin{aligned} RIDBN_y(t, f) &= \sum_{\tau=-\infty}^{+\infty} \sum_{u=-|\tau|}^{+|\tau|} \frac{1}{2^{2|\tau|+1}} \binom{2|\tau|+1}{|\tau|+u+1} \\ &\cdot y[t+u+\tau] y^*[t+u-\tau] e^{-j4\pi f \tau}. \end{aligned} \quad (17)$$

Another RID, designed with a kernel based on the Hanning window (RIDH), is [85]:

$$\begin{aligned} RIDH_y(t, f) &= \int_{-\infty}^{+\infty} h(\tau) \int_{-\frac{|\tau|}{2}}^{+\frac{|\tau|}{2}} \frac{g(u)}{|\tau|} \\ &\cdot \left(1 + \cos\left(\frac{2\pi u}{\tau}\right)\right) y\left(t + u + \frac{\tau}{2}\right) y^*\left(t + u - \frac{\tau}{2}\right) du \\ &\cdot e^{-j2\pi f \tau} d\tau. \end{aligned} \quad (18)$$

Next, we have also used the RID that uses a triangular window-based kernel (RIDT), defined as [85], [91]:

$$\begin{aligned} RIDT_y(t, f) &= \int_{-\infty}^{+\infty} h(\tau) \int_{-\frac{|\tau|}{2}}^{+\frac{|\tau|}{2}} \frac{2g(u)}{|\tau|} \left(1 - \frac{2|u|}{|\tau|}\right) \\ &\cdot y\left(t + u + \frac{\tau}{2}\right) y^*\left(t + u - \frac{\tau}{2}\right) du e^{-j2\pi f \tau} d\tau. \end{aligned} \quad (19)$$

Above defined TFRs are next used to train three CNN architectures for classification of GWs.

2) NON-STATIONARY SIGNAL TFR EXAMPLES

Previously defined 12 TFRs (SP, WV, PWV, SPWV, CW, BUD, BJ, ZAM, RIDB, RIDBN, RIDH, and RIDT) are next applied to 100 000 time series – resulting in 12 datasets (each containing 100 000 TFR images with a total of 1.2 million TFR images). Each TFR image shows the time-frequency distribution of a noisy signal 0.5 s long, with frequencies in

the range of 0 – 1024 Hz. The image resolution is 256 × 256. This image resolution was chosen to reduce the computational memory demands and to adjust the images to the size of the input layers of the CNN models.

The following examples illustrate the application of TFRs to the original time-series GW strain data. Fig. 2 shows the example of the time-series data representing the GW signal embedded in the noise, including the raw detector noise, the simulated noise-free GW signal, the noisy GW signal, and the whitened and high-pass filtered noisy GW signal. The peak of this signal is centered within the time interval shown. The desired NOMF-SNR of 19 dB is chosen for this illustrative example as the average of the 8 – 30 dB range used for the simulations. This data example's raw, noisy time series corresponds to the OMF-SNR value of 17.31 dB and the SNR value of –82.38 dB. On the other hand, Fig. 3 shows the randomly chosen time-series data example containing only the real-life noise without GW events. Fig. 3 shows both the raw and the whitened and high-pass filtered noise time series. Finally, Figs. 4 and 5 show the respective TFRs of the two time-series data examples considered here.

D. DEEP LEARNING CLASSIFICATION ALGORITHMS

1) INPUT DATA

We used a dataset with 100 000 time-series data examples of 1024 samples length for training and evaluation of the base model. The data generation procedure is described in subsection II-B. Half of the examples contain only noise, while the other half consists of the GW signals embedded in noise. Each data example represents a unique realization of both the GW signal waveform and the noise.

We applied 12 TFRs to the time-series dataset as described in the previous section, obtaining 12 datasets, each consisting of 100 000 TFR images of 256 × 256 size. These datasets are used with three deep CNN architectures: ResNet-101, Xception, and EfficientNet, which are discussed in the following subsection.

Each dataset is divided into three subsets: the training, validation, and test subset. The training subset comprises 70%, the validation subset 15%, and the test subset 15% of the examples in the dataset. Each subset maintains the same ratio of noise-only data examples to data examples with injected signals as in the complete set. No single data example belongs simultaneously to more than one subset.

We also normalized both the time-series data and the TFR images before feeding them into the CNN models. Each data example is scaled separately, according to the following expression:

$$y = \frac{x - x_{min}}{x_{max} - x_{min}}, \quad (20)$$

where y is the normalized value, x is the value to be normalized, while x_{min} and x_{max} are the minimum and maximum values of the data example, respectively. After normalization, all values are within the range of [0, 1].

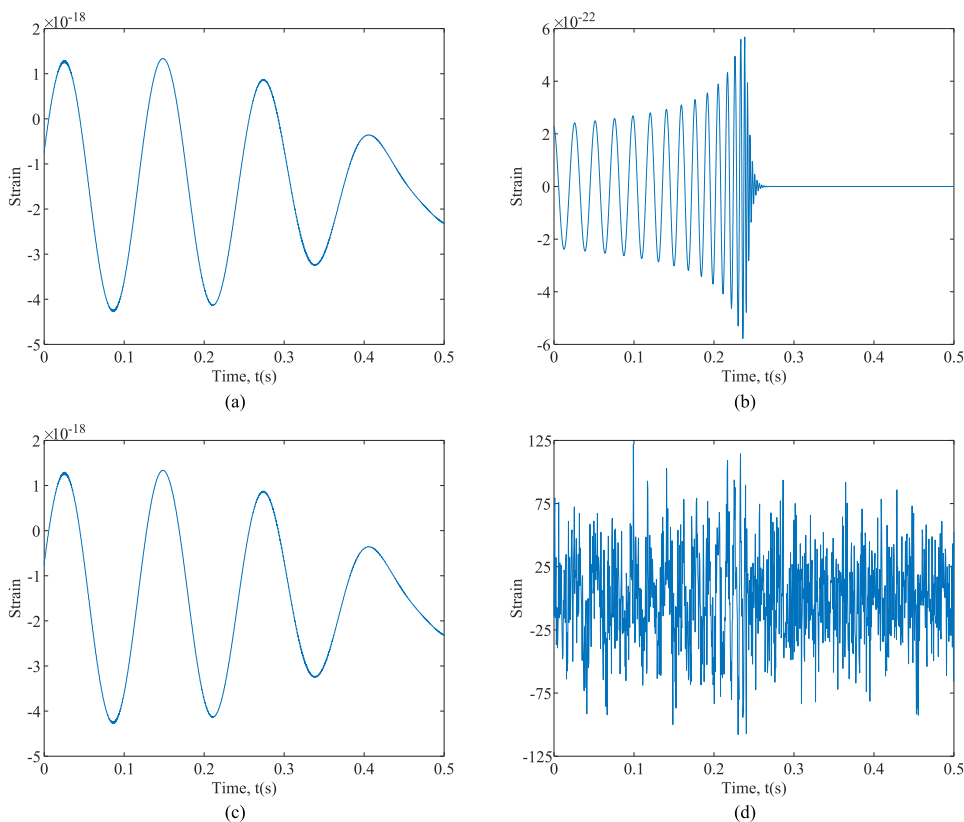


FIGURE 2. Time-series data example containing the GW signal embedded in the noise (NOMF-SNR = 19 dB, OMF-SNR = 17.31 dB, SNR = -82.38 dB). (a) Raw noise. (b) Simulated noise-free GW signal. (c) Noisy GW signal. (d) Whiten and high-pass filtered noisy GW signal.

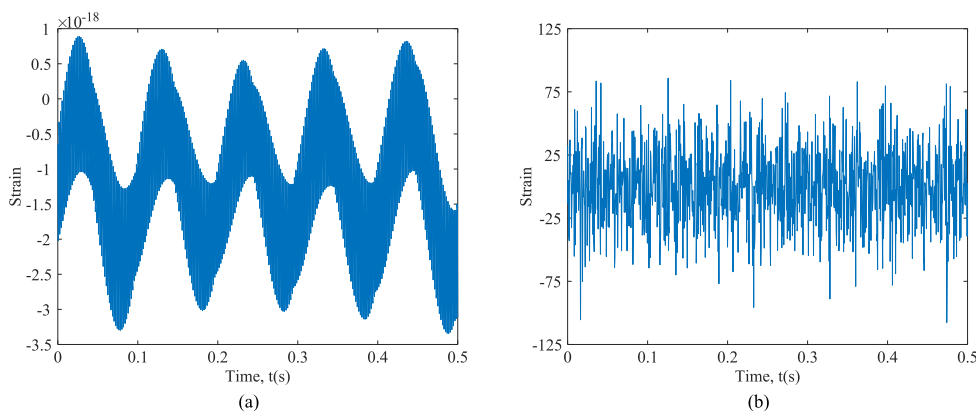


FIGURE 3. Time-series data example containing only noise. (a) Raw noise. (b) Whiten and high-pass filtered noise.

2) CNN MODELS

The base model used for the classification of the time-series GW data and the comparison with the method proposed in this paper is an adapted deep learning model presented in [35]. This model is based on the 1D deep CNN for the classification of the input time-series GW signals. The considered base model represents the state-of-the-art method for deep learning-based GW detection found in the literature, providing high classification performances and the corresponding

paper having the highest number of citations in the field of deep learning-based GW detection. The model is adapted in this study by using smaller convolution kernels. Namely, in the original paper, the input time-series vector had a length of 8192 samples, while in our work, the input time-series vector has eight times fewer samples - 1024. Since our input vector is eight times smaller, we reduced the kernel size by a factor of eight. Other parameters, such as stride, number of kernels, and dilations, were kept the same as in

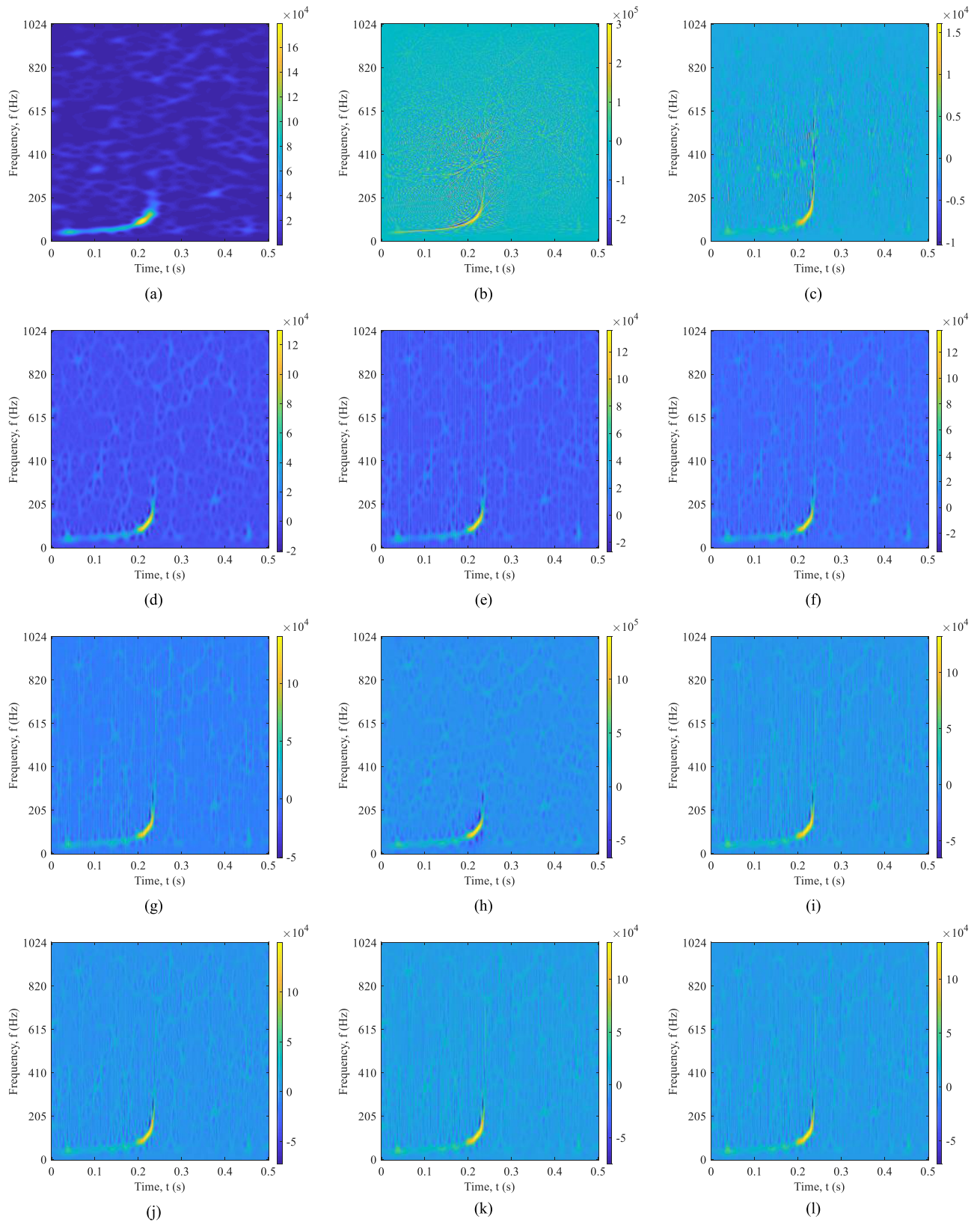


FIGURE 4. TFRs of the data example containing the GW signal embedded in the noise (NOMF-SNR = 19 dB, OMF-SNR = 17.31 dB, SNR = -82.38 dB). (a) SP. (b) WV. (c) PWV. (d) SPWV. (e) CW. (f) BUD. (g) BJ. (h) ZAM. (i) RIDB. (j) RIDBN. (k) RIDH. (l) RIDT.

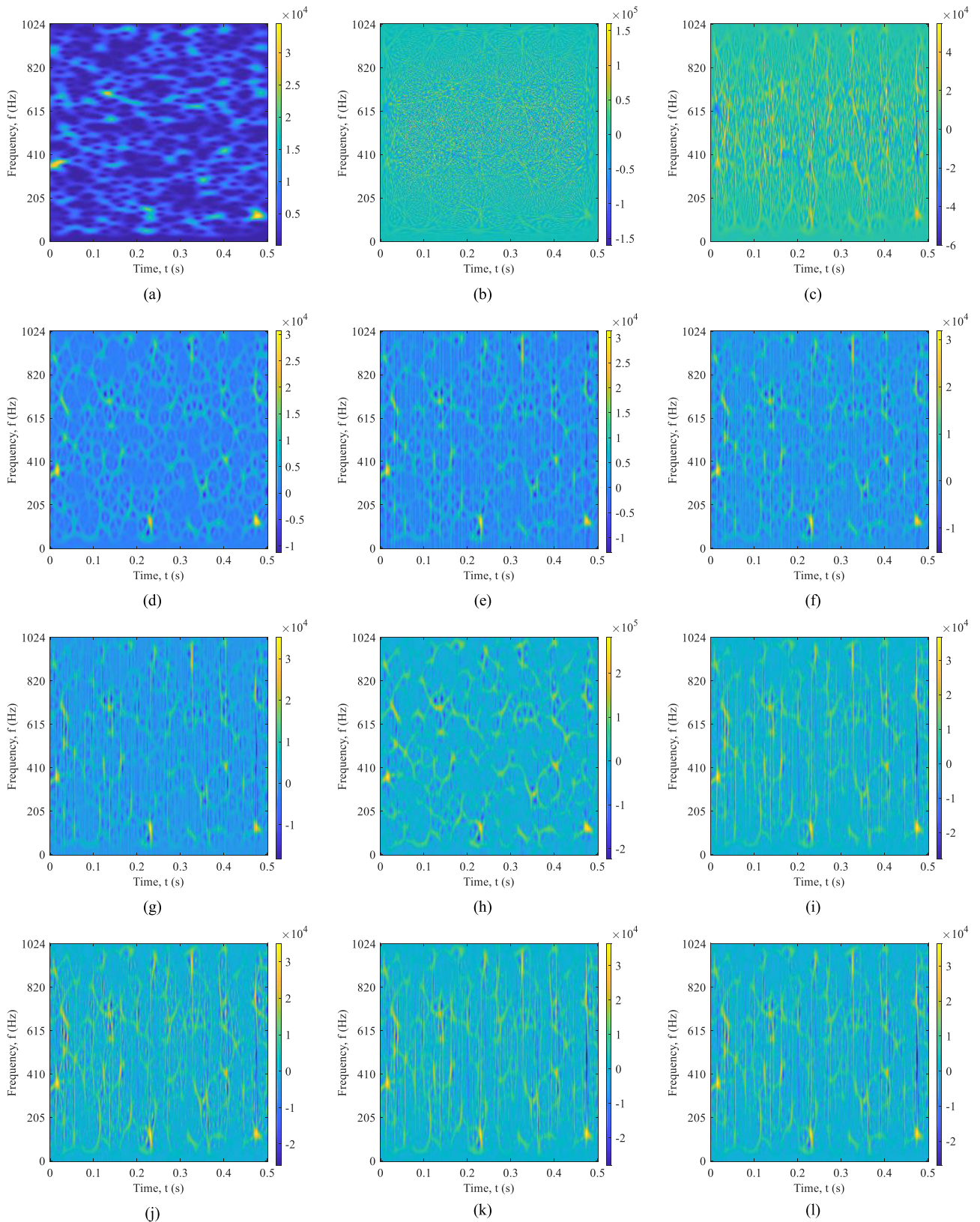


FIGURE 5. TFRs of the data example containing only noise. (a) SP. (b) WV. (c) PWV. (d) SPWV. (e) CW. (f) BUD. (g) BJ. (h) ZAM. (i) RIDB. (j) RIDBN. (k) RIDH. (l) RIDT.

the original paper. We also adopted the final output layer to one neuron with a sigmoid activation function: $e^x / (e^x + 1)$. This output neuron provides the probability that the input vector represents a GW signal. During the training, we used the binary cross-entropy loss function [92] and the Adam optimizer [93] with a learning rate $\alpha = 1 \times 10^{-5}$, while setting the batch size to 32. To find the best learning rate for the optimizer, we evaluated the following values on the validation dataset: $\alpha \in \{1 \times 10^{-1}, 1 \times 10^{-2}, \dots, 1 \times 10^{-6}\}$.

For the classification of TFRs, we have tested three different state-of-the-art deep learning architectures: ResNet-101, Xception, and EfficientNet. Except for the last-prediction layer, all models were the same as proposed by their authors in their respective papers. The prediction layer was adapted to predict only one number (instead of 1000). Hence it had one neuron with a sigmoid activation function (instead of the *softmax* activation [94]). Moreover, we used the binary cross-entropy loss function in all tested models and selected the best learning rate based on an experimental evaluation over the validation dataset. We tested the following learning rates for each of the models: $\alpha \in \{1 \times 10^{-1}, 1 \times 10^{-2}, \dots, 1 \times 10^{-6}\}$. We trained each deep learning model from scratch, i.e., no pre-trained models and transfer learning techniques were used. The overview of the classification procedure using CNN-based deep learning models with TFRs of the GW data as input is depicted in Fig. 6.

The first tested architecture is ResNet-101 [95]. The ResNet neural network class represents CNNs enhanced with skip connections [96]. Skip connections help with the vanishing gradient problem, which ultimately leads to deeper neural networks with higher accuracy. ResNet has several sizes based on the number of layers: ResNet-50, ResNet-101, and ResNet-152. Taking into account the available computational resources, we have selected the ResNet-101 architecture for our research. We utilized the Adam optimizer with the learning rate $\alpha = 1 \times 10^{-6}$ to train the ResNet-101. The batch size was set to 16.

Next tested CNN is Xception [97]. The name of this neural network comes from the *Extreme Inception*, as its main module may be interpreted as an extreme version of the Inception module found in the Inception CNN [98]. The Xception CNN is entirely based on the depthwise separable convolutions that significantly improve its accuracy and convergence speed. Besides depthwise separable convolutions, the Xception neural network utilizes the previously mentioned skip connections. We used the Adam optimizer with a learning rate set to $\alpha = 1 \times 10^{-4}$ for training the Xception, while the batch size was set to 32.

The final tested CNN is EfficientNet, which can be considered the current state-of-the-art CNN in terms of classification tasks [99]. The idea behind EfficientNet lies in compound scaling. Namely, the ResNet-based topologies achieve an increased accuracy by adding more layers. This approach is also known as "going deeper". On the other hand, network width scaling is commonly used for small-size

TABLE 1. Parameters for training the deep learning models.

CNN architecture	Parameters			
	Loss function	Optimizer	Learning rate	Batch size
Base model	Binary cross-entropy	Adam	1×10^{-5}	32
ResNet-101	Binary cross-entropy	Adam	1×10^{-6}	16
Xception	Binary cross-entropy	Adam	1×10^{-4}	32
EfficientNet	Binary cross-entropy	RMSProp	1×10^{-4}	32

models [100]. As discussed in [101], wider neural networks can capture more detailed features. However, if the neural network is shallow and extremely wide, it tends not to capture high-level features. Last but not least, resolution scaling has also been shown to be an important factor in achieving high accuracy [102]. Therefore, instead of focusing on finding the best neural network topology, EfficientNet is focused on finding the best ratio of the width, depth, and resolution for a given base topology. For this purpose, EfficientNet utilizes the compound scaling method, which uses the compound coefficient ϕ that scales network width, depth, and resolution. In our paper, we have utilized the EfficientNet-B2 neural network. During the training, we used a batch size of 32 and the RMSProp optimizer [103] with the learning rate $\alpha = 1 \times 10^{-4}$.

The main parameters used for training the above-described base model and deep learning models based on the 2D CNN architectures are summarized in Table 1.

3) PERFORMANCE METRICS

We evaluated the model performance on the test dataset using the following evaluation metrics: classification accuracy, area under the receiver operating characteristic (ROC) curve (ROC AUC), recall, precision, F1 score, and area under the precision-recall curve (PR AUC).

The classification accuracy represents the ratio between the number of correct predictions and the total number of predictions made. We calculated the classification accuracy for each TFR-CNN model using the probability threshold that provides the optimal model performance. The probability threshold was selected based on the evaluation performed on the validation dataset. For our binary classification problem, data examples containing GW signals can be denoted as a positive class, while those examples with no GW signals are considered a negative class. Thus, the classification accuracy can be defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (21)$$

where TP , TN , FP , and FN are True Positives, True Negatives, False Positives, and False Negatives, respectively.

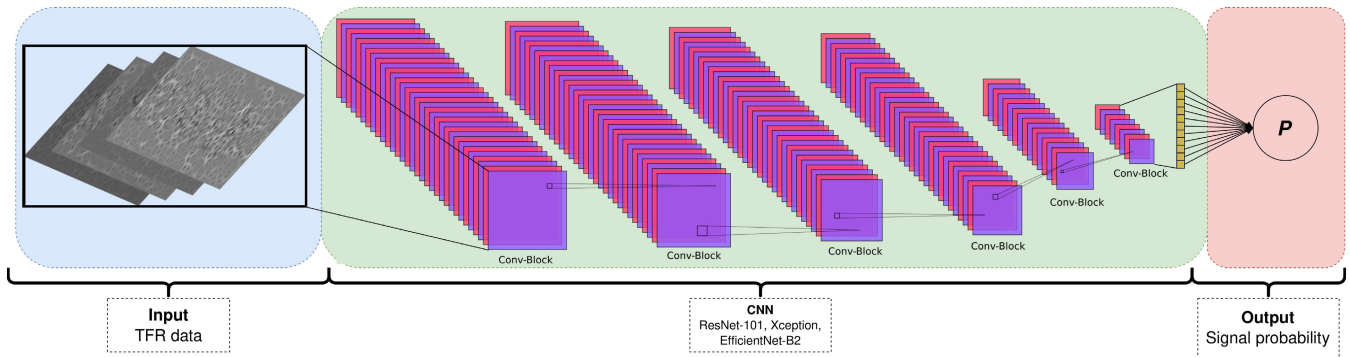


FIGURE 6. CNN models-based deep learning classification of GW data TFRs.

The true positive rate (*TPR*) or sensitivity is calculated as

$$TPR = \frac{TP}{TP + FN}, \tag{22}$$

whereas the false positive rate (*FPR*) is defined as

$$FPR = \frac{FP}{FP + TN}. \tag{23}$$

The *TPR* and *FPR* have values in the range [0, 1]. The ROC curve is a plot showing the *TPR* against the *FPR* values obtained by model evaluation at different threshold values used to map the probabilities to class labels. The ROC AUC value is calculated as the area under the ROC curve. The ROC AUC provides information on how effective the model is at discriminating classes. The models with higher ROC AUC values maintain their performance across different probability thresholds and provide good class separation, with the ideal value of 1.0 indicating the perfect classifier.

The recall measures how well the positive class was predicted and is calculated in the same way as *TPR* in (22). On the other hand, precision is the percentage of examples assigned to the positive class that actually belong to the positive class:

$$Precision = \frac{TP}{TP + FP}. \tag{24}$$

F1 score is an evaluation metric that combines precision and recall into a single score used to balance both metrics simultaneously. F1 score is obtained as a harmonic mean of the precision and recall:

$$F1\ score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{25}$$

A plot obtained by calculating precision and recall values for the model evaluated at different probability threshold settings is called the precision-recall curve. Finally, the PR AUC is calculated as the area under this curve. High PR AUC values indicate high precision and high recall, with the value of 1.0 representing a perfect model performance.

4) STATISTICAL SIGNIFICANCE TESTS

We performed a series of statistical tests on the obtained classification results to check whether the differences in the results generated by different deep learning models were statistically significant. For this purpose, we used McNemar’s test [104]. The McNemar’s test was shown to be the only statistical test with an acceptable type I error, i.e., the probability of falsely detecting a difference when there is no difference, for algorithms that can be run only once [105]. These algorithms include deep learning models characterized by computationally intensive and time-consuming training processes. We paired the results obtained by the base model with those obtained by each TFR-CNN combination and computed McNemar’s test statistic. The significance level was set to $\alpha = 0.01$. Since each TFR is used with three different deep learning algorithms, we used Bonferroni correction and divided the significance level by 3, thus obtaining $\alpha = 0.00333$. The obtained statistical results, described and elaborated in the next section, show whether the individual TFR enhances the information provided by the original time-series data. This can be observed as a significant improvement in the performance of the CNN-based deep learning models compared to the base model for the classification of non-stationary GW signals.

III. RESULTS AND DISCUSSION

Next, we present the classification results. We trained each deep CNN architecture (ResNet-101, Xception, and EfficientNet) on each of the 12 TFRs for 75 000 input signals, validated on 15 000 signals, and evaluated the model performance on the test dataset of 15 000 signals using the beforementioned performance metrics.

Table 2 provides the values of probability thresholds used with the base model and each TFR-CNN model on the test dataset. The probability threshold values were chosen as the ones providing the optimal model performance based on its evaluation on the validation dataset for different threshold values with a step of 0.1.

Table 3 shows the classification accuracy results obtained by the base model and each combination of the TFR and CNN architecture. The base model achieved a classification

TABLE 2. Probability threshold values used for model evaluation on the test dataset.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.5	0.8	0.6
WV	0.6	0.5	0.4
PWV	0.7	0.4	0.8
SPWV	0.4	0.6	0.8
CW	0.7	0.8	0.7
BUD	0.5	0.5	0.4
BJ	0.2	0.5	0.3
ZAM	0.8	0.2	0.4
RIDB	0.7	0.5	0.5
RIDBN	0.7	0.7	0.5
RIDH	0.8	0.7	0.7
RIDT	0.6	0.5	0.9
Base model	0.5		

TABLE 3. Classification accuracy for model evaluation on the test dataset. Highest accuracy values for each CNN architecture are marked in bold.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.9691	0.9695	0.9710
WV	0.9654	0.9704	0.9682
PWV	0.9689	0.9697	0.9693
SPWV	0.9676	0.9699	0.9691
CW	0.9695	0.9684	0.9698
BUD	0.9683	0.9683	0.9689
BJ	0.9683	0.9680	0.9699
ZAM	0.9681	0.9682	0.9657
RIDB	0.9685	0.9677	0.9683
RIDBN	0.9693	0.9691	0.9680
RIDH	0.9679	0.9687	0.9700
RIDT	0.9680	0.9686	0.9685
Base model	0.9315		

accuracy of 93.15%. As seen in Table 3, all TFR-CNN combinations show very similar performance with very high classification accuracies. The ResNet-101 architecture achieves the lowest classification accuracy (96.54%) when used with the WV representation of the input data and the highest accuracy (96.95%) when used with the CW representation. The Xception CNN architecture provides classification accuracies in the range between 96.77% and 97.04%, obtained using the RIDB and the WV data representation, respectively. The EfficientNet provides the lowest accuracy value (96.57%) for the ZAM and the highest value (97.10%) for the SP representation.

The overall classification accuracy of the TFR-CNN models ranges from 96.54% (achieved by WV time-frequency representation and the ResNet-101 architecture) to 97.10% (obtained by the SP - EfficientNet combination). Thus, the classification accuracy of the CNN models trained on TFRs of the input data outperforms the classification accuracy of the base model trained on the time-series input data by 3.39% to 3.95%.

The obtained ROC AUC values are given in Table 4. The base model achieved an ROC AUC of 0.9679. As seen in

TABLE 4. ROC AUC results for model evaluation on the test dataset. Highest ROC AUC values for each CNN architecture are marked in bold.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.9873	0.9881	0.9882
WV	0.9854	0.9871	0.9857
PWV	0.9865	0.9873	0.9869
SPWV	0.9868	0.9880	0.9877
CW	0.9870	0.9873	0.9885
BUD	0.9880	0.9867	0.9869
BJ	0.9880	0.9871	0.9882
ZAM	0.9871	0.9876	0.9875
RIDB	0.9881	0.9873	0.9864
RIDBN	0.9880	0.9878	0.9875
RIDH	0.9863	0.9875	0.9880
RIDT	0.9871	0.9862	0.9850
Base model	0.9679		

TABLE 5. Recall results for model evaluation on the test dataset. Highest recall values for each CNN architecture are marked in bold.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.9475	0.9445	0.9472
WV	0.9425	0.9519	0.9447
PWV	0.9443	0.9495	0.9505
SPWV	0.9500	0.9512	0.9481
CW	0.9455	0.9415	0.9479
BUD	0.9519	0.9515	0.9472
BJ	0.9553	0.9488	0.9491
ZAM	0.9479	0.9587	0.9553
RIDB	0.9449	0.9533	0.9444
RIDBN	0.9507	0.9520	0.9485
RIDH	0.9424	0.9467	0.9495
RIDT	0.9481	0.9497	0.9428
Base model	0.8885		

Table 4, each TFR-CNN model provides very high ROC AUC values. The ResNet-101 achieves the lowest ROC AUC (0.9854) when coupled with the WV representation and the highest (0.9881) when combined with the RIDB representation. The ROC AUC values of the Xception CNN range from 0.9862 (achieved by the RIDT) to 0.9881 (provided by the SP). The lowest ROC AUC value (0.9850) of the EfficientNet is obtained with the RIDT, while the highest value (0.9885) with the CW time-frequency data representation.

Overall, the ROC AUC values obtained by the TFR-CNN models are in the range between 0.9850 (provided by the RIDT representation with the EfficientNet architecture) and 0.9885 (produced by the CW - EfficientNet combination), which exceeds the ROC AUC value of the base model by 1.71% to 2.06%.

Table 5 gives the recall values obtained by the base model and TFR-CNN models evaluation. The base model achieves a recall of 88.85%. The results presented in Table 5 show that all TFR-CNN models have high recall values and outperform the base model. The ResNet-101 has recall values ranging between 94.24% (achieved by the RIDH) and 95.53%

TABLE 6. Precision results for model evaluation on the test dataset. Highest precision values for each CNN architecture are marked in bold.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.9904	0.9942	0.9945
WV	0.9877	0.9885	0.9913
PWV	0.9931	0.9894	0.9875
SPWV	0.9847	0.9881	0.9896
CW	0.9933	0.9951	0.9914
BUD	0.9843	0.9845	0.9902
BJ	0.9807	0.9867	0.9903
ZAM	0.9879	0.9773	0.9755
RIDB	0.9917	0.9816	0.9919
RIDBN	0.9874	0.9856	0.9870
RIDH	0.9930	0.9902	0.9901
RIDT	0.9874	0.9870	0.9940
Base model	0.9720		

(obtained by the BJ). The lowest (94.15%) and the highest (95.87%) recall values for the Xception are obtained with the CW and the ZAM data representation, respectively. The EfficientNet gives recall values between 94.28% (provided by the RIDT) and 95.53% (achieved by the ZAM).

Taking into account the performance across all three CNN architectures, the recall values range from 94.15% (achieved by the CW - Xception) to 95.87% (obtained by the ZAM - Xception). Therefore, the recall values of the TFR-CNN models are 5.30% to 7.02% higher than that of the base model.

The values of precision obtained by the tested models are shown in Table 6. The base model offers a precision of 97.20%, which is surpassed by the precision values obtained by the TFR-CNN models. The ResNet-101 provides precision values ranging between 98.07% (for the BJ) and 99.33% (for the CW), the Xception has precision between 97.73% (for the ZAM) and 99.51% (for the CW), while the EfficientNet offers precision values from the range of 97.55% (for the ZAM) to 99.45% (for the SP).

The precision of the TFR-CNN models ranges from 97.55% (for the ZAM - EfficientNet combination) to 99.51% (for the CW - Xception combination), thus exceeding that of the base model by 0.35% to 2.31%.

Table 7 provides the F1 score values obtained by the considered deep learning models. The base model has an F1 score of 92.84%, while the TFR-CNN models exceed this value. The F1 score values obtained by the ResNet-101 range from 96.46% (for the WV) to 96.88% (for the CW). The Xception provides an F1 score from the range of 96.73% (for the RIDB) to 96.98% (for the WV), whereas the EfficientNet achieves F1 scores between 96.53% (for the ZAM) and 97.03% (for the SP).

The TFR-CNN models provide F1 score values from 96.46% (for the WV - ResNet-101) to 97.03% (for the SP - EfficientNet). These values surpass the F1 score obtained by the base model by 3.62% to 4.19%.

The final evaluation metric of the tested models is the PR AUC, whose values are shown in Table 8. The base model

TABLE 7. F1 score results for model evaluation on the test dataset. Highest F1 score values for each CNN architecture are marked in bold.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.9684	0.9688	0.9703
WV	0.9646	0.9698	0.9674
PWV	0.9681	0.9690	0.9687
SPWV	0.9670	0.9693	0.9684
CW	0.9688	0.9675	0.9691
BUD	0.9678	0.9677	0.9682
BJ	0.9679	0.9674	0.9692
ZAM	0.9675	0.9679	0.9653
RIDB	0.9678	0.9673	0.9676
RIDBN	0.9687	0.9685	0.9674
RIDH	0.9670	0.9680	0.9694
RIDT	0.9674	0.9680	0.9677
Base model	0.9284		

TABLE 8. PR AUC results for model evaluation on the test dataset. Highest PR AUC values for each CNN architecture are marked in bold.

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	0.9912	0.9916	0.9918
WV	0.9900	0.9911	0.9901
PWV	0.9908	0.9912	0.9909
SPWV	0.9909	0.9916	0.9914
CW	0.9912	0.9913	0.9920
BUD	0.9916	0.9908	0.9909
BJ	0.9916	0.9911	0.9917
ZAM	0.9910	0.9915	0.9913
RIDB	0.9916	0.9912	0.9907
RIDBN	0.9916	0.9915	0.9913
RIDH	0.9906	0.9913	0.9916
RIDT	0.9910	0.9905	0.9899
Base model	0.9772		

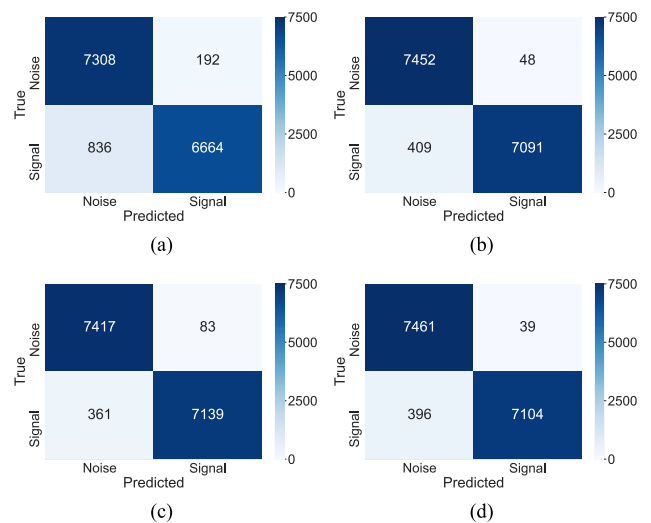


FIGURE 7. Confusion matrices for the selected deep learning models. (a) Base model. (b) CW-ResNet-101 model. (c) WV-Xception model. (d) SP-EfficientNet model.

provides a PR AUC of 0.9772, lower than the values obtained by the TFR-CNN models. The ResNet-101 has PR AUC

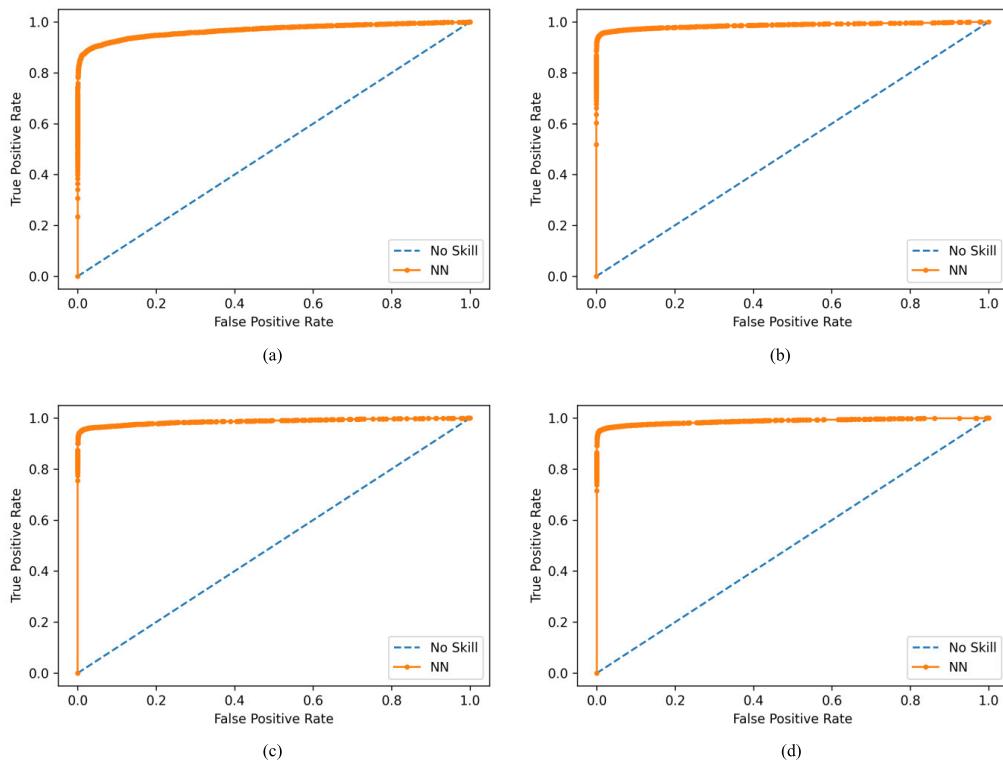


FIGURE 8. ROC curves for the selected deep learning models. (a) Base model. (b) CW-ResNet-101 model. (c) WV-Xception model. (d) SP-EfficientNet model.

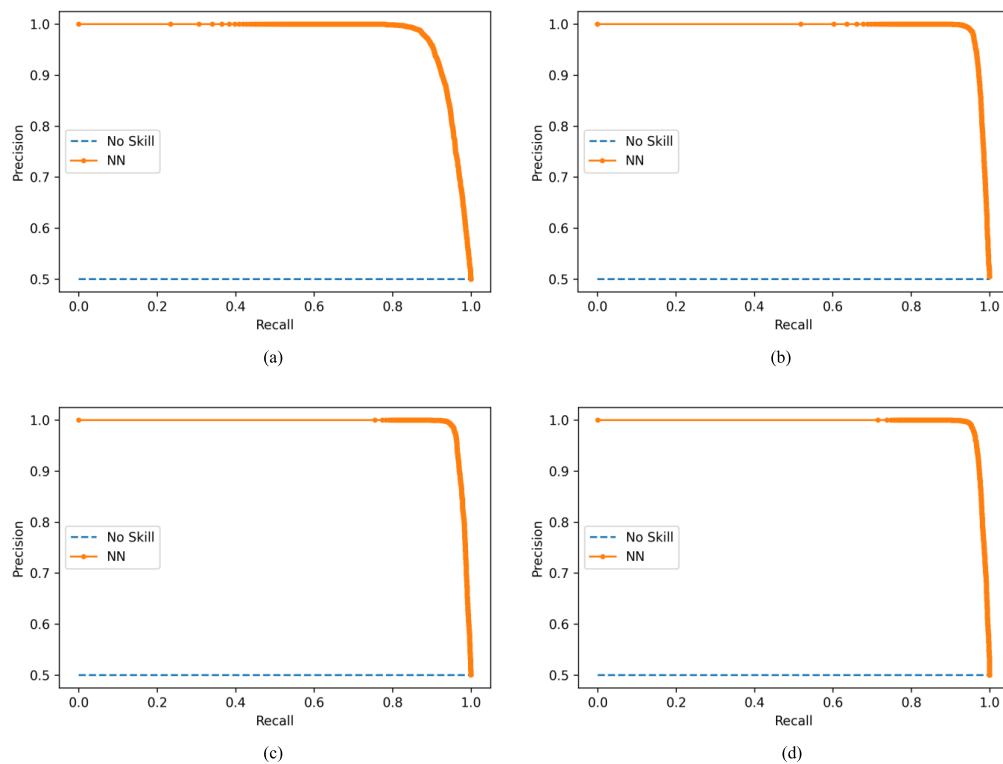


FIGURE 9. Precision-recall curves for the selected deep learning models. (a) Base model. (b) CW-ResNet-101 model. (c) WV-Xception model. (d) SP-EfficientNet model.

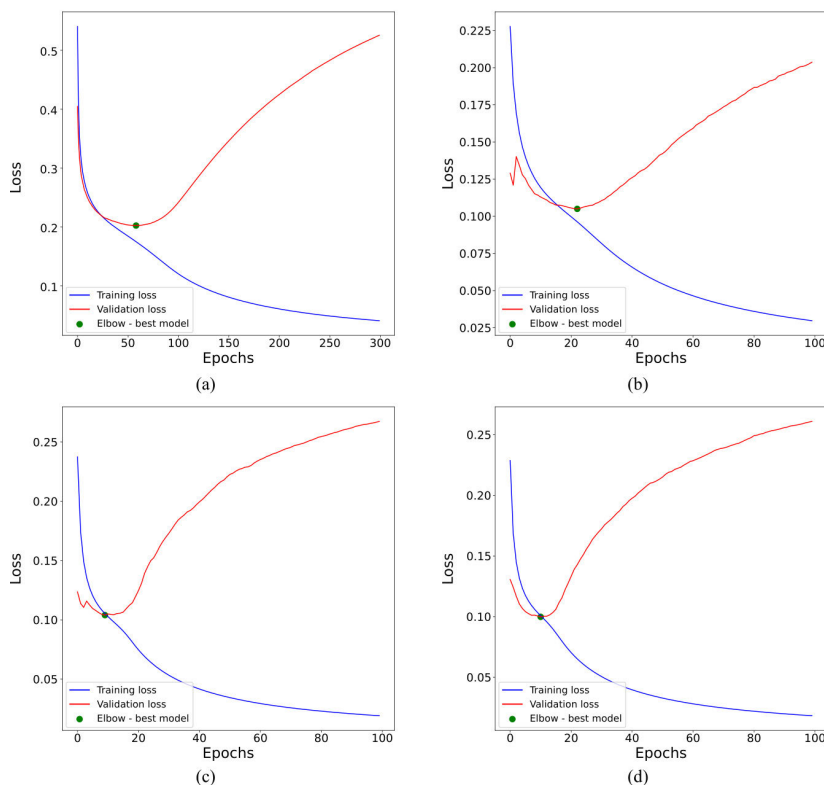


FIGURE 10. Training and validation loss curves for the selected deep learning models. (a) Base model. (b) CW-ResNet-101 model. (c) WV-Xception model. (d) SP-EfficientNet model.

values in the range from 0.9900 (for the WV) to 0.9916 (for the BUD), while the Xception covers the range of 0.9905 (for the RIDT) to 0.9916 (for the SP). The EfficientNet achieves PR AUC between 0.9899 (for the RIDT) and 0.9920 (for the CW).

Overall, the PR AUC values of the tested TFR-CNN models range from 0.9899 (for the RIDT - EfficientNet) to 0.9920 (for the CW - EfficientNet), thus surpassing the base model value by 1.27% to 1.48%.

Next, we selected the TFRs that give the highest classification accuracy value for each of the three tested CNN architectures and presented additional detailed performance indicators for these TFR-CNN models. The selected TFR-CNN combinations are the CW - ResNet-101, WV- Xception, and SP - EfficientNet. The confusion matrices for the base model and the selected TFR-CNN models are shown in Fig. 7. In addition, Fig. 8 displays the ROC curves, while Fig. 9 depicts the precision-recall curves for the considered deep learning models. These additional performance indicators confirm the high performance of the TFR-CNN models, which are also superior to the base model.

Furthermore, the learning curves for the base model and the selected TFR-CNN models are also provided, including the training and validation loss curves shown in Fig. 10 and the training and validation accuracy curves shown in Fig. 11. The elbow method was used during the training, i.e., the model with the parameters achieving the lowest

validation loss was selected to prevent the potential overfitting of the deep learning models. The selected model is marked by a green dot in the plots shown in Figs. 10 and 11. As shown in Fig. 10, the base model achieved the lowest validation loss after 58 epochs, the CW - ResNet-101 model after 22 epochs, the WV - Xception model after 9 epochs, and the SP - EfficientNet model after 10 epochs. Therefore, the models did not overfit. Moreover, the models did not underfit either since the corresponding learning curves in Figs. 10 and 11 show their high performance on the training and validation datasets. It is also important to notice that since there were approximately 2 343 or 4 686 updates (depending on the utilized batch size) in one epoch, the models could dramatically improve their accuracy between the epochs.

Finally, Table 9 contains the p -values obtained by McNemar's statistical test for each combination of the TFR of the input data and the CNN model. Each p -value reported in Table 9 is less than the significance level ($p < 0.00333$). In fact, the obtained p -values are very close to zero. Therefore, we can reject the null hypothesis of the statistical test and infer that the differences in the classification results produced by the CNN models using TFRs and those produced by the base model are statistically significant.

The analysis of the obtained results shows that utilizing the TFRs of the non-stationary time-series significantly improves the performance of the deep learning classification algorithms compared to the base model using the original

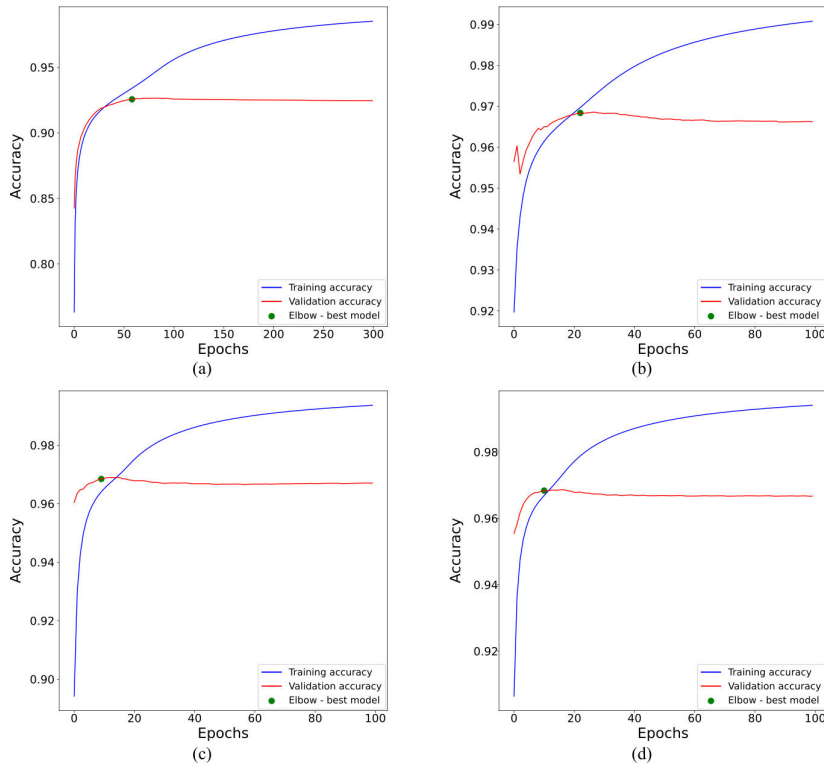


FIGURE 11. Training and validation accuracy curves for the selected deep learning models. (a) Base model. (b) CW-ResNet-101 model. (c) WV-Xception model. (d) SP-EfficientNet model.

TABLE 9. p-value results of the statistical tests ($\alpha = 0.00333$).

TFR	CNN architecture		
	ResNet-101	Xception	EfficientNet
SP	1.3×10^{-93}	9.9×10^{-100}	2.6×10^{-108}
WV	5.7×10^{-79}	1.4×10^{-95}	9.7×10^{-93}
PWV	7.4×10^{-97}	9.5×10^{-97}	1.2×10^{-91}
SPWV	8.4×10^{-84}	3.0×10^{-93}	1.1×10^{-93}
CW	1.8×10^{-99}	1.1×10^{-98}	1.9×10^{-98}
BUD	4.2×10^{-84}	6.6×10^{-84}	5.1×10^{-93}
BJ	5.8×10^{-81}	2.0×10^{-85}	2.3×10^{-96}
ZAM	4.0×10^{-87}	2.2×10^{-78}	1.4×10^{-67}
RIDB	8.8×10^{-95}	3.1×10^{-79}	1.2×10^{-92}
RIDBN	4.0×10^{-92}	8.2×10^{-89}	2.0×10^{-85}
RIDH	5.7×10^{-91}	5.8×10^{-92}	8.7×10^{-96}
RIDT	2.2×10^{-87}	3.4×10^{-89}	3.7×10^{-97}

time-series data only. The improved performance is reflected in increased classification accuracy, ROC AUC, recall, precision, F1 score, and PR AUC values. Each considered Cohen's class TFR paired with each considered CNN architecture achieves high values of the performance metrics, which are also similar to the values obtained by using the SP data representation. Since the SP is commonly used for time-frequency representation and analysis of non-stationary signals, the obtained results suggest that the alternative TFRs of Cohen's class can also successfully upgrade deep learning classification algorithms.

Finally, the increased accuracy of the deep learning classification distinguishing between data containing GW signals

in intensive noise and those containing only noise can be utilized in the field of GW astronomy to improve the detection of GW events and further increase the detection rate.

IV. CONCLUSION

In this work, we proposed a method for the classification of noisy non-stationary signals based on Cohen's class TFRs and deep learning algorithms. We analyzed the efficiency of the proposed approach on the problem of detecting GW signals in intensive real-life, non-stationary, non-white, and non-Gaussian noise, with the SNR values ranging from -123.46 to -2.27 dB. For this purpose, we trained three CNN architectures (ResNet-101, Xception, and EfficientNet) with 12 different TFRs of the input data generated using the real detector noise and simulated GW signals. We also compared the results obtained by the proposed method to the results obtained by the base deep learning model with the original time-series form of GW signals as input, where the considered base model represents the state-of-the-art method for deep learning-based GW detection. The analysis of the obtained results shows that the proposed method outperforms the base model by 3.39% to 3.95% in terms of classification accuracy, 1.71% to 2.06% in terms of ROC AUC, 5.30% to 7.02% in terms of recall, 0.35% to 2.31% in terms of precision, 3.62% to 4.19% in terms of F1 score, and 1.27% to 1.48% in terms of PR AUC. Moreover, the superior performance of the proposed method over the base model is also shown with the additional performance

indicators, including the confusion matrices, the ROC curves, and the precision-recall curves. Finally, a series of statistical tests confirmed the statistical significance of the obtained differences in performances.

The obtained results indicate that the alternative TFRs from Cohen's class, when used with deep learning classification algorithms, can through the better structuring of the information and the improved intelligibility of the representation significantly improve the performance of these algorithms for the classification of noisy non-stationary time-series signals.

In addition to its practical application in GW astronomy, the approach proposed in this paper using deep CNN models trained with Cohen's class TFRs of the input data can also be applied in other fields dealing with noisy non-stationary signals, such as seismology, speech analysis, and EEG signal analysis. These applications remain potential subjects of future studies. Due to their usefulness in deep learning demonstrated in this study, Cohen's class TFRs can also be used as data augmentation methods when dealing with reduced-size datasets by applying multiple high-performance TFRs to the original data, thus obtaining additional data for training and testing deep learning models.

Also, future research could explore the possibility of ensemble learning with Cohen's class TFRs, as well as using multiple TFRs of the same signal as simultaneous inputs to the CNN model.

Finally, the effects of applying the locally adaptive filtering algorithms to the noisy non-stationary signals before the TFR extraction and deep learning-based classification could also be investigated, as planned for our future work.

ACKNOWLEDGMENT

This research has made use of data, software, and/or web tools obtained from the Gravitational Wave Open Science Center (<https://www.gw-openscience.org/>), a service of LIGO Laboratory, the LIGO Scientific Collaboration, and the Virgo Collaboration. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC), U.K., the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN), and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, and Spain.

REFERENCES

- [1] B. P. Abbott, "Observation of gravitational waves from a binary black hole merger," *Phys. Rev. Lett.*, vol. 116, no. 6, Feb. 2016, Art. no. 061102.
- [2] J. Aasi, B. P. Abbott, R. Abbott, T. Abbott, M. R. Abernathy, K. Ackley, C. Adams, T. Adams, and P. Addesso, "Advanced LIGO," *Classical Quantum Gravity*, vol. 32, no. 7, Mar. 2015, Art. no. 074001.
- [3] B. P. Abbott *et al.*, "Characterization of transient noise in advanced LIGO relevant to gravitational wave signal GW150914," *Classical Quantum Gravity*, vol. 33, no. 13, Jun. 2016, Art. no. 134001.
- [4] D. V. Martynov, E. D. Hall, B. P. Abbott, R. Abbott, T. D. Abbott, C. Adams, R. X. Adhikari, R. A. Anderson, S. B. Anderson, K. Arai, and M. A. Arain, "Sensitivity of the advanced LIGO detectors at the beginning of gravitational wave astronomy," *Phys. Rev. D, Part. Fields*, vol. 93, Jun. 2016, Art. no. 112004.
- [5] S. A. Usman *et al.*, "The PyCBC search for gravitational waves from compact binary coalescence," *Classical Quantum Gravity*, vol. 33, no. 21, Oct. 2016, Art. no. 215004.
- [6] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, "Rapid detection of gravitational waves from compact binary mergers with PyCBC live," *Phys. Rev. D, Part. Fields*, vol. 98, no. 2, Jul. 2018, Art. no. 024050.
- [7] A. Torres-Forné, E. Cuoco, A. Marquina, J. A. Font, and J. M. Ibáñez, "Total-variation methods for gravitational-wave denoising: Performance tests on advanced LIGO data," *Phys. Rev. D, Part. Fields*, vol. 98, no. 8, Oct. 2018, Art. no. 084013.
- [8] N. Lopac, J. Lerga, and E. Cuoco, "Gravitational-wave burst signals denoising based on the adaptive modification of the intersection of confidence intervals rule," *Sensors*, vol. 20, no. 23, p. 6920, Dec. 2020.
- [9] E. Hossain, I. Khan, F. Un-Noor, S. S. Sikander, and M. S. H. Sunny, "Application of big data and machine learning in smart grid, and associated security concerns: A review," *IEEE Access*, vol. 7, pp. 13960–13988, 2019.
- [10] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019.
- [11] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.
- [12] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.
- [13] J. Zhang, J. Yu, S. Fu, and X. Tian, "Adoption value of deep learning and serological indicators in the screening of atrophic gastritis based on artificial intelligence," *J. Supercomput.*, vol. 77, no. 8, pp. 8674–8693, Jan. 2021.
- [14] E. Cuoco *et al.*, "Enhancing gravitational-wave science with machine learning," *Mach. Learn., Sci. Technol.*, vol. 2, no. 1, Dec. 2020, Art. no. 011002.
- [15] A. Torres-Forné, A. Marquina, J. A. Font, and J. M. Ibáñez, "Denoising of gravitational wave signals via dictionary learning algorithms," *Phys. Rev. D, Part. Fields*, vol. 94, no. 12, Dec. 2016, Art. no. 124040.
- [16] W. Wei and E. A. Huerta, "Gravitational wave denoising of binary black hole mergers with deep learning," *Phys. Lett. B*, vol. 800, Jan. 2020, Art. no. 135081.
- [17] G. Vajente, Y. Huang, M. Isi, J. C. Driggers, J. S. Kissel, M. J. Szczepańczyk, and S. Vitale, "Machine-learning nonstationary noise out of gravitational-wave detectors," *Phys. Rev. D, Part. Fields*, vol. 101, no. 4, Feb. 2020, Art. no. 042003.
- [18] A. J. K. Chua, C. R. Galley, and M. Vallisneri, "Reduced-order modeling with artificial neurons for gravitational-wave inference," *Phys. Rev. Lett.*, vol. 122, no. 21, May 2019, Art. no. 211101.
- [19] H. Nakano, T. Narikawa, K.-I. Oohara, K. Sakai, H.-A. Shinkai, H. Takahashi, T. Tanaka, N. Uchikata, S. Yamamoto, and T. S. Yamamoto, "Comparison of various methods to extract ringdown frequency from gravitational wave data," *Phys. Rev. D, Part. Fields*, vol. 99, no. 12, Jun. 2019, Art. no. 124032.
- [20] A. J. K. Chua and M. Vallisneri, "Learning Bayesian posteriors with neural networks for gravitational-wave inference," *Phys. Rev. Lett.*, vol. 124, no. 4, Jan. 2020, Art. no. 041102.
- [21] M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson, T. K. Lee, C. Lintott, T. B. Littenberg, A. Lundgren, C. Østerlund, J. R. Smith, L. Trouille, and V. Kalogera, "Gravity spy: Integrating advanced LIGO detector characterization, machine learning, and citizen science," *Classical Quantum Gravity*, vol. 34, no. 6, Feb. 2017, Art. no. 064003.
- [22] J. Powell, A. Torres-Forné, R. Lynch, D. Trifirò, E. Cuoco, M. Cavaglià, I. S. Heng, and J. A. Font, "Classification methods for noise transients in advanced gravitational-wave detectors II: Performance tests on advanced LIGO data," *Classical Quantum Gravity*, vol. 34, no. 3, Jan. 2017, Art. no. 034002.

- [23] D. George, H. Shen, and E. A. Huerta, "Classification and unsupervised clustering of LIGO data with deep transfer learning," *Phys. Rev. D, Part. Fields*, vol. 97, no. 10, May 2018, Art. no. 101501.
- [24] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, "Deep-learning continuous gravitational waves," *Phys. Rev. D, Part. Fields*, vol. 100, no. 4, Aug. 2019, Art. no. 044009.
- [25] P. G. Krastev, "Real-time detection of gravitational waves from binary neutron stars using artificial neural networks," *Phys. Lett. B*, vol. 803, Apr. 2020, Art. no. 135330.
- [26] M. B. Schäfer, F. Ohme, and A. H. Nitz, "Detection of gravitational-wave signals from binary neutron star mergers using machine learning," *Phys. Rev. D, Part. Fields*, vol. 102, no. 6, Sep. 2020, Art. no. 063015.
- [27] M. L. Chan, I. S. Heng, and C. Messenger, "Detection and classification of supernova gravitational wave signals: A deep learning approach," *Phys. Rev. D, Part. Fields*, vol. 102, no. 4, Aug. 2020, Art. no. 043022.
- [28] P. Astone, P. Cerdá-Durán, I. Di Palma, M. Drago, F. Muciaccia, C. Palomba, and F. Ricci, "New method to observe gravitational waves emitted by core collapse supernovae," *Phys. Rev. D, Part. Fields*, vol. 98, no. 12, Dec. 2018, Art. no. 122002.
- [29] A. L. Miller, P. Astone, S. D'Antonio, S. Frasca, G. Intini, I. La Rosa, P. Leaci, S. Mastrogianni, F. Muciaccia, A. Mitidis, C. Palomba, O. J. Piccinni, A. Singhal, B. F. Whiting, and L. Rei, "How effective is machine learning to detect long transient gravitational waves from neutron stars in a real search?" *Phys. Rev. D, Part. Fields*, vol. 100, no. 6, Sep. 2019, Art. no. 062005.
- [30] A. Iess, E. Cuoco, F. Morawski, and J. Powell, "Core-collapse supernova gravitational-wave search and deep learning classification," *Mach. Learn., Sci. Technol.*, vol. 1, no. 2, Jun. 2020, Art. no. 025014.
- [31] T. Gebhard, N. Kilbertus, G. Parascandolo, I. Harry, and B. Schölkopf, "CONVWAVE: Searching for gravitational waves with fully convolutional neural nets," in *Proc. Workshop Deep Learn. Phys. Sci. (DLPS), 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2017, pp. 1–6.
- [32] A. Rebei, E. A. Huerta, S. Wang, S. Habib, R. Haas, D. Johnson, and D. George, "Fusing numerical relativity and deep learning to detect higher-order multipole waveforms from eccentric binary black hole mergers," *Phys. Rev. D, Part. Fields*, vol. 100, no. 4, Aug. 2019, Art. no. 044025.
- [33] H. Wang, S. Wu, Z. Cao, X. Liu, and J.-Y. Zhu, "Gravitational-wave signal recognition of LIGO data by deep learning," *Phys. Rev. D, Part. Fields*, vol. 101, no. 10, May 2020, Art. no. 104003.
- [34] D. George and E. Huerta, "Deep neural networks to enable real-time multimessenger astrophysics," *Phys. Rev. D, Part. Fields*, vol. 97, no. 4, Feb. 2018, Art. no. 044039.
- [35] D. George and E. A. Huerta, "Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced LIGO data," *Phys. Lett. B*, vol. 778, pp. 64–70, Mar. 2018.
- [36] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, "Applying deep neural networks to the detection and space parameter estimation of compact binary coalescence with a network of gravitational wave detectors," *Sci. China Phys., Mech. Astron.*, vol. 62, no. 6, pp. 1–8, Jun. 2019.
- [37] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, "Matching matched filtering with deep networks for gravitational-wave astronomy," *Phys. Rev. Lett.*, vol. 120, no. 14, Apr. 2018, Art. no. 141103.
- [38] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, "Convolutional neural networks: A magic bullet for gravitational-wave detection?" *Phys. Rev. D, Part. Fields*, vol. 100, no. 6, Sep. 2019, Art. no. 063015.
- [39] A. Menéndez-Vázquez, M. Kolstein, M. Martínez, and L. M. Mir, "Searches for compact binary coalescence events using neural networks in the LIGO/virgo second observation period," *Phys. Rev. D, Part. Fields*, vol. 103, no. 6, Mar. 2021, Art. no. 062004.
- [40] G. Li, W. Yan, and Z. Wu, "Discovering shapelets with key points in time series classification," *Expert Syst. Appl.*, vol. 132, pp. 76–86, Oct. 2019.
- [41] W. Yan, G. Li, Z. Wu, S. Wang, and P. S. Yu, "Extracting diverse-shapelets for early classification on time series," *World Wide Web*, vol. 23, no. 6, pp. 3055–3081, Nov. 2020.
- [42] B. Bai, G. Li, S. Wang, Z. Wu, and W. Yan, "Time series classification based on multi-feature dictionary representation and ensemble learning," *Expert Syst. Appl.*, vol. 169, May 2021, Art. no. 114162.
- [43] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Müller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019.
- [44] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.
- [45] W. Jiang, "Time series classification: Nearest neighbor versus deep learning models," *Social Neww. Appl. Sci.*, vol. 2, no. 4, p. 721, Mar. 2020.
- [46] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Müller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining Knowl. Discovery*, vol. 34, no. 6, pp. 1936–1962, Nov. 2020.
- [47] J. Zhou, X. Zhang, and Z. Jiang, "Recognition of imbalanced epileptic EEG signals by a graph-based extreme learning machine," *Wireless Commun. Mobile Comput.*, vol. 2021, Jun. 2021, Art. no. 5871684.
- [48] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Neww.*, vol. 116, pp. 237–245, Aug. 2019.
- [49] F. Karim, S. Majumdar, and H. Darabi, "Insights into LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 7, pp. 67718–67725, 2019.
- [50] J. Huang, B. Chen, B. Yao, and W. He, "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network," *IEEE Access*, vol. 7, pp. 92871–92880, 2019.
- [51] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 1089–1093.
- [52] G. Ruffini, D. Ibañez, M. Castellano, L. Dubreuil-Vall, A. Soria-Frisch, R. Postuma, J.-F. Gagnon, and J. Montplaisir, "Deep learning with EEG spectrograms in rapid eye movement behavior disorder," *Frontiers Neurol.*, vol. 10, p. 806, Jul. 2019.
- [53] L. Yuan and J. Cao, "Patients' EEG data analysis via spectrogram image with a convolution neural network," in *Proc. Int. Conf. Intell. Decis. Technol. (IDT)*, vol. 72, Cham, Switzerland: Springer, 2017, pp. 13–21.
- [54] T. D. Gebhard and N. Kilbertus, "timothygebhard/ggwd: Version 1.0," Zenodo, Apr. 2019, doi: [10.5281/zenodo.2649359](https://doi.org/10.5281/zenodo.2649359).
- [55] B. P. Abbott, "GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and virgo during the first and second observing runs," *Phys. Rev. X*, vol. 9, Sep. 2019, Art. no. 031040.
- [56] *The O2 Data Release*, LIGO Sci. Collaboration, Virgo interferometer, Italy, Feb. 2019, doi: [10.7935/CA75-FM95](https://doi.org/10.7935/CA75-FM95).
- [57] R. Abbott *et al.*, "Open data from the first and second observing runs of advanced LIGO and advanced virgo," *SoftwareX*, vol. 13, Jan. 2021, Art. no. 100658.
- [58] *LIGO Algorithm Library—LALSuite, Free Software (GPL)*, LIGO Sci. Collaboration, Virgo interferometer, Italy, 2018, doi: [10.7935/GT1W-FZ16](https://doi.org/10.7935/GT1W-FZ16).
- [59] A. H. Nitz *et al.*, "gwastro/pycbc: PyCBC release v1.13.6," Zenodo, Apr. 2019, doi: [10.5281/zenodo.2643618](https://doi.org/10.5281/zenodo.2643618).
- [60] A. Bohé, L. Shao, A. Taracchini, A. Buonanno, S. Babak, I. W. Harry, I. Hinder, S. Ossokine, M. Pürrer, V. Raymond, T. Chu, H. Fong, P. Kumar, H. P. Pfeiffer, M. Boyle, D. A. Hemberger, L. E. Kidder, G. Lovelace, M. A. Scheel, and B. Szilágyi, "Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors," *Phys. Rev. D, Part. Fields*, vol. 95, no. 4, Feb. 2017, Art. no. 044028.
- [61] B. Allen, W. G. Anderson, P. R. Brady, D. A. Brown, and J. D. E. Creighton, "FINDCHIRP: An algorithm for detection of gravitational waves from inspiraling compact binaries," *Phys. Rev. D, Part. Fields*, vol. 85, no. 12, Jun. 2012, Art. no. 122006.
- [62] C. Cutler and É. E. Flanagan, "Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral waveform?" *Phys. Rev. D, Part. Fields*, vol. 49, no. 6, pp. 2658–2697, Mar. 1994.
- [63] R. J. E. Smith, I. Mandel, and A. Vecchio, "Studies of waveform requirements for intermediate mass-ratio coalescence searches with advanced gravitational-wave detectors," *Phys. Rev. D, Part. Fields*, vol. 88, no. 4, Aug. 2013, Art. no. 044010.
- [64] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya, C. Affeldt, and M. Agathos, "A guide to LIGO–virgo detector noise and extraction of transient gravitational-wave signals," *Classical Quantum Gravity*, vol. 37, no. 5, Feb. 2020, Art. no. 055002.
- [65] P. D. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Trans. Audio Electroacoust.*, vol. AU-15, no. 2, pp. 70–73, Jun. 1967.
- [66] L. Cohen, *Time-Frequency Analysis*. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.

- [67] B. Boashash, *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*. New York, NY, USA: Academic, 2015.
- [68] J. B. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-25, no. 3, pp. 235–238, Jun. 1977.
- [69] K. Samiee, P. Kovács, and M. Gabbouj, "Epileptic seizure classification of EEG time-series using rational discrete short-time Fourier transform," *IEEE Trans. Biomed. Eng.*, vol. 62, no. 2, pp. 541–552, Feb. 2015.
- [70] M. H. Ackroyd, "Short-time spectra and time-frequency energy distributions," *J. Acoust. Soc. Amer.*, vol. 50, no. 5A, pp. 1229–1231, Nov. 1971.
- [71] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," *IEEE J. Biomed. Health Inform.*, vol. 21, no. 1, pp. 56–64, Jan. 2017.
- [72] J. Ville, "Theorie et application de la notion de signal analytique," *Câbles Transmissions*, vol. 2, no. 1, pp. 61–74, 1948.
- [73] B. Boashash, "Note on the use of the Wigner distribution for time-frequency signal analysis," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 36, no. 9, pp. 1518–1521, Sep. 1988.
- [74] P. Huang, G. Liao, Z. Yang, X. G. Xia, J. Ma, and X. Zhang, "A fast SAR imaging method for ground moving target using a second-order WVD transform," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 4, pp. 1940–1956, Apr. 2016.
- [75] T. Claassen and W. Mecklenbräuer, "The Wigner distribution—A tool for time-frequency signal analysis," *Philips J. Res.*, vol. 35, no. 6, pp. 217–250, 1980.
- [76] L. Yang, L. Zhao, Z. Song, and G. Bi, "Sparsity-driven SAR imaging for highly maneuvering ground target by the combination of time-frequency analysis and parametric Bayesian learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 4, pp. 1443–1455, Apr. 2017.
- [77] P. Flandrin, "Some features of time-frequency representations of multicomponent signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, vol. 9, Mar. 1984, pp. 266–269.
- [78] A. K. Marnerides, P. Smith, A. Schaeffer-Filho, and A. Mauthe, "Power consumption profiling using energy time-frequency distributions in smart grids," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 46–49, Jan. 2015.
- [79] H.-I. Choi and W. J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 6, pp. 862–871, Jun. 1989.
- [80] F. Hlawatsch, T. G. Manickam, R. L. Urbanke, and W. Jones, "Smoothed pseudo-wigner distribution, Choi-Williams distribution, and cone-kernel representation: Ambiguity-domain analysis and experimental comparison," *Signal Process.*, vol. 43, no. 2, pp. 149–168, May 1995.
- [81] Y. Yu, L. Luo, B. Li, K. Soga, and J. Yan, "Quadratic time-frequency transforms-based Brillouin optical time-domain reflectometry," *IEEE Sensors J.*, vol. 17, no. 20, pp. 6622–6626, Oct. 2017.
- [82] A. Papandreou and G. F. Boudreaux-Bertels, "Generalization of the Choi-Williams distribution and the Butterworth distribution for time-frequency analysis," *IEEE Trans. Signal Process.*, vol. 41, no. 1, pp. 463–472, Jan. 1993.
- [83] E. Sejdić, I. Djurović, and J. Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digit. Signal Process.*, vol. 19, no. 1, pp. 153–183, Jan. 2009.
- [84] L. Cohen, "Generalized phase-space distribution functions," *J. Math. Phys.*, vol. 7, no. 5, pp. 781–786, May 1966.
- [85] J. Jeong and W. J. Williams, "Kernel design for reduced interference distributions," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 402–412, Feb. 1992.
- [86] E. Cordero, M. de Gosson, and F. Nicola, "On the reduction of the interferences in the Born-Jordan distribution," *Appl. Comput. Harmon. Anal.*, vol. 44, no. 2, pp. 230–245, Mar. 2018.
- [87] Y. Zhao, L. E. Atlas, and R. J. Marks, II, "The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 7, pp. 1084–1091, Jul. 1990.
- [88] L. Stankovic, M. Daković, and T. Thayaparan, *Time-Frequency Signal Analysis With Applications*. Boston, MA, USA: Artech House, 2013.
- [89] Z. Guo, L.-G. Durand, and H. C. Lee, "The time-frequency distributions of nonstationary signals based on a Bessel kernel," *IEEE Trans. Signal Process.*, vol. 42, no. 7, pp. 1700–1707, Jul. 1994.
- [90] W. J. Williams and J. Jeong, "Reduced interference time-frequency distributions," in *Time-Frequency Signal Analysis: Methods and Applications*, B. Boashash, Ed. New York, NY, USA: Longman-Cheshire, 1992, ch. 3, pp. 74–97.
- [91] F. Auger, P. Flandrin, P. Gonçalvès, and O. Lemoine, *Time-Frequency Toolbox*. Paris, France: CNRS/Rice Univ., 1996.
- [92] A. Buja, W. Stuetzle, and Y. Shen, "Loss functions for binary class probability estimation and classification: Structure and applications," Univ. Pennsylvania, Philadelphia, PA, USA, Tech. Rep., Nov. 2005.
- [93] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [94] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, vol. 48, 2016, pp. 507–516.
- [95] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Dec. 2016, pp. 770–778.
- [96] Z. Wu, C. Shen, and A. Van Den Hengel, "Wider or deeper: Revisiting the ResNet model for visual recognition," *Pattern Recognit.*, vol. 90, pp. 119–133, Jun. 2019.
- [97] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
- [98] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9.
- [99] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, May 2019, pp. 6105–6114.
- [100] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.
- [101] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, York, U.K.: BMVA Press, Sep. 2016, pp. 87.1–87.12.
- [102] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jun. 2017, pp. 2117–2125.
- [103] S. Ruder, "An overview of gradient descent optimization algorithms," 2016, *arXiv:1609.04747*.
- [104] B. S. Everitt, *The Analysis of Contingency Tables*, 2nd ed. London, U.K.: Chapman & Hall, 1992.
- [105] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Comput.*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998.



NIKOLA LOPAC received the B.S. and M.S. degrees (Hons.) in electrical engineering from the Faculty of Engineering, University of Rijeka, Rijeka, Croatia, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering.

From 2017 to 2020, he was a Research and Teaching Assistant with the Department of Automation and Electronics, Faculty of Engineering, University of Rijeka, where he has been a Research and Teaching Assistant with the Department of Electrical Engineering, Automation and Computing, Faculty of Maritime Studies, since 2020. Since 2020, he has also been a member of the research group within the Laboratory for Information Processing and Pattern Recognition, Centre for Artificial Intelligence and Cybersecurity, University of Rijeka. His research interests include statistical signal processing, time-frequency signal analysis, and signal processing and deep learning applications.

Mr. Lopac received the award for research excellence from the Faculty of Maritime Studies, University of Rijeka, the Annual *Hrvoje Požar* Award from the Foundation of the Croatian Energy Association, the Rector's Award from the University of Rijeka, and five Annual Dean's awards from the Faculty of Engineering, University of Rijeka.



FRANKO HRŽIĆ received the B.S. and M.S. degrees in computer engineering from the Faculty of Engineering, University of Rijeka, Rijeka, Croatia, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree in computer science, focusing on artificial intelligence (AI).

Since 2017, he has been a Research and Teaching Assistant with the Department of Computer Engineering, Faculty of Engineering, University of Rijeka. Since 2020, he has been a member of the Laboratory for Artificial Intelligence in Medicine and Biotechnology, Centre for Artificial Intelligence and Cybersecurity, University of Rijeka. His research interests include applications of deep learning methods in medical image analysis and explainable AI in medicine.



IRENA PETRIJEVČANIN VUKSANOVIĆ received the interdisciplinary Ph.D. degree from the University of Zagreb, Croatia, in 2008.

She is an Assistant Professor with the University of Osijek, Croatia. She obtained a first-class degree, scholarships, and a rector's award for editing a student magazine during her study. Currently, she is the State Secretary at the Ministry of the Interior, Croatia. She collaborates with the Military Academy in Zagreb and other universities lecturing on national security, cybersecurity, U.S. and European security issues, ICT, electronic media, and Internet legislation in the EU and Croatia.

Assoc. Prof. Petrijevcćanin Vuksanović was a member of the Croatian Parliament, the Chairperson of the Education, Science and Culture Committee, the Deputy Chairperson of the Gender Equality Committee, and the Chairperson of Parliamentary Friendship Group with the USA. She was also a member of the European Affairs Committee; the Committee for the Family, Youth and Sports; and the Committee for Information, Computerization and the Media, from 2014 to 2020. In addition, she was an ICT Advisor at the National and University Library in Zagreb, Croatia. Also, she was a member of the Council for the Electronic Media and the Head of the Council for Implementation of Digital Television (DVB-T), Croatia. She initiated establishing the Center for OpenSource, Croatia.



JONATAN LERGA received the Ph.D. degree in electrical engineering from the Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia, in 2011.

He is the Head of the Department of Computer Engineering and the Head of the Information Processing Laboratory, Faculty of Engineering, University of Rijeka, Rijeka, Croatia, where he has been with the Faculty of Engineering, University of Rijeka, since 2007. His main research interests include statistical signal and image processing, time–frequency signal analysis, information theory, coding, and signal processing applications.

Prof. Lerga received the Annual Award of the Croatian Academy of Engineering for his scientific achievements, in 2012. He also received the Annual Award of the City of Rijeka, in 2015, and the Primorje-Gorski Kotar County, in 2018. Also, he received awards from the Foundation of the University of Rijeka, in 2008, 2010, and 2018.

• • •