Detection of gravitational - wave signals from time - frequency distributions using deep learning

Lopac, Nikola

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Nikola Lopac

DETECTION OF GRAVITATIONAL-WAVE SIGNALS FROM TIME-FREQUENCY DISTRIBUTIONS USING DEEP LEARNING

DOCTORAL DISSERTATION

Rijeka, 2022.

UNIVERSITY OF RIJEKA
FACULTY OF ENGINEERING

Nikola Lopac

DETECTION OF GRAVITATIONAL-WAVE SIGNALS FROM TIME-FREQUENCY DISTRIBUTIONS USING DEEP LEARNING

DOCTORAL DISSERTATION

Thesis Supervisor: Assoc. Prof. D. Sc. Jonatan Lerga

Rijeka, 2022.

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET

Nikola Lopac

DETEKCIJA SIGNALA GRAVITACIJSKIH VALOVA IZ VREMENSKO-FREKVENCIJSKIH DISTRIBUCIJA KORIŠTENJEM DUBOKOGA UČENJA

DOKTORSKA DISERTACIJA

Mentor: izv. prof. dr. sc. Jonatan Lerga

Rijeka, 2022.

Doctoral thesis supervisor: Assoc. Prof. D. Sc. Jonatan Lerga, University of Rijeka, Faculty of Engineering, Croatia

The doctoral thesis was defended on _____ at the University of Rijeka, Faculty of Engineering, Croatia, in front of the following Evaluation Committee:

1. Prof. D. Sc. Miroslav Vrankić, University of Rijeka, Faculty of Engineering, Croatia - Committee Chair
2. Prof. D. Sc. Damir Seršić, University of Zagreb, Faculty of Electrical Engineering and Computing, Croatia
3. Assist. Prof. D. Sc. Nicoletta Saulig, Juraj Dobrila University of Pula, Faculty of Engineering, Pula, Croatia

To my mother

"And he said to all, 'If anyone would come after me,
let him deny himself
and take up his cross daily and follow me.'"

Luke 9:23

ACKNOWLEDGMENTS

First of all, I would like to express my deep and sincere gratitude to my supervisor Assoc. Prof. D. Sc. Jonatan Lerga for all the help, trust, understanding, encouragement, and helpful advice during my doctoral studies and writing this thesis. During this time, he was not just an outstanding supervisor providing continuous support and guidance, but also much more, always finding time to answer my questions at any time of day (and night). Without him, this thesis would not have been possible. I look forward to our future work together.

I would also like to thank the members of my thesis evaluation committee, Prof. D. Sc. Miroslav Vrankić, Prof. D. Sc. Damir Seršić, and Assist. Prof. D. Sc. Nicoletta Saulig, for their valuable time and suggestions.

Special thanks go to my friend and colleague Franko Hržić, mag. ing. comp., for his help and advice during the preparation of this thesis.

Finally, I owe my deepest gratitude to my family for all the support and understanding during my studies and life in general.

Author

ABSTRACT

This thesis proposes a method for classifying noisy, non-stationary signals based on deep learning algorithms and Cohen's class of time-frequency distributions (TFDs). The proposed approach is demonstrated on the challenging task of detecting gravitational-wave (GW) signals in intensive real-life, non-stationary, non-Gaussian, and non-white noise. By retrieving real-life measurements from Laser Interferometer Gravitational-Wave Observatory detectors and performing extensive GW waveform simulations, a diverse time-series dataset of 100 000 examples was obtained with the signal-to-noise ratio (SNR) in the range from -123.46 to -2.27 dB. Next, 12 TFDs were calculated from the preprocessed time series, resulting in 1.2 million TFD images, then used as input to the deep learning classification algorithms utilizing three state-of-the-art two-dimensional convolutional neural network (CNN) architectures (ResNet-101, Xception, and EfficientNet).

The results obtained by evaluating each of 36 TFD-CNN models show excellent classification performance of the proposed approach, with classification accuracy, area under the receiver operating characteristic curve (ROC AUC), recall, precision, F1 score, and area under the precision-recall curve (PR AUC) up to 97.100%, 0.98854, 95.867%, 99.507%, 97.029%, and 0.99195, respectively. Moreover, the proposed approach significantly outperforms the baseline deep learning model trained on the time-series data in terms of all considered metrics, with the statistical significance confirmed by McNemar's test.

The obtained results indicate that the proposed technique can improve the classification of non-stationary GW signals at very low SNRs with the potentials to be extended to other practical applications.

Keywords: Non-stationary signals, noisy signals, time-frequency signal analysis, Cohen's class of time-frequency distributions, deep learning, convolutional neural networks, gravitational waves.

PROŠIRENI SAŽETAK

Analiza nestacionarnih signala predstavlja izazovan zadatak u različitim istraživačkim područjima zbog vremenski promjenjivog frekvencijskog spektra takvih signala. Pritom njihova analiza često zahtijeva napredne alate za istovremeni prikaz signala u zajedničkoj vremensko-frekvencijskoj domeni, a koji nadilaze standardne tehnike zasebne analize signala u vremenskoj, odnosno frekvencijskoj domeni. Osim toga, nestacionarni su signali u stvarnim primjenama često višekomponentni, kao i dodatno narušeni šumom.

U sklopu ove doktorske disertacije predložena je i razvijena metoda za klasifikaciju nestacionarnih signala u intenzivnom šumu temeljena na algoritmima dubokoga učenja i dvodimenzionalnim vremensko-frekvencijskim distribucijama iz Cohenove klase. Ove kvadratne vremensko-frekvencijske distribucije posjeduju svojstvo vremenske i frekvencijske kovarijantnosti, a predložena metoda demonstrirana je na zahtjevnom problemu detekcije signala gravitacijskih valova u intenzivnom, stvarnom i nestacionarnom šumu koji pritom nema karakteristike ni bijelog ni Gaussovog šuma.

Predloženi je pristup eksperimentalno provjeren, a razvijeni se postupak sastoji od triju glavnih faza: pripreme i predobrade odgovarajućeg skupa podataka, treniranja i testiranja modela dubokoga učenja te evaluacije postignutih performansi navedenih modela. Korištenjem stvarnih mjerenja iz *Laser Interferometer Gravitational-Wave Observatory* (LIGO) detektora i provođenjem iscrpnih simulacija valnih oblika gravitacijskih valova dobiven je opsežan i raznolik skup podataka koji uključuje 100 000 primjera podataka u vremenskoj domeni. Pritom se vrijednosti omjera signala i šuma u generiranomu skupu podataka kreću u rasponu između -123.46 i -2.27 dB. Nakon odgovarajuće predobrade podataka u vremenskoj domeni izračunano je 12 vremensko-frekvencijskih distribucija iz Cohenove klase, uključujući spektrogram, Wigner-Villeovu, pseudo Wigner-Villeovu, izglađenu pseudo Wigner-Villeovu, Choi-Williamsovu, Butterworthovu, Born-Jordanovu i Zhao-Atlas-Marksovu distribuciju, te distribucije sa smanjenim interferencijama i jezgrama temeljenima na Besselovoj funkciji, binomnim koeficijentima, Hanningovom otvoru i trokutastom otvoru. Navedeni izračun rezultirao je s ukupno 1 200 000 slika vremensko-frekvencijskih distribucija, raspodijeljenima u 12 skupova podataka, a koje su potom korištene kao ulazi za treniranje i testiranje algoritama dubokoga učenja za klasifikaciju temeljenih na trima naprednim dvodimenzionalnim arhitekturama konvolucijskih neuronskih

mreža (ResNet-101, Xception i EfficientNet).

Rezultati postignuti evaluacijom svakog od 36 dobivenih modela dubokoga učenja konvolucijskih neuronskih mreža i vremensko-frekvencijskih distribucija pokazuju izvrsne klasifikacijske performanse predloženoga pristupa. Pritom točnost klasifikacije postiže vrijednosti do 97.100%, površina ispod krivulje značajke djelovanja prijamnika do 0.98854, odziv do 95.867%, preciznost do 99.507%, F1-mjera do 97.029% i površina ispod krivulje preciznost-odziv do 0.99195. Osim toga, usporedba ostvarenih performansi predloženoga pristupa s performansama referentnoga modela dubokoga učenja, koji kao ulaze koristi izvorne podatke u vremenskoj domeni, pokazuje da predloženi pristup značajno nadmašuje referentni model s obzirom na sve korištene pokazatelje kvalitete performansi. Naime, postignute vrijednosti točnosti klasifikacije više su do 3.953%, površine ispod krivulje značajke djelovanja prijamnika do 2.067%, odziva do 7.014%, preciznosti do 2.307%, F1-mjere do 4.190% i površine ispod krivulje preciznost-odziv do 1.475%. Analiza dodatnih detaljnih pokazatelja kvalitete, uključujući matrice konfuzije, krivulje značajke djelovanja prijamnika i krivulje preciznost-odziv, također ukazuje na bolje performanse predloženoga pristupa, pri čemu je statistička značajnost dobivenih razlika u performansama potvrđena McNemarovim statističkim testom.

Analiza dobivenih rezultata ukazuje na to da predloženi pristup primjene kvadratnih vremensko-frekvencijskih distribucija iz Cohenove klase u kombinaciji s algoritmima dubokoga učenja temeljenima na dvodimenzionalnim konvolucijskim neuronskim mrežama može postići poboljšanu kvalitetu klasifikacije nestacionarnih signala u vremenskoj domeni u uvjetima vrlo niskih vrijednosti omjera signala i šuma. U sklopu ove doktorske disertacije analizirana je i potvrđena mogućnost praktične primjene predloženoga pristupa u detekciji signala gravitacijskih valova, pri čemu su postignute vrlo visoke performanse. Osim navedene primjene, predloženi pristup također ima potencijal za proširenje na druga područja znanstvenog istraživanja i praktične primjene koje zahtijevaju analizu različitih vrsta nestacionarnih signala.

Ključne riječi: Nestacionarni signali, signali u šumu, vremensko-frekvencijska analiza signala, Cohenova klasa vremensko-frekvencijskih distribucija, duboko učenje, konvolucijske neuronske mreže, gravitacijski valovi.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

This chapter provides introductory information on the research presented in this thesis. The introductory part describes the scientific motivation, problem formulation, and a brief note on the previous research. Next, the research hypotheses and the main objectives and contributions of the thesis are outlined. Furthermore, the methodology of the conducted research and the overview of the thesis structure are briefly reviewed.

1.1 Scientific Motivation

Time-domain signal representation, showing signal amplitude as a function of time, is commonly used for signal analysis as it reflects the mode in which the most signal measurements are acquired. Another commonly used signal representation is Fourier transform-based frequency-domain representation that provides information about the signal's frequency content without information on its time localization. Thus, the time-domain and frequency-domain representations are standard signal analysis approaches suitable for analyzing stationary signals characterized by a frequency spectrum constant in time.

However, these one-dimensional representations are not adequate for analyzing non-stationary signals, i.e., signals with time-varying frequency content. In addition, non-stationary signals in practical applications are often multi-component and corrupted by noise, which further complicates their analysis. Therefore, non-stationary signal analysis requires advanced and robust tools for the simultaneous signal representation in the joint time-frequency domain, leading to the development of two-dimensional time-frequency distributions (TFDs) [36, 64, 207]. TFDs, as more informative signal representations, provide insight into the signal energy distribution as a function of both time and frequency.

TFDs can be divided into two main categories: linear and quadratic distributions. The quadratic TFDs with the common properties of time and frequency covariance are also known under the name of Cohen's class of distributions, exhibiting many useful

mathematical properties for non-stationary signal analysis [36, 64, 119].

On the other side, machine learning has recently received rapidly increasing interest in different research fields and practical applications. This term encompasses different computational algorithms with the ability to learn from provided data examples [136]. Deep learning, as a subfield of machine learning, includes algorithms utilizing deep artificial neural networks (ANNs) that can automatically learn from the provided input data without the need for the previously extracted hand-crafted data features [103, 150]. In recent years, these algorithms have experienced intensive development and expansion to different application areas. The special type of feedforward ANNs called convolutional neural networks (CNNs) have achieved state-of-the-art performances in many fields, especially in image recognition, classification, and segmentation applications [103, 150].

However, despite the above-mentioned beneficial properties of Cohen's class of TFDs for non-stationary signal representation, the utilization of different Cohen's class TFDs with the advanced, powerful CNN-based deep learning algorithms used for pattern recognition in case of intensive noise has not been adequately studied in detail in the literature (especially applied in the field of physics), to the best of author's knowledge based on the extensive literature review. Namely, deep CNNs are primarily applied to classify time-series signals [86, 137, 203, 268] or their spectrogram representations [27, 270, 277, 281]. On the other hand, TFDs from Cohen's class have mainly been used only to extract characteristic features from the corresponding two-dimensional signal representations, which are then used as one-dimensional input to the conventional machine learning algorithms [39, 93, 111, 189, 193] or feedforward ANNs [20, 171, 210, 241, 256, 259, 260, 285] for classification purposes. Nevertheless, selecting the hand-crafted features to be extracted from TFDs is time-consuming and requires domain-specific expertise. Thus, these approaches fail to take advantage of the CNN-based deep learning algorithms' high performance on two-dimensional input data and their ability to automatically learn and extract useful features.

Accordingly, this thesis proposes and thoroughly analyzes an approach for classifying non-stationary signals in intensive noise utilizing various TFDs from Cohen's class combined with the deep learning classification algorithms based on the different state-of-the-art two-dimensional CNNs. The proposed approach is validated on the example of detecting gravitational-wave (GW) signals in intensive noise.

Detection of GW signals was chosen in this thesis to demonstrate the usefulness and practical applicability of the proposed approach since it is a challenging task that is in the focus of many recent studies and research efforts. Namely, the measurements containing these non-stationary GW signals include non-stationary, non-Gaussian, and non-white noise [5]. Moreover, the real-life GW data are characterized by very low signal-to-noise ratio (SNR) values due to their low amplitudes compared to the background noise, rendering GW detection a demanding data analysis problem.

The existing approaches to GW detection may be divided into three main categories: matched filtering-based techniques, denoising techniques, and machine learning-based techniques. However, deep learning techniques have been mostly utilized to classify time-series GW signals [85, 91, 96, 97, 98] or their spectrogram representations [22, 130, 184], while the utilization of other two-dimensional signal representations has not been adequately addressed in the literature [67]. Moreover, after an extensive literature review, no approach combining deep learning techniques with Cohen's class of TFDs for GW detection has been found. A more detailed literature review of the previous research on Cohen's class of TFDs, deep learning techniques, GW detection, and deep learning-based signal classification, with a particular focus on the potential application of Cohen's class of TFDs, is provided in the following chapters.

Next, the concrete objectives and contributions of this thesis are defined.

1.2 Research Objectives and Scientific Contributions of the Thesis

Based on the scientific motivation discussed above, the main objective of the research presented in this thesis is to develop a method for classifying non-stationary signals in intensive noise using two-dimensional, quadratic TFDs from Cohen's class of distributions and advanced deep learning algorithms based on two-dimensional CNN architectures. Within this thesis, the performance of the proposed approach is demonstrated on the example of detecting GW signals in real-life, non-stationary, non-white, and non-Gaussian noise, which represents a challenging task due to the very low SNR values.

The first hypothesis of the conducted research is that TFDs from Cohen's class combined with advanced deep learning techniques can provide high-performance detection of non-stationary signals in an intensive noise environment. The second research hypothesis assumes that two-dimensional, quadratic TFDs from Cohen's class, when combined with two-dimensional deep CNN architectures, can achieve higher signal classification performances than the CNN-based deep learning classification utilizing only the original noisy time-series signals. This performance improvement is expected since these high-resolution TFDs can provide a signal representation with better-structured information and enhanced intelligibility compared to the time-series signal representations, which can then be efficiently utilized by two-dimensional deep CNNs.

The main scientific contribution of the research conducted within this thesis consists of a developed approach for classifying non-stationary signals heavily corrupted by noise utilizing their TFDs from Cohen's class and deep learning algorithms based on state-of-the-art two-dimensional CNN architectures. The proposed approach is within this thesis validated on the example of GW signal detection, achieving excellent performances and

demonstrating its practical applicability in this field where it can further contribute to improved GW detection rates. However, besides the application in GW data analysis, the proposed approach also has a great potential for many other practical applications in various fields dealing with non-stationary signal analysis.

The research methodology utilized in developing and validating the proposed approach is described in the sequel.

1.3 Research Methodology

The proposed approach to detecting non-stationary signals in intensive noise was developed by combining beneficial properties of Cohen's class of TFDs as powerful tools for non-stationary signal analysis with the state-of-the-art deep CNNs for pattern recognition. The developed technique was experimentally validated through the process involving three main stages: data generation and preprocessing, deep learning classification, and performance evaluation.

The time-series dataset containing 100 000 different data examples was generated by simulating GW waveforms and utilizing the real-life data from the Advanced Laser Interferometer Gravitational-Wave Observatory (LIGO) detectors as background noise. The raw time-series data examples within this dataset were characterized by very low SNR values. After preprocessing the raw time-series data, 12 different TFDs from Cohen's class were computed, obtaining 12 TFD datasets, with each dataset containing 100 000 TFD images (1.2 million TFD images in total).

Each TFD dataset was appropriately divided into three subsets: training, validation, and test dataset. Next, three two-dimensional deep CNN architectures, including ResNet-101, Xception, and EfficientNet, were trained and used to classify the TFD images from 12 datasets. Thus, a total of 36 TFD-CNN deep learning models was tested.

The performance of the considered deep learning models was thoroughly evaluated using several evaluation metrics. Moreover, the obtained performance was compared to the one obtained by the baseline deep learning model using CNN to classify the time-series data. Finally, the classification improvement achieved by the proposed technique was also supported by statistical significance results examined using McNemar's test.

1.4 Organization of the Thesis

This thesis is organized into seven chapters and two appendices, providing a structured presentation of the conducted research. A short outline of the thesis chapters is given next.

Chapter 1 discusses the scientific motivation of the presented research, outlines the main

objectives and contributions of the thesis, and briefly describes the research methodology. Moreover, this chapter also gives a brief overview of the thesis structure.

The main concepts of non-stationary signal analysis, with a particular focus on the benefits of time-frequency representation, are introduced in Chapter 2. The quadratic TFDs of Cohen's class are defined, and their mathematical properties are discussed. The concept is also demonstrated in the non-stationary signal examples.

Chapter 3 provides a brief overview of the deep learning field and explains the main concepts of ANNs, with special emphasis on CNNs. The chapter additionally reviews the recent studies on deep learning-based signal classification, particularly focusing on the potential utilization of Cohen's class of TFDs.

Chapter 4 introduces the main concepts and challenges of GW research and provides a literature review of the existing approaches to GW detection, divided into three main categories: matched filtering-based techniques, denoising techniques, and machine learning-based techniques.

Chapter 5 represents the central part of the thesis, presenting the proposed method for detecting GW signals in intensive noise based on deep learning algorithms and Cohen's class of TFDs. The developed approach and the experimental setup are elaborated on in detail, including the data generation procedure, TFD computations, input datasets, deep learning models, and performance evaluation methodology based on the evaluation metrics and statistical significance test.

The results obtained by the proposed method are presented in Chapter 6, accompanied by a detailed analysis and discussion.

Finally, the main findings and conclusions are summarized in Chapter 7. Next, based on the research presented in this thesis and the obtained results, the directions for future work are addressed.

CHAPTER 2

TIME-FREQUENCY SIGNAL ANALYSIS

This chapter explains the main concepts of non-stationary signal analysis and emphasizes the importance of the joint time-frequency signal representations. The TFDs are introduced by defining the mathematical framework and reviewing the main properties, with a particular focus on the quadratic distributions from Cohen's class. All introduced concepts of time-frequency signal analysis are also demonstrated on the examples of the multi-component non-stationary signals.

2.1 Non-Stationary Signals

The time-domain representation of the signal $s(t)$ presents the signal amplitude as a function of time t. The time-domain representation is a commonly used signal representation because it represents the way most natural phenomena signals are measured. In addition, the squared signal amplitude $|s(t)|^2$ represents the instantaneous power of the signal, which provides information on the distribution of signal energy over time.

On the other hand, the frequency-domain representation provides insight into the frequency components contained in the signal. The spectral analysis of the signal is based on the Fourier transform $\mathcal{F}$, which is for the signal $s(t)$ defined as

$$S(f) = \underset{t \to f}{\mathcal{F}} \{s(t)\} = \int_{-\infty}^{\infty} s\,(t)\, e^{-j2\pi ft} dt, \tag{2.1}$$

where f denotes the frequency.

The complex Fourier transform consists of the magnitude spectrum and the phase spectrum. The squared magnitude spectrum $|S(f)|^2$ represents the signal energy spectrum that shows the distribution of the signal energy over the frequency domain.

The described time-domain representation and frequency-domain representation are

standard tools for one-dimensional signal analysis. These tools are appropriate for analyzing stationary signals, i.e., signals that have a constant frequency spectrum. However, non-stationary signals are characterized by a time-varying frequency spectrum, which makes the separate analysis of the traditional representations in the time and frequency domain insufficient.

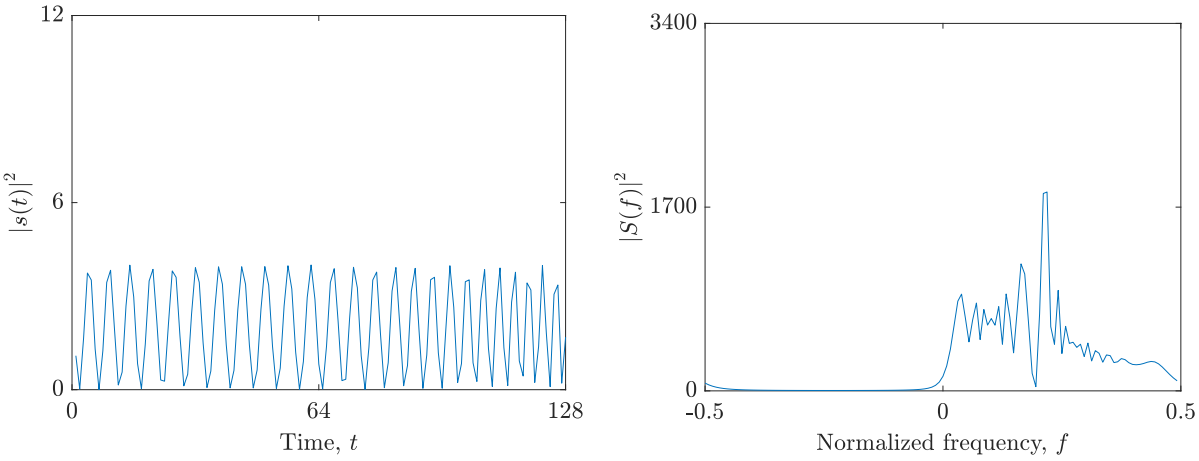
Moreover, analyzing the time-domain representation of the signal whose frequency content changes with time represents a challenging task. Likewise, the frequency-domain representation offers information on the frequency components contained in the signal but does not provide any information on their time localization.

Furthermore, non-stationary signals are also often multi-component, and their components may have different, linear or non-linear, frequency modulations with different time supports. The classical approaches that treat the signal representations in the time and frequency domain separately cannot provide information on the number of signal components, the frequency modulations, or time supports.

These drawbacks of the one-dimensional signal representations are demonstrated here in the example of the synthetic multi-component non-stationary signal with two components. The first signal component in the example has a parabolic frequency modulation with the normalized frequency values in the range $0.2 - 0.5$, while the normalized frequency of the second component changes linearly in the range between 0.0 and 0.2 (linear frequency modulation (LFM)). The signal representations are provided in Figure 2.1, where Figure 2.1(a) shows the time-domain representation in the form of the instantaneous power $|s(t)|^2$, while Figure 2.1(b) depicts the frequency-domain representation in the form of the energy spectrum $|S(f)|^2$. The normalized frequency is obtained by dividing the signal frequency by the sampling frequency value.

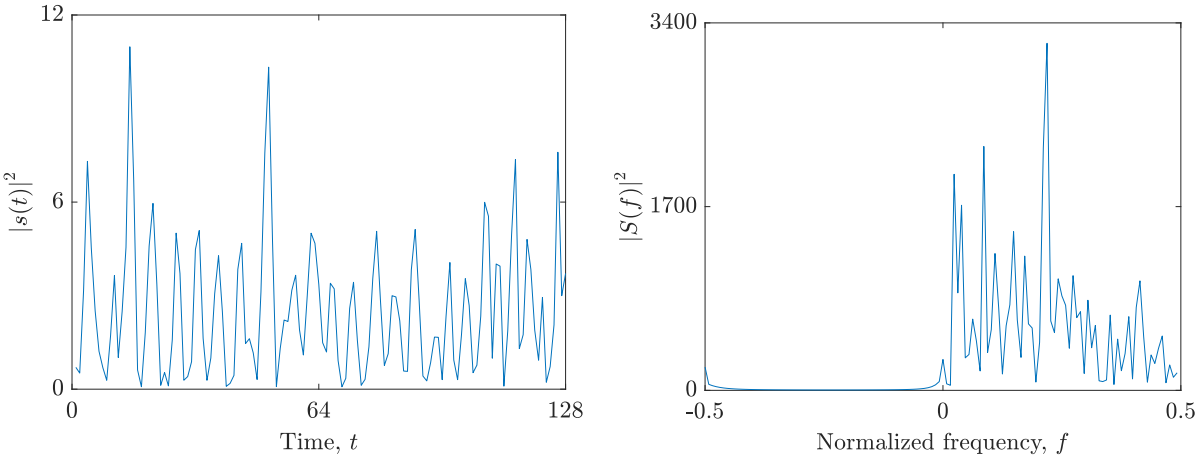
Additionally, non-stationary signals acquired in real-life applications are often corrupted by noise, complicating their analysis using standard techniques. The influence of the noise on the one-dimensional signal representations is illustrated in Figure 2.2, which provides the time-domain representation and the frequency-domain representation of the previously analyzed signal, but, in this case, the signal is additionally corrupted by the additive white Gaussian noise with a 5 dB signal-to-noise ratio (SNR). As seen in Figure 2.2(a) and 2.2(b), the presence of the noise makes the analysis of the considered non-stationary signal even more challenging, making the separate time-domain representation and the frequency-domain representation impractical tools to interpret the noisy signals with the time-varying frequency content.

Due to the above-described disadvantages of representing the signal separately in the time and frequency domain, the analysis of the non-stationary signals requires more advanced tools for the simultaneous signal representation in both the time and frequency domain. Therefore, the TFDs are developed to show the distribution of the signal energy as a function of two variables - time and frequency. This way the joint signal representations



(a) (b)

Figure 2.1 Example of a two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component: (a) Time-domain representation, $|s(t)|^2$; (b) Frequency-domain representation, $|S(f)|^2$.



(a) (b)

Figure 2.2 Example of a noisy two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component (SNR = 5 dB): (a) Time-domain representation, $|s(t)|^2$; (b) Frequency-domain representation, $|S(f)|^2$.

in the two-dimensional time-frequency plane are obtained [36, 64, 119, 207].

The ideal TFD of the multi-component non-stationary signal would show the instantaneous frequency laws of each signal component in the time-frequency plane [36]. The instantaneous frequency $f_i(t)$ describes the variation of signal frequency in time and is defined for the mono-component signal $s(t)$ as [36, 64]

$$f_i(t) = \frac{1}{2\pi}\frac{d}{dt}\left[\arg s(t)\right] = \frac{1}{2\pi}\frac{d\phi(t)}{dt}, \tag{2.2}$$

where $\phi(t)$ is the instantaneous phase of the signal.

The dual of the instantaneous frequency is the time delay that shows the dominant time when a given frequency occurs [36]. The time delay $\tau_d(f)$ of the signal $s(t)$ is defined as [36, 64]

$$\tau_d(f) = -\frac{1}{2\pi}\frac{d}{df}\left[\arg S(f)\right] = -\frac{1}{2\pi}\frac{d\theta(f)}{df}, \tag{2.3}$$

where $\theta(f)$ is the phase of the Fourier transform of the signal.

The group delay $\tau_g(f)$ of the signal $s(t)$ is defined in the same way as the time delay $\tau_d(f)$ in (2.3) [36]. The difference between these two definitions is that the time delay corresponds to an impulse, while the group delay corresponds to the envelope of a narrowband signal [36].

The ideal TFD of the considered example of the noise-free two-component non-stationary signal is shown in Figure 2.3. As seen in Figure 2.3, the ideal TFD provides clear information on the number of signal components, their instantaneous frequency laws, the frequency ranges, and time supports.

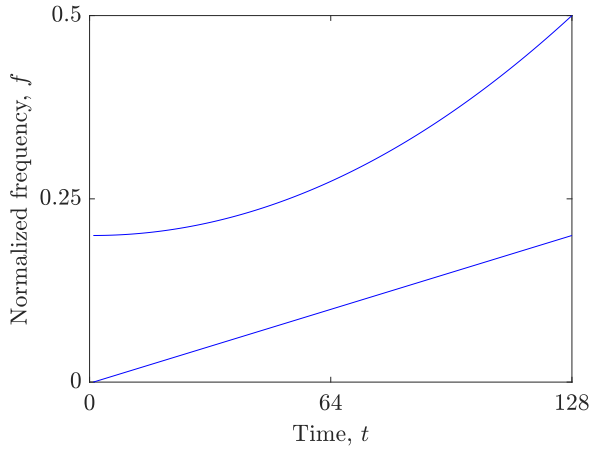


Figure 2.3 Ideal TFD of the considered example of a two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component.

The calculation of the ideal TFD is based on the formula given in (2.2), which requires knowledge of the analytic expression for the signal phase. However, the analytic expressions for the signals are not available in practical, real-life applications, thus rendering this approach to determine the instantaneous frequency unsuitable for the TFD calculation.

Therefore, various techniques for the numerical calculation of the TFDs are developed. These techniques are discussed in detail in the next section. The main requirement placed on thus obtained TFDs is to provide the two-dimensional signal representation in which the signal energy is concentrated around the instantaneous frequency laws of the signal components.

Before proceeding to define the TFD methods, the analytic signal associated with the real signal $r(t)$, or the analytic associate, needs to be defined. The analytic associate $s(t)$

is a signal that has no negative frequency components [36, 64]

$$S(f) = 0, \quad f < 0. \tag{2.4}$$

Namely, since the real signal $r(t)$ has the frequency spectrum that shows Hermitian symmetry, $R(-f) = R^*(f)$, the negative frequency components can be removed without losing information. As it will be seen in the next section dealing with the quadratic TFDs, the absence of the negative signal frequencies is a desirable property because it eliminates the cross-terms between the signal's positive and negative frequency components, thus enhancing the time-frequency representation readability and interpretation [36]. Therefore, the analytic associate form of the signals will be used throughout this thesis.

The complex-valued analytic associate $s(t)$ of the real signal $r(t)$ is obtained using the Hilbert transform $\mathcal{H}$ as [36, 64]

$$s(t) = r(t) + j\mathcal{H}\{r(t)\}. \tag{2.5}$$

The Hilbert transform is defined as [36]

$$\mathcal{H}\left\{r(t)\right\} = \underset{t \leftarrow f}{\mathcal{F}^{-1}} \left\{ (-j \operatorname{sgn}(f)) \underset{t \rightarrow f}{\mathcal{F}} \{r(t)\} \right\}, \tag{2.6}$$

where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform, j is the imaginary unit, and sgn() is the signum function, defined as

$$\operatorname{sgn}(x) = \begin{cases} -1, & x < 0, \\ 0, & x = 0, \\ 1, & x > 0. \end{cases} \tag{2.7}$$

2.2 Time-Frequency Distributions

The two-dimensional TFDs, developed as tools for the simultaneous signal analysis in both the time and frequency domain, are presented next. First, some mathematical properties of TFDs are defined.

2.2.1 Desirable Properties of Time-Frequency Distributions

The following mathematical properties of TFDs, denoted as $\rho_s(t, f)$, are often considered desirable in various practical applications [36]:

1. Non-negativity - TFD has non-negative values:

$$\rho_s(t, f) \geq 0, \quad \forall t, f. \tag{2.8}$$

2. Realness - TFD is real for all s, t, and f:

$$\rho_s(t,f) \in \mathbb{R}, \quad \forall t, f. \tag{2.9}$$

3. Time-shift invariance (time covariance) - a time shift in the signal results in the same time shift in the TFD:

$$s_{ts}(t) = s(t-t_0) \Longrightarrow \rho_{s_{ts}}(t,f) = \rho_s(t-t_0,f). \tag{2.10}$$

4. Frequency-shift invariance (frequency covariance) - a frequency shift in the signal results in the same frequency shift in the TFD:

$$s_{fs}(t) = s(t)e^{j2\pi f_0 t} \Longrightarrow \rho_{s_{fs}}(t,f) = \rho_s(t,f-f_0). \tag{2.11}$$

5. Time marginal - the instantaneous power $|s(t)|^2$ is obtained by integrating the TFD over the frequency:

$$\int_{-\infty}^{\infty} \rho_s(t,f)df = |s(t)|^2. \tag{2.12}$$

6. Frequency marginal - the energy spectrum $|S(f)|^2$ is obtained by integrating the TFD over time:

$$\int_{-\infty}^{\infty} \rho_s(t,f)dt = |S(f)|^2. \tag{2.13}$$

7. Instantaneous frequency - the instantaneous frequency $f_i(t)$ is obtained as the first moment of the TFD with respect to frequency:

$$\frac{\int_{-\infty}^{\infty} f\rho_s(t,f)df}{\int_{-\infty}^{\infty} \rho_s(t,f)df} = f_i(t) = \frac{1}{2\pi}\frac{d}{dt}\left[\arg s(t)\right]. \tag{2.14}$$

8. Group delay - the group delay $\tau_g(f)$ is obtained as the first moment of the TFD with respect to time:

$$\frac{\int_{-\infty}^{\infty} t\rho_s(t,f)dt}{\int_{-\infty}^{\infty} \rho_s(t,f)dt} = \tau_g(f) = -\frac{1}{2\pi}\frac{d}{df}\left[\arg S(f)\right]. \tag{2.15}$$

9. Time support - the duration of the signal $s(t)$ limits the time support of the TFD:

$$s(t) = 0 \text{ for } t < t_1, t > t_2 \Longrightarrow \rho_s(t,f) = 0 \text{ for } t < t_1, t > t_2. \tag{2.16}$$

10. Frequency support - the bandwidth of the signal $s(t)$ limits the frequency support of

the TFD:

$$S(f) = 0 \text{ for } f < f_1, f > f_2 \Longrightarrow \rho_s(t, f) = 0 \text{ for } f < f_1, f > f_2. \tag{2.17}$$

11. Reduced interferences - the TFD suppresses the interference terms (cross-terms) in comparison to the signal components (auto-terms).

2.2.2 Spectrogram

The first step towards the joint time-frequency signal representation is an intuitive and straightforward method called the short-time Fourier transform (STFT). This method calculates the Fourier transform of the signal $s(\tau)$ segment inside the sliding window function $h(\tau)$ centered at that time instance t for each t, thus obtaining the signal frequency spectrum as a function of time t [18, 64]. The STFT is calculated as [18]

$$STFT_s(t, f) = \int_{-\infty}^{\infty} s(\tau)\, h(\tau - t)\, e^{-j2\pi f\tau} d\tau. \tag{2.18}$$

The STFT decomposes the signal into the elementary components well localized in time and frequency [119]. As seen in (2.18), the STFT calculation is based on the signal amplitude $s(\tau)$. Therefore, the STFT belongs to the group of the linear TFDs.

In addition to the linear TFDs, another approach to the time-frequency signal representations includes the methods whose calculations exploit the quadratic dependence on the signal amplitude. This group of quadratic signal transforms represents the distributions of the signal energy in the joint time-frequency domain [64].

The transition between the linear and the quadratic group of the TFDs is obtained by the spectrogram (SP), which is calculated by taking the squared modulus of the calculated STFT [13, 89, 119]

$$SP_s(t, f) = |STFT_s(t, f)|^2 = \left| \int_{-\infty}^{\infty} s(\tau)\, h(\tau - t)\, e^{-j2\pi f\tau} d\tau \right|^2. \tag{2.19}$$

The spectrogram satisfies the following mathematical properties: non-negativity, realness, time-shift invariance, frequency-shift invariance, and reduced interferences [36, 119, 135]. Besides the relatively simple implementation, the main advantage of the SP is the signal representation with a low level of unwanted interference terms. On the other hand, despite this desirable property, the usefulness of the SP is limited due to the poor resolution property. Namely, the SP representation cannot provide a simultaneously good resolution in both the time and frequency domain [119]. The SP will always exhibit the trade-off between the time and frequency resolution caused by using the sliding window of a fixed size [36, 64, 119]. The time resolution can be increased by using smaller-size

windows, but, consequently, the frequency resolution will be decreased. On the other hand, longer windows provide better frequency resolution and lower time resolution.

Due to the above-described limitations of the SP, the alternative approaches to the TFD calculation are investigated. The work presented in this thesis focuses on the quadratic TFDs belonging to Cohen's class, characterized by the time and frequency covariance properties [36, 64, 119].

2.2.3 Wigner-Ville Distribution

The Wigner-Ville distribution (WVD) [264] represents a fundamental TFD of Cohen's class. The WVD is obtained by applying the Fourier transform to the instantaneous autocorrelation function (IAF) of the signal $s(t)$ [36]. The IAF $K_s(t,\tau)$ of the signal $s(t)$ is defined as [36]

$$K_s(t,\tau) = s\left(t+\frac{\tau}{2}\right)s^*\left(t-\frac{\tau}{2}\right), \tag{2.20}$$

where $s^*(t)$ denotes the complex conjugate of the signal $s(t)$.

Therefore, the WVD is obtained as [35]

$$WVD_s(t,f) = \int_{-\infty}^{\infty} s\left(t+\frac{\tau}{2}\right)s^*\left(t-\frac{\tau}{2}\right)e^{-j2\pi f\tau}d\tau. \tag{2.21}$$

As seen in (2.21), the representation in the time-frequency (t,f) domain is obtained by calculating the Fourier transform of the representation in the time-lag (t,τ) domain with respect to lag τ. The lag variable τ represents a time shift [36]. Another domain widely used in the design of the quadratic TFDs is the Doppler-lag (ν,τ) domain, also known as the ambiguity function domain [36]. The Doppler variable represents a frequency shift [36]. The Doppler-lag domain is defined as the Fourier transform of the time-lag domain with respect to time t, i.e., the ambiguity function $A_s(\nu,\tau)$ is obtained by applying the Fourier transform to the IAF $K_s(t,\tau)$ with respect to time t [36, 64]

$$A_s(\nu,\tau) = \int_{-\infty}^{\infty} K_s(t,\tau)e^{-j2\pi\nu t}dt. \tag{2.22}$$

When transformed to the Doppler-lag domain, the signal components exhibiting slow variation in the time-frequency domain will be located near the origin of the new domain [36]. On the other hand, the rapid-varying signal components will be located farther away from the origin of the Doppler-lag domain [36].

The WVD, defined in (2.21), satisfies almost all previously defined mathematical properties that are desirable for the TFDs, including realness, time-shift invariance, frequency-shift invariance, time marginal, frequency marginal, instantaneous frequency, group delay, time support, and frequency support [36, 61, 119, 135]. Thus, the only two properties that the WVD does not satisfy are non-negativity (as it takes negative values)

and reduced interferences [36, 119, 135]. In addition, the WVD also satisfies the following properties [36, 119]:

1. Global energy - the signal energy E_s is obtained by integrating the TFD over the entire time-frequency plane:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\rho_s(t,f)dtdf = E_s. \tag{2.23}$$

2. Convolution invariance - the TFD of the time-convolution of two signals $s_1(t)$ and $s_2(t)$ is equal to the time-convolution of the TFDs of these two signals:

$$s_3(t) = s_1(t) \underset{t}{*} s_2(t) \Longrightarrow \rho_{s_3}(t,f) = \rho_{s_1}(t,f) \underset{t}{*} \rho_{s_2}(t,f). \tag{2.24}$$

3. Modulation invariance - the TFD of the frequency-convolution of two signals $s_1(t)$ and $s_2(t)$ is equal to the frequency-convolution of the TFDs of these two signals:

$$s_3(t) = s_1(t)s_2(t) \Longrightarrow \rho_{s_3}(t,f) = \rho_{s_1}(t,f) \underset{f}{*} \rho_{s_2}(t,f). \tag{2.25}$$

4. Invertibility - the signal $s(t)$ can be reconstructed from the TFD up to a constant phase as:

$$\int_{-\infty}^{\infty}\rho_s\left(\frac{t}{2},f\right)e^{j2\pi ft}df = z(t)\ z^*(0). \tag{2.26}$$

5. Inner-product invariance (unitarity property) - the TFD preserves the inner product from the time-domain to the time-frequency domain:

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\rho_{s_1}(t,f)\rho_{s_2}(t,f)dtdf = \left|\int_{-\infty}^{\infty}s_1(t)s_2^*(t)dt\right|^2. \tag{2.27}$$

Moreover, the WVD provides a high time-frequency resolution of the true signal components. The WVD is perfectly localized in the time-frequency plane for mono-component LFM signals [36, 119]. However, the quadratic nature of the mathematical expression used to calculate the WVD is the cause of the interference terms, also known as cross-terms, that occur between the true signal components (auto-terms) in the time-frequency representation of the multi-component signals [36, 119, 120]. The cross-terms are located in the middle between the two auto-terms and oscillate orthogonally to the line connecting these auto-terms, with an oscillation rate proportional to the distance between them [36, 119, 120]. The cross-terms include the outer terms that appear between the different signal components, and the inner terms that occur due to the non-linear frequency modulation of the individual signal component [36, 120].

The cross-terms can make a visual interpretation of the TFD challenging. Therefore, in order to obtain interpretable time-frequency representations, the unwanted cross-terms need to be attenuated. The attenuation of the cross-terms is possible by smoothing the WVD using the filtering functions, also known as kernels [36, 64].

2.2.4 Reduced-Interference Distributions

The Doppler-lag domain, defined in (2.22), is convenient for the design of the TFD kernels [36]. The kernels in this domain can be designed as two-dimensional low-pass filters [36]. Namely, as previously stated, the highly oscillatory cross-terms are located away from the origin of the Doppler-lag domain and can thus be removed using the low-pass filter [36, 64, 240]. However, the low-pass filtering also removes a part of the auto-terms located around the origin, thus reducing the resolution of the signal auto-terms in the time-frequency representation [36, 265]. This filtering is causing the trade-off between the obtained time-frequency resolution of the signal auto-terms and the level of the cross-term attenuation [36, 64]. The TFDs obtained with the kernels used to smooth the WVD are referred to as the reduced-interference distributions [36, 64, 135].

Windowing the IAF by the time window $h(t)$ and applying the same procedure defined in (2.21) for the WVD, the pseudo Wigner-Ville distribution (PWVD) is obtained [61]. Multiplying by the window function in the time domain is equivalent to the smoothing of the WVD in the frequency domain [119, 120]. The frequency smoothing has two consequences: the first is the desirable attenuation of the cross-terms oscillating in the frequency direction, while the second is the unwanted decrease in the frequency resolution of the signal auto-terms [25, 119]. The PWVD is calculated as [61, 90]

$$PWVD_s(t,f) = \int_{-\infty}^{\infty} h(\tau)\; s\left(t+\frac{\tau}{2}\right) s^*\left(t-\frac{\tau}{2}\right) e^{-j2\pi f\tau} d\tau. \tag{2.28}$$

The PWVD is known to satisfy the following properties: realness, time-shift invariance, frequency-shift invariance, time marginal, global energy, instantaneous frequency, time support, and reduced interferences [36, 119].

As mentioned above, the PWVD suppresses the cross-terms oscillating in the frequency direction but does not affect those oscillating in the time direction. The issue of the remaining cross-terms is addressed by the smoothed pseudo Wigner-Ville distribution (SPWVD). This TFD additionally applies the time-smoothing window $g(t)$, thus smoothing the PWVD in the time direction [88]. Therefore, by choosing the lengths of the windows $h(t)$ and $g(t)$ in the SPWVD, the smoothing of the WVD may be adjusted independently in the time and frequency domain [88, 119, 120]. Nevertheless, there remains a trade-off between the level of the cross-terms in the representation and the obtained time-frequency resolution, which is characteristic of reduced-interference distributions [25]. The SPWVD

is defined as [88]

$$SPWVD_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \int_{-\infty}^{\infty} g(u-t)\ s\left(u+\frac{\tau}{2}\right) s^*\left(u-\frac{\tau}{2}\right) du\ e^{-j2\pi f\tau} d\tau. \quad (2.29)$$

The SPWVD is known to have the following properties: realness, time-shift invariance, frequency-shift invariance, global energy, and reduced interferences [66, 119, 121].

The Choi-Williams distribution (CWD) is a reduced-interference distribution designed with an exponential kernel [57]. The exponential kernel's width σ controls the trade-off between the attenuation of the cross-terms and the achieved time-frequency resolution [57, 240]. The smaller values of σ allow better suppression of the cross-terms, whereas larger values provide better resolution of the signal auto-terms [57, 240]. However, independent control over the smoothings in the time and frequency domains is not possible using the CWD [121]. Moreover, the time-frequency representation obtained with the CWD exhibits higher levels of cross-terms when applied to the signals whose components have overlapping time or frequency supports [121]. The reduction of the low-frequency cross-terms is prevented by the relatively slow decay of the exponential kernel [206]. Furthermore, the auto-terms can be distorted due to the not very flat passband of the exponential kernel [206]. The CWD is obtained as [26, 57, 119]

$$CWD_s(t,f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{\frac{\sigma}{\pi}} \frac{1}{2\,|\tau|}\ e^{-\frac{\sigma u^2}{16\tau^2}}\ s\left(t+u+\frac{\tau}{2}\right) s^*\left(t+u-\frac{\tau}{2}\right) du\ e^{-j2\pi f\tau} d\tau. \quad (2.30)$$

Some of the properties satisfied by the CWD are: realness, time-shift invariance, frequency-shift invariance, time marginal, frequency marginal, global energy, instantaneous frequency, group delay, reduced interferences, and invertibility [36, 119, 121, 135].

Next, the Butterworth distribution (BUD) uses the kernel in the form of a two-dimensional low-pass filter in the ambiguity function domain whose parameters of the variably flat passband and the narrow transition region can be adjusted [24, 206]. Thus, the BUD upgrades the CWD, simultaneously better preserving the resolution of the signal auto-terms and better suppressing the low-frequency cross-terms in the representation [206]. The BUD is defined as [26, 206]

$$BUD_s(t,f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \sqrt{\sigma} \frac{1}{2\,|\tau|}\ e^{-\frac{\sqrt{\sigma}|u|}{|\tau|}}\ s\left(t+u+\frac{\tau}{2}\right) s^*\left(t+u-\frac{\tau}{2}\right) du\ e^{-j2\pi f\tau} d\tau. \quad (2.31)$$

The BUD preserves useful properties of the CWD, i.e.; it has the following properties: realness, time-shift invariance, frequency-shift invariance, time marginal, frequency marginal, instantaneous frequency, group delay, and reduced interferences [206].

Another reduced-interference distribution with the property of preserving the time and frequency supports is the Born-Jordan distribution (BJD) [63]. The BJD is based on the

narrowband sinc kernel in the ambiguity function domain and results in a time-frequency representation in which the cross-terms are well attenuated [65, 135]. However, at the same time, the time-frequency resolution of the signal auto-terms is reduced [65, 135]. The BJD is calculated as [88, 135]

$$BJD_s(t,f) = \int_{-\infty}^{\infty} \frac{1}{|\tau|} \int_{t-\frac{|\tau|}{2}}^{t+\frac{|\tau|}{2}} s\left(u+\frac{\tau}{2}\right) s^*\left(u-\frac{\tau}{2}\right) du\ e^{-j2\pi f\tau} d\tau. \tag{2.32}$$

The BJD is known for the following properties: realness, time-shift invariance, frequency-shift invariance, time marginal, frequency marginal, global energy, instantaneous frequency, group delay, time support, frequency support, and reduced interferences [36, 119, 135].

By smoothing the BJD in the frequency direction, the Zhao-Atlas-Marks distribution (ZAMD) is obtained [64, 286]. The ZAMD is designed with a cone-shaped kernel that contributes to the cross-term attenuation while simultaneously providing a good time-frequency resolution, especially for the multiple sinusoidal burst signals characterized by the quasi-stationary instantaneous frequencies [120, 201, 286]. The cone-shaped kernel performs the directional band-pass filtering of the WVD [120]. The ZAMD is defined as [201, 286]

$$ZAMD_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \int_{t-\frac{|\tau|}{2}}^{t+\frac{|\tau|}{2}} s\left(u+\frac{\tau}{2}\right) s^*\left(u-\frac{\tau}{2}\right) du\ e^{-j2\pi f\tau} d\tau. \tag{2.33}$$

Some of the properties of the ZAMD are: realness, time-shift invariance, frequency-shift invariance, time marginal, time support, and reduced interferences [36, 119].

The reduced-interference distribution with a kernel based on the first kind Bessel function of order one (RIDB) is another distribution having a property of efficiently attenuating the cross-terms in the time-frequency representation while simultaneously maintaining a high time-frequency resolution [110, 230]. The RIDB is calculated as [26, 110]

$$RIDB_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \int_{t-|\tau|}^{t+|\tau|} \frac{2g(u)}{|\tau|\,\pi} \sqrt{1-\left(\frac{u-t}{\tau}\right)^2}\ s\left(u+\frac{\tau}{2}\right) s^*\left(u-\frac{\tau}{2}\right) du\ e^{-j2\pi f\tau} d\tau. \tag{2.34}$$

Another reduced-interference distribution, designed with a kernel based on the binomial coefficients (RIDBN), can be defined as [26, 36, 275]

$$RIDBN_s(t,f) = \sum_{\tau=-\infty}^{\infty} \sum_{u=-|\tau|}^{|\tau|} \frac{1}{2^{1+2|\tau|}} \binom{1+2|\tau|}{1+u+|\tau|} s\left[t+u+\tau\right] s^*\left[t+u-\tau\right]\ e^{-j4\pi f\tau}. \tag{2.35}$$

Next, the reduced-interference distribution that uses a Hanning window-based kernel (RIDH) is given by [26, 135]

$$RIDH_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \int_{-\frac{|\tau|}{2}}^{\frac{|\tau|}{2}} \frac{g(u)}{|\tau|} \left(\cos\left(\frac{2\pi u}{\tau}\right)+1\right) s\left(t+u+\frac{\tau}{2}\right) s^*\left(t+u-\frac{\tau}{2}\right) du\, e^{-j2\pi f\tau} d\tau. \tag{2.36}$$

Finally, the reduced-interference distribution with a kernel based on the triangular window (RIDT) is obtained as [26, 135]

$$RIDT_s(t,f) = \int_{-\infty}^{\infty} h(\tau) \int_{-\frac{|\tau|}{2}}^{\frac{|\tau|}{2}} \frac{2g(u)}{|\tau|} \left(1-\frac{2|u|}{|\tau|}\right) s\left(t+u+\frac{\tau}{2}\right) s^*\left(t+u-\frac{\tau}{2}\right) du\, e^{-j2\pi f\tau} d\tau. \tag{2.37}$$

Note that most of the above-presented TFDs of Cohen's class do not satisfy non-negativity property and thus cannot be considered energy distributions. Namely, the joint time-frequency representations cannot be considered probability density functions in a strict mathematical sense due to the time and frequency being non-commuting variables [180]. Nevertheless, the TFDs still represent powerful analysis tools providing useful insights into the time-frequency characteristics of the non-stationary signals.

The above-described properties of the quadratic TFDs from Cohen's class are demonstrated on the considered synthetic signal example. Figure 2.4 shows the 12 different quadratic TFDs (BJD, BUD, CWD, PWVD, RIDB, RIDBN, RIDH, RIDT, SP, SPWVD, WVD, and ZAMD) calculated for the example of the two-component non-stationary signal $s(t)$, while Figure 2.5 gives the TFDs of the noise-corrupted signal version. As seen in Figure 2.4, the considered TFDs provide different levels of time-frequency resolution and cross-term attenuation. Moreover, the presence of noise introduces additional cross-terms between the signal components and the noise, as can be seen in Figure 2.5.

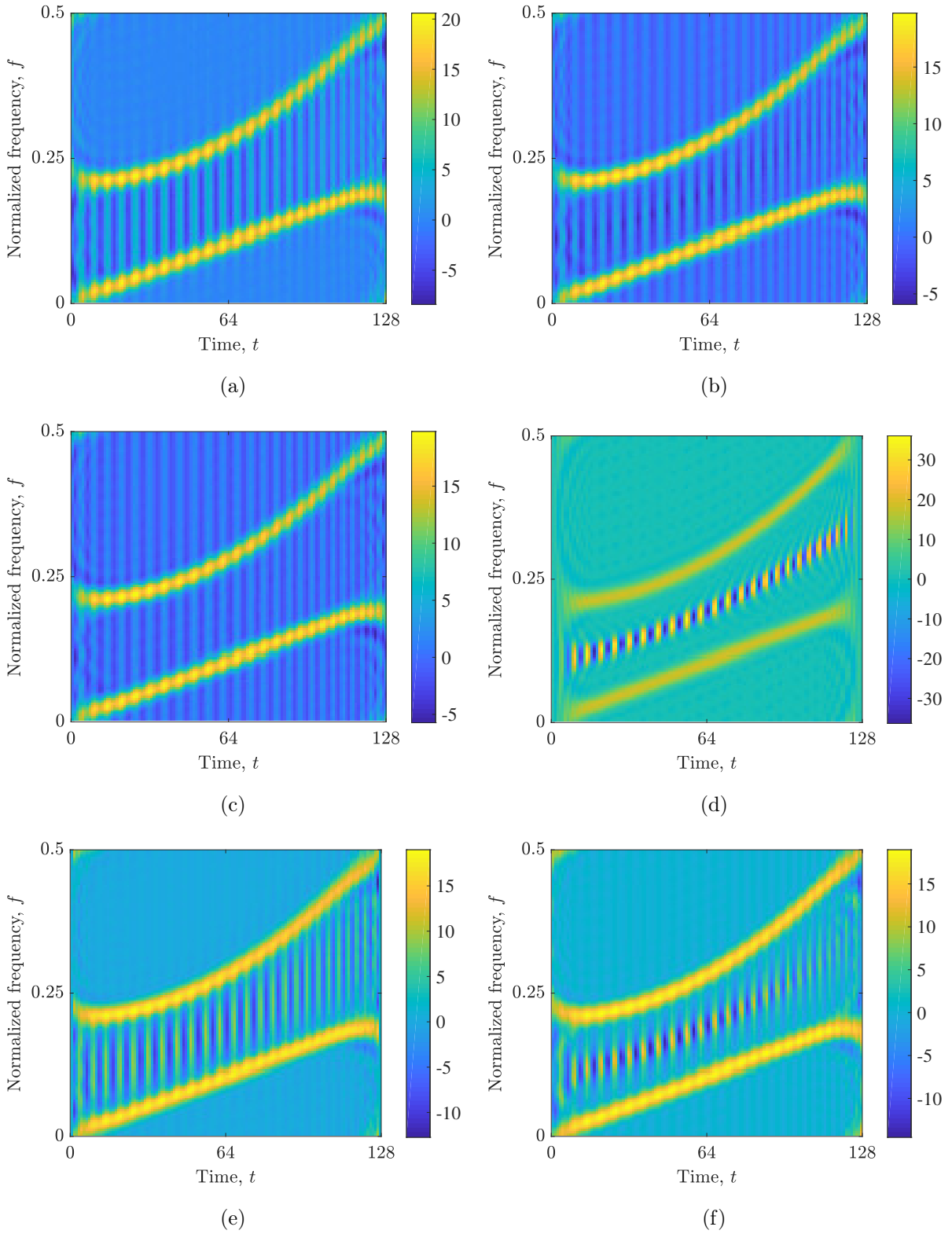


Figure 2.4 TFDs of the considered example of a two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

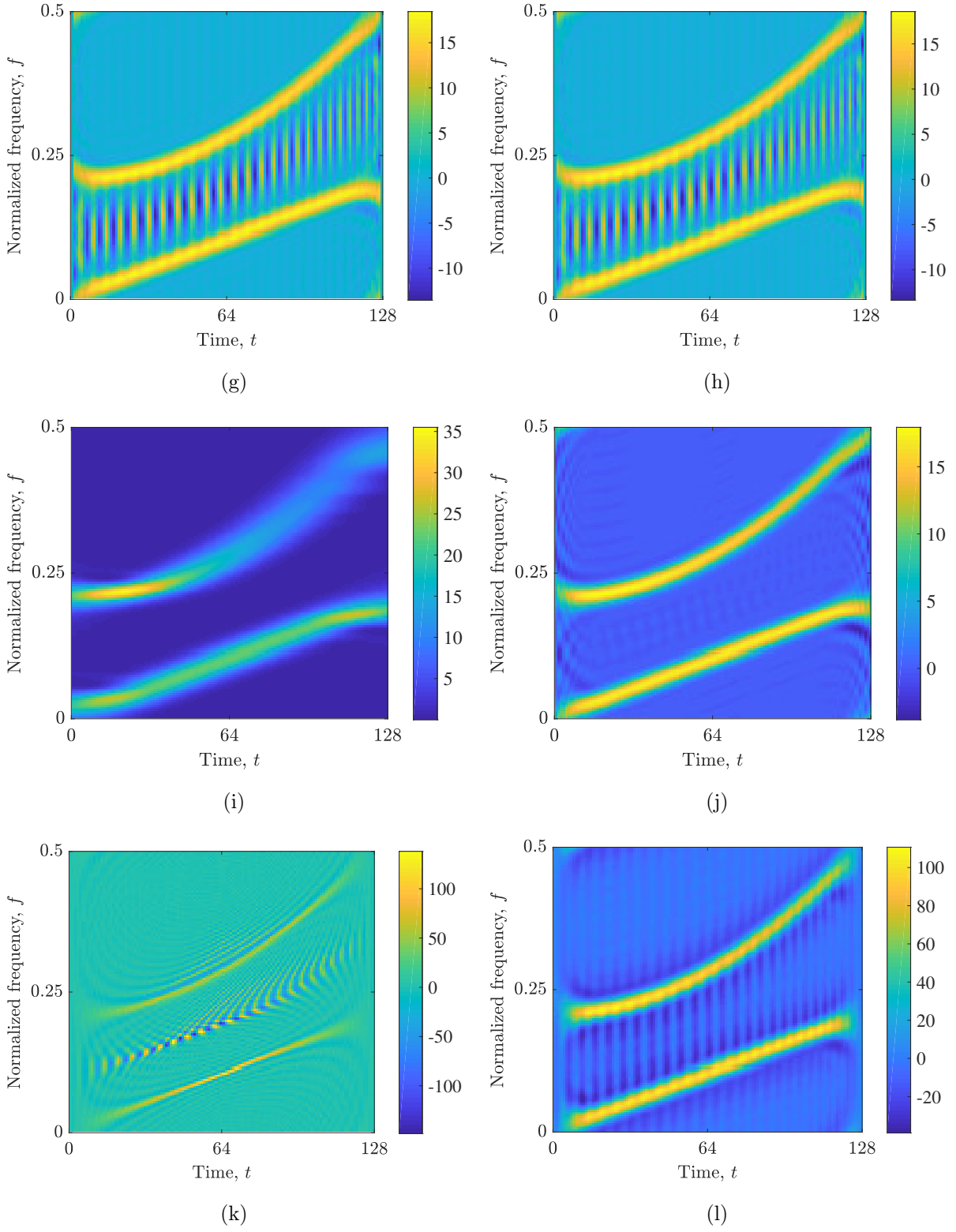


Figure 2.4 (cont.) TFDs of the considered example of a two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

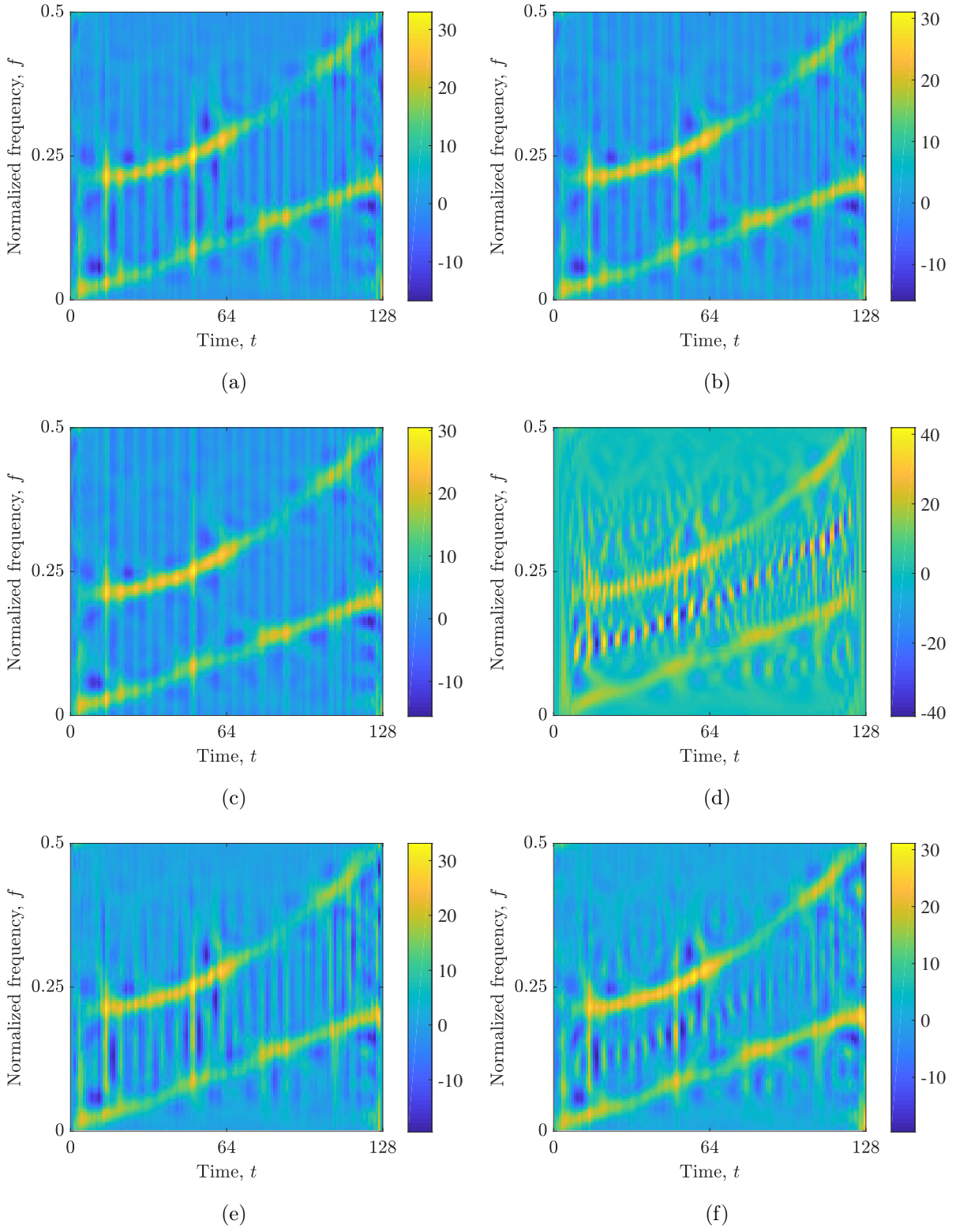


Figure 2.5 TFDs of the considered example of a noisy two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

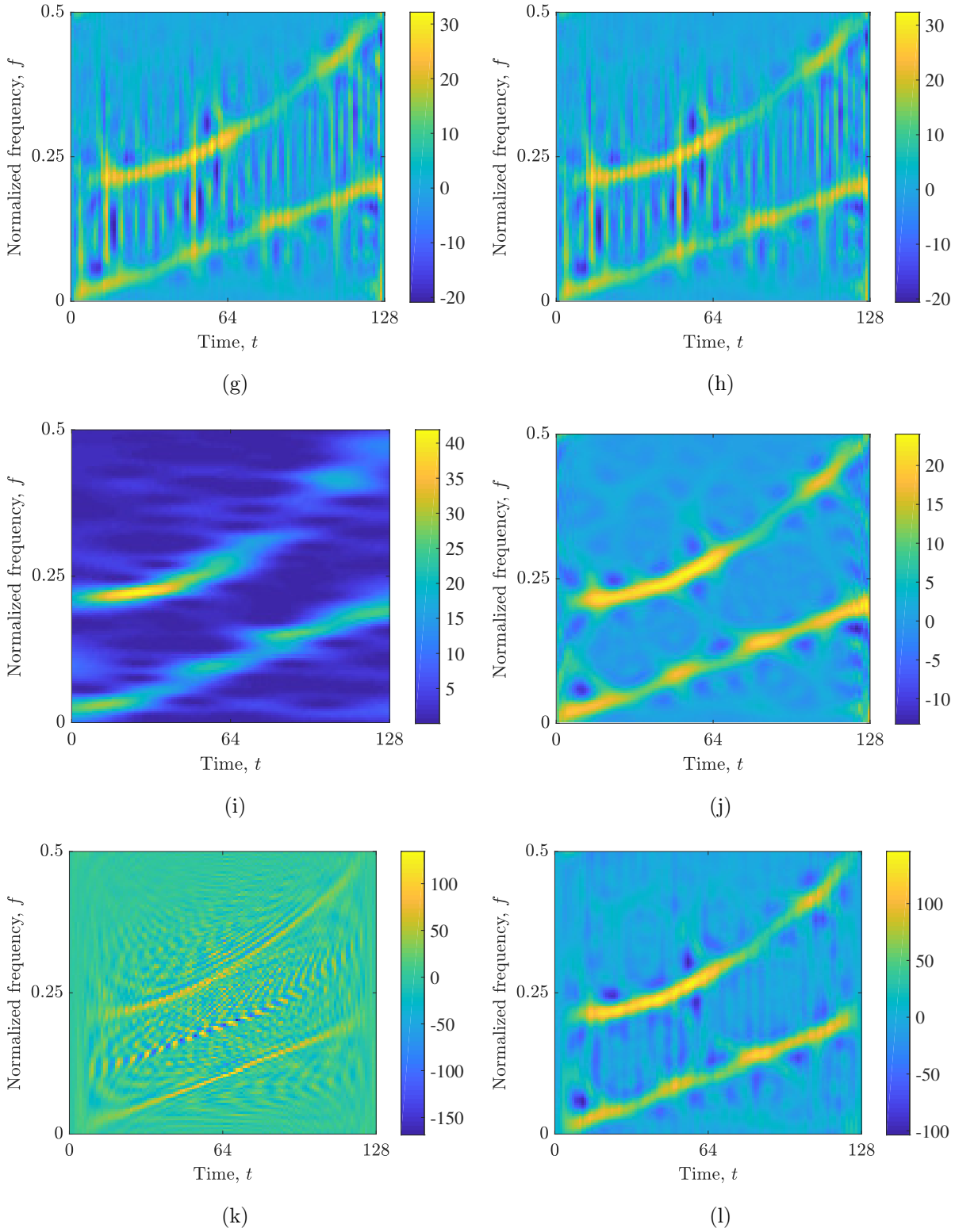


Figure 2.5 (cont.) TFDs of the considered example of a noisy two-component non-stationary signal $s(t)$ with a parabolic and a linear frequency modulated component: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

CHAPTER 3

DEEP LEARNING

This chapter presents a brief insight into the field of machine learning, considering recent trends, general principles, theory, and application. The particular focus is given to the subset of machine learning, called deep learning. This chapter also introduces the main concepts and principles of ANNs. CNNs, representing the type of deep neural networks that form the core of the non-stationary GW signal detection method proposed in this thesis, are discussed in more detail. Finally, a brief overview of recent deep learning approaches for signal classification is provided, emphasizing the utilization of two-dimensional time-frequency signal representations based on Cohen's class of TFDs.

3.1 Machine Learning Algorithms

The term artificial intelligence (AI) refers to the computer algorithms and systems that mimic cognitive processes characteristic for humans, including the ability to learn from experience, reason, and generalize, in order to make decisions and deal with problems that humans struggle to describe formally but can easily solve intuitively [103]. The alternative definition of AI states that this research field deals with the construction of intelligent agents that can make decisions in order to achieve the best expected outcome, i.e., to do the right thing according to the provided objective [224]. Today, AI represents a rapidly developing field with numerous real-world applications.

Machine learning is a subfield of the AI that encompasses algorithms that can automatically learn from provided data examples to solve new tasks [136]. Learning is defined as the improvement of the algorithm's performance at some tasks with experience, which is measured by some performance measure [103, 188]. Some of the machine learning tasks include classification, classification with missing inputs, regression, transcription, machine translation, structured output, anomaly detection, synthesis and sampling, imputation of missing values, denoising, density estimation, and many more [103].

The generalization to the new problems is achieved without the need for reprogramming

of these algorithms, in contrast to the classical task-specific algorithms based on strictly defined rules [103]. Thus, machine learning algorithms can be applied to problems in different fields by providing different data during the training procedure [103]. This property has led to the expansion of machine learning in recent years to various research and technology fields and commercial applications, such as speech recognition, computer vision, data analysis, robot control, natural language processing, and other [136].

Machine learning can be divided into two main categories: supervised and unsupervised learning [103, 136, 229]. As the most commonly used form of machine learning, supervised learning is based on providing algorithms with correctly labeled data [103, 136, 150]. The algorithm receives an input during the training and produces an output based on the learned mapping between input and output [136, 150]. Some mapping types include neural networks, support vector machines (SVMs), logistic regression, decision forests, decision trees, Bayesian classifiers, and kernel machines [136]. These mappings are estimated using different learning algorithms based on optimization theory [136].

The outputs predicted by the machine learning algorithm can be of different types. For example, the output can take on one of two labels in binary classification, while in multiclass classification, the output can assume one of the multiple considered labels [136]. Moreover, other problems tackled by machine learning include multilabel classification where the output of the algorithm can be assigned with multiple labels simultaneously, ranking problems where the output provides the order of some elements in the set, and general structured prediction problems where the output represents some combinatorial object [136]. The error between the obtained output and the desired (known) output is calculated using an objective function [150]. The algorithm then modifies its parameters to reduce the error [150]. After the training, the machine learning algorithm's performance, i.e., its generalization ability, is then checked on the test dataset containing different data examples than those provided during the training [103, 150].

On the other hand, in unsupervised learning, machine learning algorithms are provided with unlabeled data for learning, assuming some structural properties of these data [103, 136, 229]. These assumptions are represented by a criterion function which is then optimized [136]. Some unsupervised machine learning algorithms are designed to learn the probability distribution of the provided dataset, while others perform the clustering of the dataset into groups of similar data examples [103].

However, despite their success in a wide variety of applications, traditional machine learning algorithms are limited in their capacity to process raw data directly [103, 150]. Namely, the application of these types of learning systems requires transforming the data to the appropriate representation using a domain-specific feature extractor, i.e., the data must frequently be manually reduced into a representation appropriate for a particular task [103, 150]. In addition to requiring domain expertise, determining the appropriate hand-crafted representation is challenging and time-consuming, even for the

subject specialists [103]. Therefore, the applicability of these conventional algorithms is severely limited, which has motivated the development of representation learning and deep learning.

Representation learning is a subfield of machine learning that seeks to overcome this issue of traditional machine learning algorithms by developing algorithms that can take raw data as input and learn on their own to create appropriate representations of the data and automatically extract important features that will be used for detection or classification [32, 103, 150]. Compared to hand-crafted representations, learned representations allow rapid adaptation to new problems, often with improved performance [103].

Deep learning is a subfield of representation learning with numerous emerging applications, achieving state-of-the-art performance in image recognition, speech recognition, and many other fields [103, 150]. The recent trend in the expansion of deep learning to various fields is due to several developments. The most important one is the increased availability of big data, i.e., large datasets that can be used to train deep learning algorithms [103]. These datasets of numerous images, videos, audios, and other formats have become widely available due to the increasing digitalization of modern society [103]. Another important reason is that today's computational resources allow training larger deep learning models than before [103]. The increased computational capabilities are based on hardware development, including faster and more powerful central processing units (CPUs) and graphics processing units (GPUs), the application of distributed computing, and the improved software infrastructure and network connectivity [103].

Deep learning algorithms are characterized by their ability to extract multiple levels of features for each task automatically, thus identifying relevant representations of the raw input data and building the hierarchy of concepts [103, 150]. The algorithms are based on deep computational architectures, composed of multiple interconnected processing layers of simple but non-linear modules, combined with advanced optimization algorithms [103, 150]. Each layer transforms the data representation, where initial network layers learn simpler features while deeper layers utilize the outputs of the previous layers to learn more abstract representations [103, 150, 229]. By composing hierarchical internal representations, these deep learning algorithms can capture complex non-linear functions [103, 150, 229]. The key advantage of deep learning algorithms is that these representations are learned automatically from the data during the training process, thus eliminating the need for manual design done by engineers [150].

Deep neural networks are computational models that form the basis of deep learning applications. Their name comes from the fact that they are loosely based on neurological processes and that they are composed of various functions forming chain structures, which can be represented as a network with multiple layers whose number determines the depth of the model [103]. Deep networks may express functions of increasing complexity by adding more layers and more units inside each layer [103]. Deep feedforward networks,

also known as feedforward neural networks or multilayer perceptrons, are the type of deep learning models with no feedback connections that would feed outputs of the model back into its structure, i.e., information propagates through these models from the input, through the intermediate computation functions, to the output [103]. Feedforward neural networks are used to approximate a function mapping an input to output [103]. On the other hand, deep neural networks that include feedback connections are called recurrent neural networks (RNNs) [103].

ANNs represent building blocks of deep neural networks, and their basic concepts are presented next.

3.2 Artificial Neural Networks

ANNs are computational models composed of fundamental units called artificial neurons [197, 229]. These models are able to learn from provided data examples [197]. Artificial neurons are based on perceptrons whose biologically inspired concept [106] was first introduced in [219] in 1957. The perceptron represents a linear binary classifier used in supervised learning that operates on the input vector $\mathbf{x}$ to produce the weighted output $f(\mathbf{x})$ [106, 197]. This process can be modeled as [197]

$$f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b, \tag{3.1}$$

where $\mathbf{w}$ is the vector of real-valued weights determining the importance of the respective inputs to the output and b is the bias representing the offset value.

Weight and bias parameters are tuned during the training procedure while the algorithm learns from the provided data. Although the potential application of the algorithms based on the single perceptron is severely limited [187], combining multiple interconnected artificial neurons into multiple layers, and thus forming ANNs, opens a much larger range of applications [229]. Moreover, it was shown in [123] that a multilayer feedforward network with a linear output layer and only one hidden layer represents a universal approximator if there is a sufficient number of hidden units, i.e., the network can approximate any Borel measurable function to any desired degree of accuracy.

ANNs generally consist of an input layer, at least one hidden layer, and an output layer. Figure 3.1 depicts the architecture of an example of a simple feedforward neural network with three neurons in the input layer, five neurons in the hidden layer, and two neurons in the output layer. ANNs utilize non-linear activation functions applied to the outputs within the hidden layers and the output layer [150]. Commonly applied activation functions include the rectified linear unit (ReLU) activation function mathematically defined as $\max(0, x)$, the logistic sigmoid activation function defined as $1/(1 + e^{-x})$, and the hyperbolic tangent activation function defined as $\tanh(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$

[150]. The ReLU is currently the most popular activation function option as it has provided notable results in many applications of deep neural architectures [101, 132, 194].

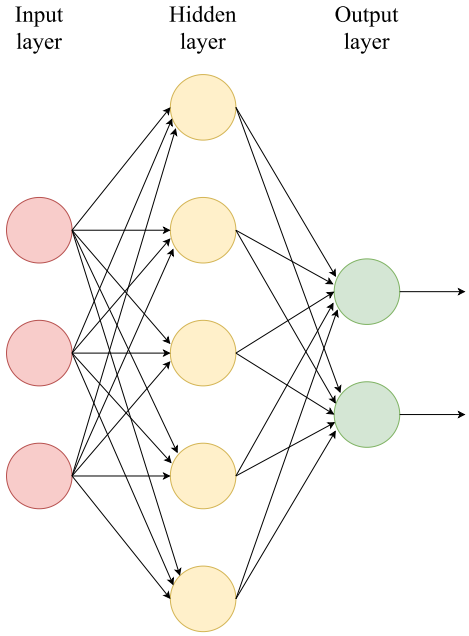


Figure 3.1 Example of a simple feedforward neural network architecture with one hidden layer.

In addition to defining the appropriate input data and the model architecture, another important factor in applying ANNs is the learning algorithm. The most dominantly used learning algorithm is the backpropagation algorithm, proposed in [222], combined with some gradient descent methods [151, 229]. The training procedure usually starts with randomly initializing the weights of the ANN to the small values and then performing the backpropagation over multiple epochs to minimize the cost function using the calculated gradient of the cost function with respect to the weight and bias parameters [103]. The backpropagation procedure is based on the chain rules for derivatives and consists in propagating the errors from the output layer, through the hidden layers, back to the input layer of the ANN each time the performance of the model is evaluated [103, 150, 151]. In this procedure, each artificial neuron's weights and biases are adjusted to reduce the cost function using the gradient-based optimization methods [103, 151].

The stochastic gradient descent (SGD) method has been conventionally used for this purpose [150, 220]. The SGD method is based on the noisy estimate of the average gradient of the cost function over all data examples, where this estimate is obtained using only a smaller subset of the total data available for the training [103, 150, 220]. These randomly chosen training data examples are called mini-batches [103, 220]. After all training data examples are used in this procedure, an epoch of the training is completed [197]. The learning rate parameter controls the magnitude of changes applied to the weight values in each iteration of the gradient descent [197]. Recently developed methods based on the

adaptive learning rates have shown improved performance in various tasks [220].

The appropriate initialization of the weights [100] and the selection of the optimal hyperparameter values is significant to enable the successful training of the neural network [31, 103, 197]. The selection of hyperparameters is often made using heuristic approaches based on human expertise [197]. The hyperparameter selection can also be made by conducting an extensive grid search in the hyperparameter space [31], but some more sophisticated, automatic selection techniques are also used [33, 237]. Moreover, there are also many techniques to improve and speed up the training process [31], such as appropriate choice of the cost function [197], and regularization methods that increase the ability of the model to generalize beyond the provided training data [103]. Some of regularization methods include L1 and L2 regularization [103, 197], dataset augmentation [232], dropout [118, 238], bagging (ensemble methods) [42], and early stopping the training [278]. Other techniques that prevent the overfitting of the neural network to the training data include adversarial training [104, 242], batch normalization [131], pretraining [83], transfer learning [205], one-shot learning [87], and zero-shot [204] or zero-data learning [147].

CNNs represent special types of deep feedforward networks. Some of the fundamental concepts of CNNs are described next.

3.3 Convolutional Neural Networks

CNNs [148, 149], sometimes called only convolutional networks or ConvNets, are loosely inspired by the biological structure of the mammalian visual cortex [103, 128]. This type of neural network is specially designed to deal with data in the form of multidimensional arrays by utilizing their spatial structure [77, 108, 150, 197]. A typical example of such data are color images composed of three two-dimensional arrays with pixel intensities, where each array represents one color channel (red, green, blue (RGB)) [77, 150].

The name of CNNs comes from the fact that they use a mathematical operation called convolution in their layers [103]. The convolution is a linear transformation performed between the multidimensional array of input data and the multidimensional array called kernel or filter, producing the output called the feature map [77, 103]. These multidimensional arrays are referred to as tensors [103]. The kernel slides across the input, and the product between each kernel element and the input element it overlaps is calculated [77]. The obtained results are added together to produce the output at the considered location [77]. In CNN applications, the operation performed usually does not correspond to the definition of the convolution operation in the strict signal processing sense but refers to the cross-correlation operation in which no kernel flipping occurs [103]. This operation is for the two-dimensional input data (image) X and the two-dimensional kernel K defined

in the discrete form as [103]

$$Y(m,n) = (X * K)(m,n) = \sum_i \sum_j X(m+i, n+j) K(i,j). \tag{3.2}$$

Three main ideas utilized by CNNs are sparse connectivity, shared parameters, and equivariant representations [103, 149, 150, 197]. In contrast to traditional ANN layers where each input unit is connected to each output unit, CNNs have sparse connectivity, which is achieved by using the kernel whose size is smaller than the input size [103]. In this way, each output unit is affected by the region of a smaller number of input units, known as its receptive field [103, 108, 197]. The reduced number of parameters reduces memory requirements and computational costs and increases statistical efficiency [103]. In CNNs, kernels in shallow layers allow detection of smaller and simpler features, such as edges and corners, while units in deeper layers build more complicated interactions by indirectly processing larger portions of the input data, i.e., these units have larger receptive fields than those in shallow layers [103, 149].

In traditional ANNs, each weight is used only once while computing the layer's output, while in CNNs, the same kernel is used at all input locations to obtain a feature map [103, 108, 149]. Different feature maps in a layer use different kernels to extract different features [77, 108, 149, 150]. The kernel weight and bias parameters for each feature map are learned automatically according to the learning algorithm based on the backpropagation of gradients, similarly to traditional deep ANNs [103, 150]. The parameter sharing additionally reduces memory requirements and increases statistical efficiency [103, 149]. In addition, shared weights allow CNN layers to deal with the spatially translated input data automatically [103, 150]. Namely, if convolution is applied to the translated input, the convolution output will be translated in the same manner [103]. This property is called equivariance to translation [103]. For equivariance to the scaling and rotation of the input data, CNNs require other techniques [103, 150].

A typical CNN architecture contains two types of layers: convolutional and pooling layers [103, 150]. The convolutional layer performs multiple convolution operations in parallel, which results in a set of linear activations, i.e., feature maps [103]. An element-wise non-linear activation function, most often the ReLU, is then applied to each obtained linear activation to which bias is usually also added [77, 103, 108]. The pooling layer applies a pooling function to the output of the nonlinearity [103]. The described convolutional and pooling layers are in the literature also sometimes considered as a single layer with three stages: convolution, detector (non-linear activation), and pooling stage [103].

The convolutional layer performs multiple convolutions in parallel to extract multiple features at multiple input locations [103]. Moreover, CNNs usually utilize multi-channel convolution as the input is usually a grid of vector values, e.g., color images are considered three-dimensional tensors [103]. In deep CNNs, thus obtained output of one layer represents

the input to the next layer [103]. The stride is a parameter of convolution operation that defines the distance between the two consecutive positions of the kernel over the input [77]. The stride can be used for downsampling of the convolution output, resulting in reduced computational costs and somewhat reduced precision of feature extraction [77, 103].

Zero padding is another important convolution parameter, which implies padding zeroes to the input to increase its dimensions [77, 103]. When applying the convolution without zero-padding, the width of the output shrinks at each convolutional layer by the number of pixels equal to the kernel width reduced by one [103]. Therefore, zero padding can prevent shrinking the representation with depth and thus allow the design of deep CNNs with the desired number of layers [103].

There are three special cases of zero-padding applications [103]. The first one is where no zero padding is applied, and the kernel slides only over those positions for which the whole kernel is within the input image [103]. This case is known as valid convolution and results in the output pixels where each pixel is a function of the same number of input pixels [103]. The obtained output has the width of $m - k + 1$, where m is the width of the input image and k is the width of the kernel [77, 103]. Therefore, this case of zero padding causes the output shrinkage at each CNN layer and thus limits the maximum number of convolutional layers [103]. The second case of zero-padding application, also known as same convolution, consists in adding zeroes to the input to keep the output the same size as the input [77, 103]. This case removes the constraints on the number of convolutional layers in the network, but the border input pixels have a less effect on the output pixels than those in the image center [103]. Finally, the third case, known as full convolution, results in the output of width $m + k - 1$, which is achieved by adding enough zeroes to the input so that each input pixel is taken into account by the kernel k times in each direction [77, 103]. However, the border output pixels are a function of a smaller number of pixels than those close to the center [103].

Several recently developed concepts, such as dilated convolutions [279], improve the performance of CNNs even further. The dilated convolutions allow aggregation of multi-scale contextual information over larger regions without losing resolution, which is achieved by increasing the size of kernels by inserting the spaces between their elements [279]. The dilation rate is defined by the parameter d_r, where $d_r - 1$ spaces are inserted between the kernel elements [77]. In this way, the receptive field of output units is increased without increasing the kernel size [77, 108].

Pooling consists of sliding a rectangular window across the input and applying a pooling function that substitutes the output of nonlinearity at a particular location by summarizing the subregion of the neighboring outputs [77, 103]. Thus, pooling reduces the resolution of the feature maps [108]. The most commonly used pooling technique is the max pooling that selects the maximum output value within a rectangular region [40, 77, 103]. Alternative pooling functions include the average, the weighted average, and

the L^2 norm of the neighboring values within the pooling region [103]. Pooling makes the representation invariant to small local translations of the input, as the values obtained by pooling do not change much when the input is translated by a small amount [77, 103]. This invariance property is beneficial in applications that prefer the information on the existence of some feature over its exact location [103, 149, 197].

Moreover, pooling over multiple features obtained as the outputs of the convolutions with separate learned parameters can allow CNNs to learn to be invariant to various transformations of the input data [103]. Furthermore, pooling reduces the feature map size as fewer pooling units are needed than the detector units when a stride between pools greater than one is used [103, 197]. Thus, pooling makes CNNs more robust to the small shifts, distortions, and noise in the input data, while simultaneously lowering computational costs [103, 108, 149, 150].

In a typical CNN architecture, several convolutional and pooling layers are stacked and followed by the fully connected layers, also called the dense layers [103, 108, 150]. Fully connected layers are traditional neural network layers that are connected deeply, i.e., each unit in the layer is connected to all units in the previous layer. The tensor output of the convolutional layers needs to be flattened, i.e., reshaped to the one-dimensional vector, to be used as the input to the fully connected layers [103]. Fully connected layers perform high-level reasoning [108]. A typical CNN architecture is depicted in Figure 3.2. The example architecture shown in Figure 3.2 consists of two convolutional layers (C1 and C2 feature maps), two pooling layers (P1 and P2 feature maps), a fully connected layer, and an output layer.

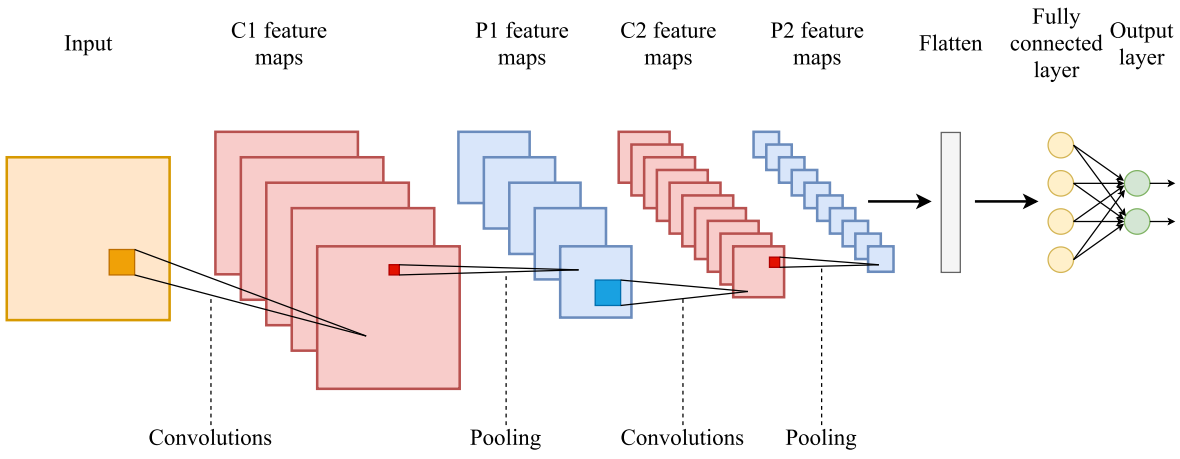


Figure 3.2 Typical CNN architecture.

The modern framework of CNNs was established in [152] and later improved in [148]. In these studies, a multilayer ANN called LeNet-5 was used to classify handwritten digits. However, CNNs have been mostly neglected in the field of computer vision [153] until the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [223] in 2012. In this

competition, a deep CNN, known as AlexNet, was used to classify 1.2 million images from the ImageNet dataset into 1000 classes, considerably outperforming the state-of-the-art methods having won the previous competitions [145]. These results were achieved due to several improvements and novel techniques, including the efficient implementation of convolution operations on GPUs, ReLU activation functions, dropout regularization technique, and dataset augmentation [145, 150]. Consequently, CNNs have gained increased popularity in various image and video processing applications, obtaining state-of-the-art performances and becoming the dominant technique for object detection and recognition [108, 145, 150]. This type of deep neural network is most efficient for the implementation on modern computer hardware, allowing rapid training and evaluation [150].

The following section presents a brief overview of the recent applications of deep learning for signal classification utilizing two-dimensional time-frequency signal representations.

3.4 Deep Learning-Based Approaches for Signal Classification

The conventional pattern recognition and traditional machine learning algorithms for time-series signal classification, based on the carefully selected and manually extracted signal features, can achieve high classification accuracy [51, 217, 251, 258]. Nevertheless, extracting hand-crafted signal features appropriate for a particular application represents a tedious task requiring domain-specific knowledge. With its recent intensive development, deep learning has also been applied to time-series signal classification in various domains, allowing automatic feature extraction [86, 137, 203, 268].

On the other hand, two-dimensional signal representations open up possibilities for applications with two-dimensional CNNs with superior performance in image recognition tasks. Most recent studies have addressed the use of CNNs with spectrogram representations in different applications, including classification of speech [27], electrocardiogram (ECG) [277], electroencephalogram (EEG) [281], and radar [270] signals.

However, alternative TFDs from Cohen's class have mainly been applied to time-series signals to extract characteristic features from time-frequency representations, which are then used as input to traditional machine learning [39, 93, 111, 189, 193] or ANN-based deep learning classification algorithms.

The SPWVD in [259] and several other Cohen's class TFDs in [260] were used to extract features from the corresponding time-frequency representations of EEG signals. The dimensionality of the extracted features representing the fractional signal energy in the specific time-frequency area was reduced by principal component analysis. The obtained features were then used as input to the simple, shallow, feedforward ANN to detect EEG signals containing epileptic seizure activity. The TFDs from Cohen's class were utilized in

[256] to extract features from ECG signals, with these features then fed to the shallow ANN, enabling arrhythmia detection. In [241], the vibration signal features extracted by the WVD were used to detect gearbox faults based on the shallow ANN. The complex image processing consisting of several stages was applied in [285] to the CWD images to extract features for Elman neural network-based classification of low-probability-of-intercept (LPI) radar signals. In [210], the ZAMD of vibration acceleration signals was decomposed by non-negative matrix factorization and used to detect faults by the neural network ensemble. The study in [171] used spectral kurtosis values calculated from the BUD as input to the radial basis function network to classify transient power quality disturbances. Finally, the WVD was applied in [20] to extract features from the electromyogram (EMG) signals to allow their ANN-based classification.

Recently, several studies on directly applying TFDs with CNNs for classification purposes have also been conducted. The binary images obtained by applying the image processing techniques to the CWD of the cognitive radio signals were used as the CNN input in [284]. The study in [109] reported using the CWD for the LPI radar signal recognition system with the pretrained deep CNN for feature extraction and the SVM for classification, while the study in [75] utilized the CWD and the CNN for detecting a single line-to-ground fault in distribution systems. The CWD was also used in the simple CNN-based algorithm for radar emitter signal detection described in [170]. Additional applications of the CWD included the study in [233] that discussed LPI radar signal recognition based on the binary CWD images and dense CNN; the study in [92] where the method for partial discharge pattern recognition based on the CWD, variational mode decomposition, and optimized CNN with the cross-layer feature fusion was proposed; and the study in [129] with the CNN-based classification of the CWD images of the LPI radar signals.

Moreover, the WVD was applied to classify the power quality disturbances in [48], the radar signals in [266], and wiring faults in electric vehicles in [55], following a similar CNN-based approach. The utilization of the WVD and deep geometric convolutional network for signal modulation classification was studied in [155]. The WVD and CWD were also combined with the deep CNNs in [269] to classify the radar signals obtained from soils with different moisture values, whereas the WVD and BJD were used in [280] in the feature fusion algorithm for automatic target recognition based on the radar-generated high-resolution range profiles and the CNN for feature extraction.

Furthermore, the approach combining the PWVD and CNN was applied to detect arrhythmia from the ECG signals in [276] and anterior cruciate ligament injury from the EMG signals in [113]. Moreover, the SPWVD was used in [168] to obtain input images to the triplet CNN for radar signal classification. The SPWVD of the EEG signals was utilized as input to the deep CNNs for emotion recognition in [140], schizophrenia detection in [142], and Parkinson's disease detection in [141]. In [247], the SPWVD was applied with

the SP and RIDH representations for radar signal classification based on the mixed CNN consisting of three two-dimensional CNNs and one three-dimensional CNN. In addition, several TFDs from Cohen's class combined with the simple CNN were studied in [29] in the application of identifying emitters of Internet of Things wireless devices.

A new kernel function for the TFD with the form characteristic for Cohen's class was introduced in [212] and thus obtained TFD images were heavily preprocessed before being utilized for the CNN-based classification of radar signals. The proposed TFD was also applied in [213] for radar signal recognition based on the convolutional denoising autoencoder used to denoise and repair the TFD images, and the deep CNN used to classify these processed images. In [211], radar signals were classified using preprocessed TFDs with multiple kernel functions of Cohen's class type as input to the simple CNN for feature extraction and the deep Q-learning network for classification.

However, the above-discussed recent studies mainly focus on particular TFDs in a specific application. Furthermore, these approaches are often based on classical CNNs with custom configurations for the studied applications. Thus, the existing research lacks a comprehensive and detailed analysis of different TFDs from Cohen's class combined with the advanced, state-of-the-art two-dimensional CNN architectures (especially in the field of physics like GW detection). Moreover, to the best of the author's knowledge, the existing studies do not address the problem of classifying non-stationary signals with very low SNR values, which is one of the main contributions of this thesis.

Before proceeding to the developed approaches and obtained results, the next chapter introduces the basic insight to GWs and presents the problem of their detection in intensive noise.

CHAPTER 4

GRAVITATIONAL-WAVE SIGNALS AND OVERVIEW OF EXISTING APPROACHES FOR THEIR DETECTION

This chapter defines GW signals, presents the fundamental concepts in GW research, and addresses three main groups of approaches for GW data analysis. Existing techniques for GW detection may be divided into the following categories: matched filtering-based techniques, denoising techniques, and machine learning-based techniques.

4.1 Gravitational Waves

GWs are ripples in the curvature of space-time caused by massive accelerating objects that represent some of the most energetic events in the Universe [159, 179, 249]. These ripples propagate in the form of waves in all directions outward from the source at the speed of light [159, 179, 249]. There are four main types of GWs with respect to their sources: compact binary coalescence (CBC), bursts, continuous, and stochastic GWs [158, 249]. Each type of GWs generates a characteristic signal pattern.

CBC GWs are generated by pairs of orbiting massive and compact objects [158, 179]. This type of GWs is the only one that has been successfully observed on Earth so far. There are three categories of binary systems that represent sources of this type of GWs: binary black hole (BBH), binary neutron star (BNS), and neutron star - black hole binary (NSBH) [158]. Although all three source types share the same GW generation mechanism, each source is characterized by a specific GW pattern [158].

The coalescence includes three main phases: the inspiral, the merger, and the ringdown

[179]. The inspiral is a process that involves a pair of dense objects orbiting each other while emitting GWs and thus gradually losing their orbital energy [158, 179]. The orbital frequency gradually increases, as does the amplitude and frequency of the radiation, resulting in a characteristic chirp waveform [179]. Over millions of years, this process results in two objects progressively accelerating, getting closer together, and eventually colliding [158, 179]. The ground-based detectors cover the frequency range from about 15 Hz up to a few kilohertz, and thus only the final stage of the inspiral phase is observed, in addition to the merger and the ringdown phase [3, 158, 179]. The duration of the final stage of the inspiral is much shorter for the merging black holes than for the neutron stars [158]. The merger phase includes two objects merging to form a single black hole [179]. Finally, the resulting black hole radiates the extra energy during the ringdown phase and settles into the fundamental state [179].

The examples of the noise-free CBC GW signals obtained by the simulation models for each of the three source types mentioned above (BBH, BNS, and NSBH) are shown in Figure 4.1 for illustration purposes. Figure 4.1 shows a longer time interval for each simulated CBC signal example, including the inspiral, the merger, and the ringdown phase. In addition, a shorter time interval around the merger is also shown where the merger and the ringdown phase can be seen in more detail. The BBH, BNS, and NSBH signal examples were generated using the LIGO Algorithm Library (LALSuite) [163] and PyCBC [198, 261] waveform models (approximants) SEOBNRv4 [37], SEOBNRv4T [117], and IMRPhenomNSBH [248], respectively. Moreover, the BBH, BNS, and NSBH signal examples were simulated based on the mass and distance parameters of the observed real-life GW events GW150914 [3], GW170817 [3], and GW191219_163120 [10], respectively.

Burst GWs are generated by the short-duration unknown or unexpected sources [158, 249]. The detection of these GWs represents a challenging task due to many unknowns, including insufficient knowledge about underlying physics and waveforms generated [158]. One possible source of burst GWs are supernovae, i.e., the explosions of massive stars at the end of their life cycles [158, 179, 249].

Continuous GWs are considered to be caused by a single spinning massive object, such as a neutron star (remnants of a massive supergiant star), that deviates from the ideal spherical shape [158, 179, 249]. This type of GWs is characterized by having the constant amplitude and frequency due to the constant spin rate of its source [158, 249].

Stochastic GWs will be the most challenging type of GWs to detect [158]. Namely, stochastic GWs consist of numerous small GWs coming from all directions and different sources, including the GWs originating from the Big Bang [158, 227, 249]. These various GWs are then randomly mixed into a stochastic signal [158, 249].

Albert Einstein predicted the existence of GWs in 1916 within his general theory of relativity [80, 81, 82]. However, their existence was confirmed only mathematically or through indirect observations until 2015 [158]. On September 14, 2015, the Advanced

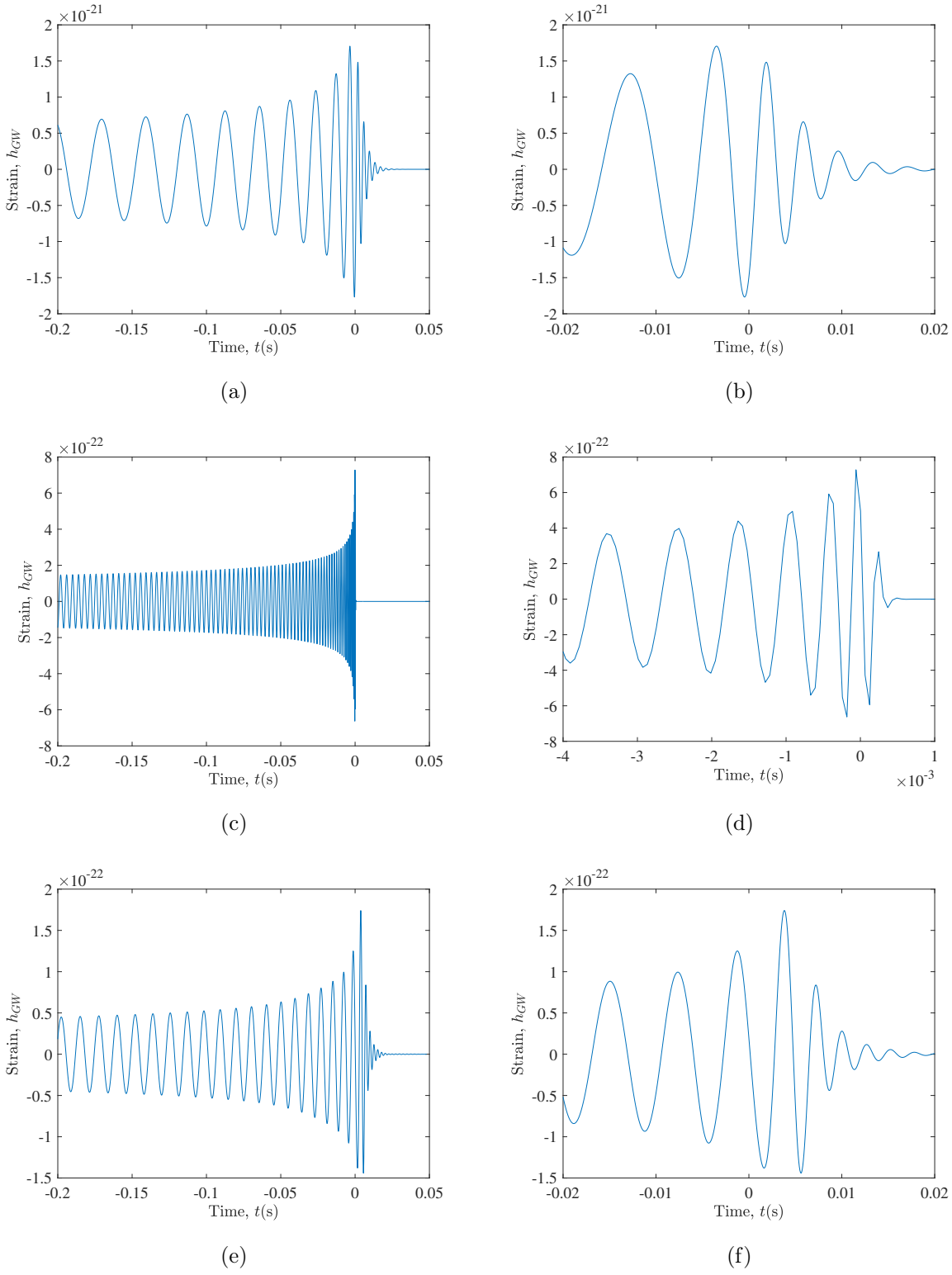
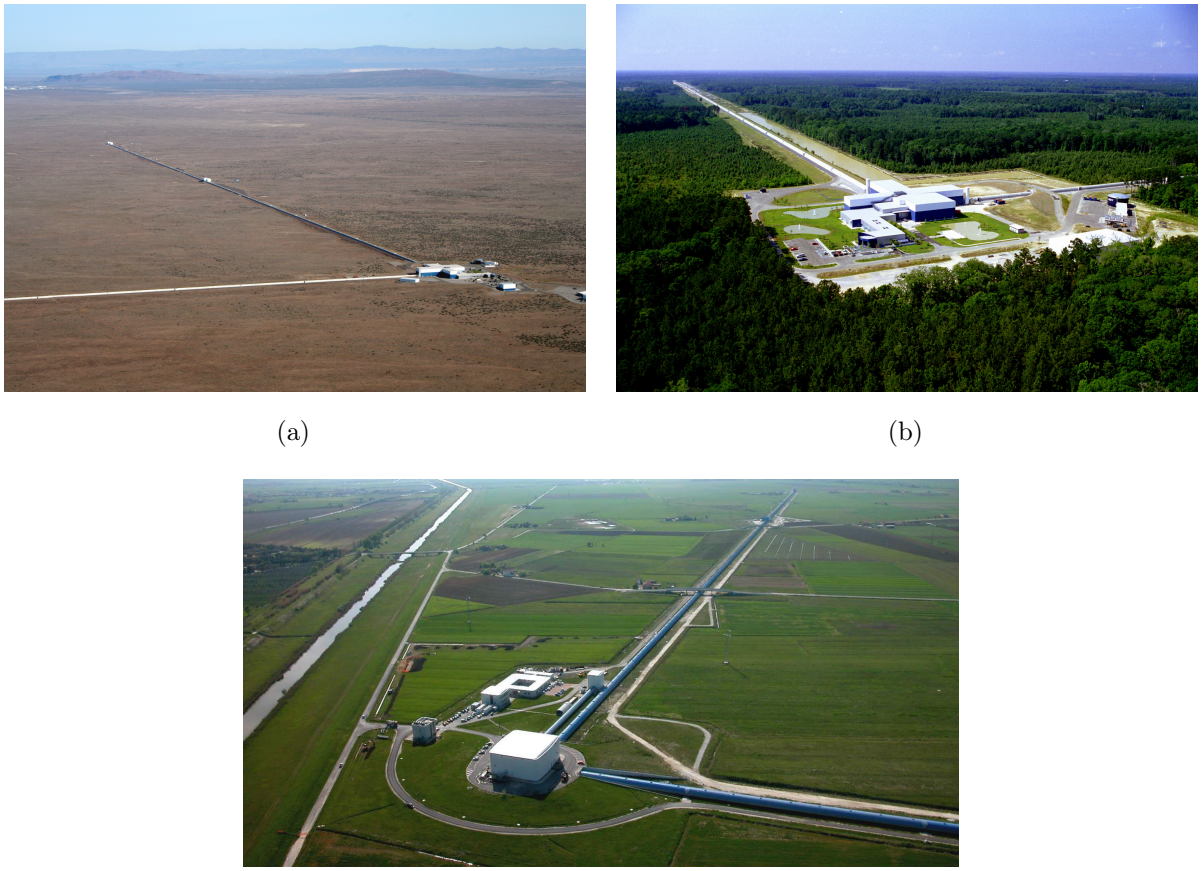


Figure 4.1 Examples of the simulated noise-free CBC GW signals: (a) BBH; (b) BBH - merger phase; (c) BNS; (d) BNS - merger phase; (e) NSBH; (f) NSBH - merger phase.

LIGO detectors [1] made the first direct detection of the CBC GW signal GW150914 [6]. This detection sparked intensive research in the GW data analysis, and the 2017 Nobel Prize in Physics was awarded to Professors Barry C. Barish and Kip S. Thorne

from the California Institute of Technology (Caltech) and Professor Rainer Weiss from the Massachusetts Institute of Technology (MIT), who were the leading scientists in the development of LIGO.

In addition to the two Advanced LIGO detectors located in the USA (Hanford, Washington and Livingston, Louisiana), the European Gravitational Observatory (EGO) runs the Advanced Virgo detector near Pisa, Italy [11, 12]. The aerial views of these three GW detector sites are shown in Figure 4.2. Moreover, two new GW detectors are expected to join the detection network, including LIGO-India and the underground Kamioka Gravitational Wave Detector (KAGRA) [15] in Japan that became operational in 2020.



(a) (b)

(c)

Figure 4.2 GW detector sites: (a) LIGO Hanford [160]; (b) LIGO Livingston [161]; (c) Virgo [162]. Courtesy Caltech/MIT/LIGO Laboratory and Virgo Collaboration.

The first observing run (O1) of the Advanced LIGO detectors ran from September 12, 2015, to January 19, 2016, and led to the detection of the GWs from three BBH mergers [3]. Additionally, the second observing run (O2) of the Advanced LIGO and Advanced Virgo detectors enabled the detection of seven BBH mergers and one BNS merger between November 30, 2016, and August 25, 2017 [3]. Moreover, the first part of the third observing run (O3a) of the Advanced LIGO and Advanced Virgo detectors reported 44 CBC GW candidate events in the period between April 1, 2019, and October 1, 2019 [8, 9]. Finally,

the second part of the third observing run (O3b) of the Advanced LIGO and Advanced Virgo detectors ran from November 1, 2019, to March 27, 2020. The analysis of the O3b data, released on November 7, 2021, within the third Gravitational-Wave Transient Catalog (GWTC), reported 35 CBC GW candidates, including the first confident observations of NSBH events [10].

In addition to the data obtained so far, a further increase in the GW detection rate is expected due to the improvements in the detector sensitivity and new detectors joining the network [7]. Due to the large amount of data collected, various specialized data processing algorithms have to be developed to identify the GW events embedded in high instrumental and environmental noise. The accurate identification of the GW events allows researchers to study the astrophysical properties of the GW sources, measure cosmological parameters, and compare the obtained results with those predicted by the general theory of relativity, to name a few.

4.2 Gravitational-Wave Detectors

The primary role of the GW detectors is to detect GW events in real-life noisy measurements and calculate their parameters. These detectors are L-shaped interferometers, with two light-reflecting mirrors placed at each end of the two orthogonal arms [157]. The length of each arm is 4 km in the Advanced LIGO detectors, while the Advanced Virgo detector consists of 3 km long arms [1, 11]. The layout of the GW interferometer is depicted in Figure 4.3. The incoming GW lengthens one arm of the interferometer while simultaneously shortening the other due to the space-time strain [6]. The interferometer measures the resulting difference in the arms' lengths $\Delta L(t)$, observed as the phase difference between the two returning laser beams and recorded by the output photodetector [6]. The acquired optical signal is proportional to the GW strain amplitude $h_{GW}(t)$, defined as [6]

$$h_{GW}(t) = \frac{\Delta L(t)}{L}, \tag{4.1}$$

where L is the length of the detector's arm.

Due to the very small magnitudes of the GWs that can be observed on Earth, the GW interferometers need to detect extremely small variations in the arms' lengths, even those up to 10 000 smaller than the width of a proton [157]. This requires building state-of-the-art measuring equipment with high measurement sensitivity and improving the standard structure of the base Michelson interferometer. First of all, the GW interferometers are much larger than the standard ones because longer interferometer arms allow the detection of smaller length changes [157]. Moreover, the GW interferometers include Fabry-Perot optical cavities formed by installing the additional mirrors in the detector's arms near the beam splitter [1, 6, 34, 157]. These mirrors enable multiple laser beam reflections, thus

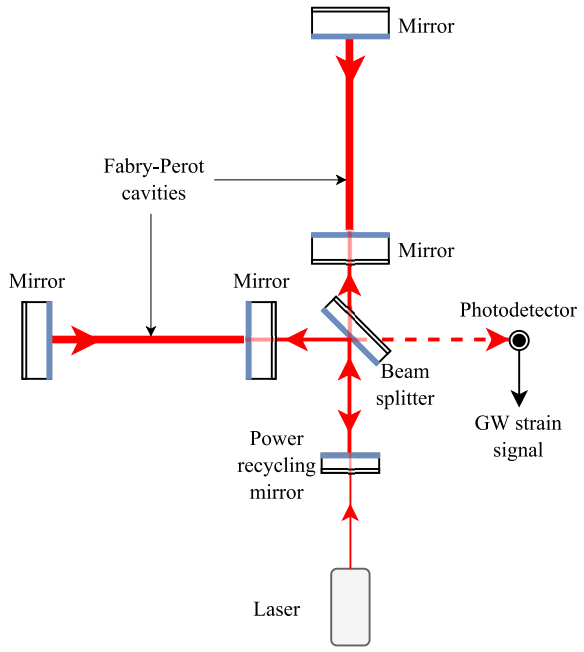


Figure 4.3 The layout of the GW interferometer.

extending the traveled distance and consequently increasing the measurement sensitivity [1, 6, 34, 157].

Furthermore, the resolving power of the detector is improved by increasing the input laser power using the power recycling mirrors that continuously reflect laser beams back into the interferometer's arms [1, 6, 157]. The obtained interference optical signal is additionally enhanced at the output by the signal recycling mirrors [1, 6, 157]. The high measurement sensitivity requires using the extremely precise 1064 nm Nd:YAG laser system in the ultrahigh vacuum, with the laser amplitude and frequency being stabilized to reduce the photon shot noise [1, 6, 146, 156].

In order to enable highly sensitive measurements in the interferometers, the influence of vibration, seismic, and thermal noise on the mirrors (often referred to as test masses in the literature) needs to be reduced as much as possible [6, 156]. The GW interferometers use two systems to protect against unwanted environmental vibrations: the active and passive damping systems [6, 156, 182]. The active damping system, also known as the internal seismic isolation system, includes the position and vibration sensors, the control system, and the actuators whose movements cancel the sensed environmental vibrations [156, 182]. On the other hand, the passive damping system consists of using the quadruple-pendulum system for each mirror [6, 156, 182]. The influence of the thermal noise on the measurements is reduced by designing the mirrors with materials characterized by low mechanical losses [105].

However, despite the above-described efforts directed towards noise reduction, various environmental and instrumental noise sources still affect the GW measurements. Figure 4.4 shows the amplitude spectral density (ASD) curves representative of the strain noise data

of the LIGO Hanford, LIGO Livingston, and Virgo detector during the O2 run. The data for the ASD curves representing the detectors' sensitivities reported in [3] were retrieved from [133, 134, 165].

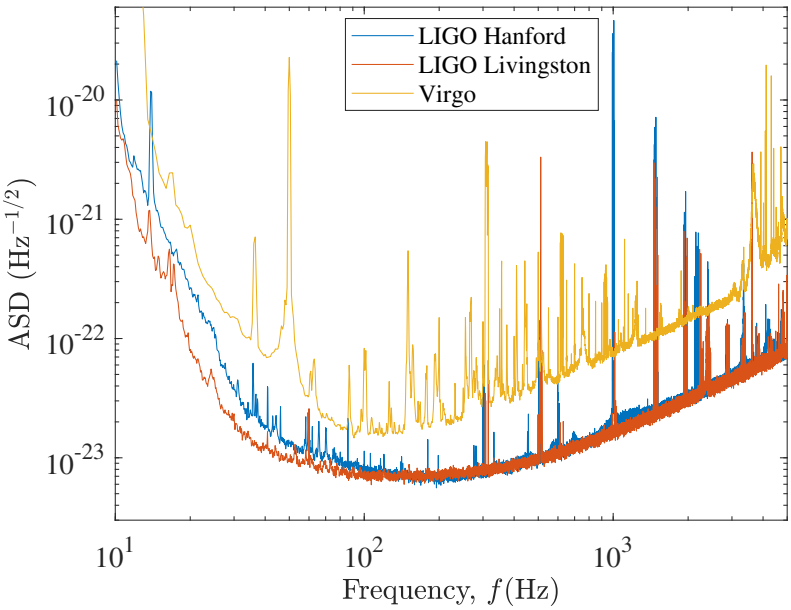


Figure 4.4 The representative ASD curves of the strain noise data of the Advanced LIGO and Advanced Virgo detectors during the O2 run.

The environmental noise sources include uncorrelated and correlated noise sources [5]. The uncorrelated noise sources include seismic noise caused by earthquakes at the frequency range from 0.03 Hz to 0.1 Hz, vibration and acoustic noise caused by human activity at the detector or the more distant locations, blip noise transients occurring between 30 Hz and 250 Hz, and magnetic noise sources [5, 79]. The correlated noise sources, representing those occurring in multiple GW detectors simultaneously, include electromagnetic noise sources, such as radio-frequency communication, lightning, and various solar events [5]. On the other hand, the instrumental noise sources include thermal noises manifesting as displacement noises, noise from the electrical power system (60 Hz in the USA and 50 Hz in Europe), calibration signals injected into the interferometer, electronic elements noise, mechanical resonances of the system, noises related to the laser and optical system, quantum noise, and gas noise [181].

Based on the above discussion, it is clear that the GW measurements are characterized by non-white, non-stationary, and non-Gaussian noise [5]. Moreover, due to the extremely low GW magnitudes observed at the ground-based detectors, the obtained GW signals are weak in comparison to the background noise, thus resulting in the measurements with very low SNR values. This makes the task of detecting GW signals in high, real-life noise rather challenging. Hence, the main research effort in the GW data analysis is the development of algorithms and methods for the detection of the GW events in the low SNR environments, with specific algorithms being developed for different types of GW signals. The rest of this

chapter gives a brief overview of the three main categories of the GW detection algorithms found in the recent literature (elaborated in the following three sections).

4.3 Matched Filtering-Based Detection of Gravitational Waves

Matched filtering techniques search for the known signal pattern in noisy data [116]. The application of matched filtering in the GW data analysis is based on correlating the noisy strain data obtained by measurements in the GW detectors to a large bank of waveform templates, i.e., reference signal waveforms, generated by GW simulations [227]. The main task is to find the optimal template (also referred to as a filter) that would maximize the correlation of the template with the noisy detector data when the GW signal is present in those data [227]. Thus selected template provides the maximum SNR [227].

It is clear from the described procedure that the databases of the simulated waveform templates need to be extensive enough to cover different astrophysical scenarios and ample parameter space. The matched filtering-based technique is primarily used in the Advanced LIGO and Advanced Virgo detectors to detect CBC GW candidates [199, 227]. Namely, these types of GW signals are thoroughly studied with established and highly accurate physical models.

The most recent GW catalog (GWTC-3) [10] described three matched filtering-based pipelines to search for CBC signals in the analyzed GW data: GstLAL [49, 50, 112, 185], Multi-Band Template Analysis (MBTA) [14, 23], and PyCBC [16, 17, 69, 70, 200, 261]. These pipelines were used for online and offline searches, i.e., low and higher latency searches [10]. Online analyses are performed in near-real-time as GW measurements are acquired, thus rapidly detecting GW candidates [10]. On the other hand, offline analyses are able to be more sensitive as they are conducted on the calibrated and cleaned GW data [10]. Moreover, offline analyses can afford to apply more computationally expensive algorithms [10].

The GstLAL search pipeline is based on matched filtering in the time domain and detects triggers for GW events using its waveform template bank [10, 185]. Triggers are generated by maximizing the matched-filter SNR (MF-SNR) over one-second time windows for each template and detector, with the defined MF-SNR threshold of 4.0 [10, 185]. The GW event candidates are formed by triggers time-coincident in more than one detector and generated by the same template [10, 185]. The event candidates are then ranked by the likelihood ratio statistic comparing the probabilities of the signal and noise hypotheses [10]. The likelihood ratio is calculated based on the MF-SNR from each GW detector, detector sensitivities, and time and phase differences between triggers [10].

The MBTA pipeline includes a preprocessing stage where GW data are downsampled,

and intervals with poor data quality are identified [10]. MBTA performs independent searches in three regions of its template bank, corresponding to the BBH, BNS, and NSBH sources [10]. Triggers occurring in a single detector are ranked based on the statistic taking into account the MF-SNR, the consistency with the template signal, and the local data quality [10]. On the other hand, triggers coincident in multiple detectors are ranked based on the statistic taking into account the calculated ranking statistics of single-detector triggers [10].

The PyCBC pipeline searches for triggers time-coincident in two or more GW detectors and calculates the ranking statistics [10]. Next, the calculated statistics are used to compute the significances, then combined into a single value [10]. The PyCBC statistics consider the ratio of the estimated signal event rate density and the noise event rate density [10, 70]. Moreover, there are two more PyCBC configurations used – PyCBC-BBH [71] and PyCBC Live [199], where the first one focuses on searching for BBH signals, while the second one is used for online searches [10].

Nevertheless, although efficient and widely used in pipelines for searching for CBC GW signals, the matched filtering technique has several drawbacks. First of all, matched filtering can be considered an optimal approach only in applications where signals are corrupted by Gaussian noise [116, 199]. However, as previously elaborated, the GW signals are embedded in high levels of non-stationary and non-Gaussian noise. Moreover, this approach is computationally expensive, and thus it is not the optimal solution for the real-time detection of the GW candidate events. Also, matched filtering is unsuitable for detecting burst and continuous GW events as it requires large databases of the expected signal waveforms. These databases are difficult to obtain for burst and continuous GWs because their numerical-relativity simulation models are highly computationally demanding or even imperfect in some cases due to limited knowledge of the potential GW sources and the underlying physical processes, thus rendering matched filtering ineffective [227]. Consequently, detecting these types of GWs requires the development of highly specialized algorithm pipelines [177, 218].

4.4 Denoising Techniques Applied to Gravitational Waves

Another research direction in GW data analysis includes denoising of the acquired noisy measurements. There were several studies on the application of machine learning-based techniques to GW data denoising. Various proposed denoising approaches included the application of the enhanced deep recurrent denoising auto-encoders on the noisy time series [231], the sparse signal reconstruction based on the dictionary learning algorithms [255], the subtraction of the non-stationary noise from the time-series GW data using deep

learning algorithms [202, 262], the removal of the noise using CNNs trained on the raw time-series data [271], and the subtraction of the noise transients (glitches) from the GW data using dictionary learning [252].

Moreover, denoising techniques are designed to remove noise, or reduce it as much as possible, without requiring *a priori* information on the underlying GW signals or the properties of their astrophysical sources. Thus, there has recently been a growing interest in adjusting noise removal methods and incorporating the existing efficient denoising methods into the advanced pipelines for GW data processing and analysis.

The recently proposed GW denoising approaches found in the literature focus on applying the total variation (TV)-based techniques. These techniques have been widely used in image processing, with the mathematical framework based on the Rudin-Osher-Fatemi variational model and the minimization of the L_1-norm [52, 53, 102, 221]. The TV-based denoising technique was applied in [254] to the simulated noisy GW signals, including BBH signals and burst signals originating from the core-collapse supernovae (CCSN). This study showed the efficiency of TV-based denoising in GW applications. Nevertheless, the analysis included only GW signals embedded in the additive Gaussian noise, which does not correspond to the real-life situation where non-stationary and non-Gaussian noise is present. Therefore, the study was expanded in [253] to include the simulated GW signals corrupted by the real-life noise data retrieved from the Advanced LIGO detectors. Moreover, the study in [253] also provided a detailed insight into selecting the optimal regularization parameter of the denoising model.

On the other hand, the application of the denoising technique combining the local polynomial approximation (LPA) and the RICI algorithm to the noisy burst GW signals was discussed by the author of this thesis and his collaborators in [174]. This locally adaptive and data-driven denoising technique uses the LPA as a filter design tool by fitting an appropriate polynomial to the noisy GW signal samples within a sliding window of varying size [139]. The optimal size of the sliding window is calculated by the algorithm based on the RICI rule [154] developed by Jonatan Lerga *et al.*, which represents an upgrade of the original intersection of confidence intervals (ICI) rule [138]. The study presented in [174] applied the LPA-RICI technique to the simulated CCSN burst GW signals injected into the actual noise data collected at the Advanced LIGO detector at the different SNR values. The higher orders of the LPA method were also considered and tested. The study showed that the LPA-RICI technique efficiently reduces the noise in the realistic GW data with the low SNR values while simultaneously preserving the characteristic signal morphologies. Furthermore, the LPA-RICI technique was shown to outperform several other denoising techniques applied to the noisy burst GW signals, including the LPA-ICI technique, the TV-based technique, and the wavelet-based techniques.

The study presented in [174] was extended in [176] by the author of this thesis and the collaborators by providing a detailed performance analysis of the ICI and RICI-based

denoising techniques applied to the burst GW signals in intensive noise and a discussion on the selection of the optimal algorithm's parameters. The utilization of the evolutionary metaheuristic optimization algorithms represents a potential solution for the efficient selection of denoising technique's parameters [173, 175].

The above-described denoising techniques are efficient tools for reducing the noise present in the GW strain measurements. These techniques can be utilized at the initial stages of the GW data analysis pipelines to reduce the noise and thus improve the opportunities for successfully detecting the GW events. However, denoising techniques do not represent standalone solutions for the GW detection but must necessarily be coupled with different classifiers (for example, machine learning-based methods).

4.5 Machine Learning-Based Detection of Gravitational Waves

As mentioned in Chapter 3, machine learning has attracted tremendous interest in recent years, with diverse applications in numerous research areas. The GW research is no exception to this trend, with many recent studies on the application of machine learning-based techniques, and deep learning-based techniques in particular, for various tasks in the GW data analysis [67]. These techniques are beneficial for GW data analysis due to their powerful algorithms' ability to generalize, the scalability, and the intensive computation process being concentrated in the initial training stage. Namely, once adequately trained, the machine learning and deep learning algorithms can provide real-time processing of the GW data at vastly reduced computational costs.

First of all, machine learning-based techniques have been used to estimate the GW source parameters [59, 60, 107, 195]. Most of these applications were based on the deep neural networks trained on the simulated time-series GW data. Moreover, as mentioned above, there were several studies on the application of machine learning-based techniques to GW data denoising [231, 255, 262, 271].

Another area of application of the machine learning-based techniques in the GW research includes classification of the glitches, i.e., the transient noise anomalies that occur in the GW detectors and heavily affect the quality of the obtained data. The study in [192] proposed using the difference-boosting neural network on the relative wavelet energy and entropy obtained by one-dimensional wavelet decomposition of the raw time-series GW data. The approach utilizing the machine learning algorithm, called Gaussian mixture model, for the unsupervised clustering of the wavelet coefficients obtained by the wavelet transform of the time-series glitch data, was presented in [208, 209]. The study in [28] provided a comparison of several machine learning algorithms used to classify the time-frequency spectrogram images obtained by the Q-transform [56] applied to the

glitch data. The tested machine learning algorithms included the multinomial logistic regression, the SVM, the CNNs, and the ensemble framework. In [215] and [283], the simpler CNN architectures were used to classify the spectrograms of the glitches, while the study in [99] reported on the use of transfer learning, i.e., several pretrained advanced deep CNN architectures (VGG [235], Inception [243], and ResNet [114]) were used for the same classification purpose.

Finally, the most studied area is the application of machine learning techniques, particularly those based on deep learning algorithms, to detect the GW events in the noisy background. There have been several studies on the detection of different types of GW signals. Some of these studies considered classifications based on the time-series GW data [54, 144, 228], while some utilized the two-dimensional spectrogram representations of the data [22, 130].

In [74], the deep neural network based on the modified one-dimensional version of the ResNet architecture was used to detect continuous GWs from spinning neutron stars by taking the frequency representation of the GW strain data as input. Based on the two-dimensional input vector containing the real and imaginary part of the fast Fourier transform of the data, the neural network classified the input data example as the noise or the GW signal embedded in the noise. Although using two input vectors, this architecture is still considered one-dimensional as each vector represents one input channel. This study was extended in [73] by considering a network of GW detectors and realistic noise conditions and testing several neural network architectures modified to accept multiple-channel one-dimensional inputs, including the ResNet and the Inception-ResNet [244] CNN architectures. In [144], the deep CNN was shown to successfully detect the BNS GW signals based on the input time-series strain data classification. Another study on detecting the BNS GWs based on the CNN and the simulated noisy time-series strain data was provided in [228]. Additionally, the detection of the CCSN burst GW signals was considered in [54], where the simulated time-series strain data were used as input to the CNN that distinguished between the two categories of the data examples: those containing the GW signal embedded in the noise and those containing only the noise.

Moreover, two-dimensional signal representations were also used for the machine learning-based detection of different types of GWs. The study in [186] provided insight into detecting the long GW signals from the isolated neutron stars. The GW detection in this study was set as a binary classification problem utilizing the CNN trained on the reduced time-frequency maps of the noisy strain data. On the other hand, the detection of the CCSN burst GWs was studied in [22] and [130]. The study presented in [22] proposed using the simple CNN for the binary classification of the time-frequency images obtained by the wavelet transform applied to the noisy simulated strain data. The spectrogram images from three GW detectors were combined into a single RGB image used as input to the CNN. In [130], the wavelet detection filter operating on the wavelet coefficients

obtained from the simulated strain data was proposed as an event trigger generator. After finding the GW candidates, the time-series data around the generated trigger times were used to calculate the spectrograms, then fed as input to the two-dimensional CNN that performed the classification.

Nevertheless, the research efforts have been primarily focused on applying machine learning-based techniques to detect the BBH GW signals. The reasons are their well-known models and the fact that the BBH signals represent the majority of the real-life GW signals that have been successfully detected by the Advanced LIGO and Advanced Virgo detectors so far.

The deep learning-based approach was introduced to the GW detection in [97] by adopting the one-dimensional deep CNN for the supervised binary classification of the raw time-series data examples containing the BBH signal templates embedded in the simulated Gaussian noise and those containing only the noise. The proposed CNN-based technique, named *Deep Filtering*, was applied to the 1 s long time-series data examples sampled at 8192 Hz. The technique was also shown to outperform several traditional machine learning classification algorithms, including the shallow neural network, the logistic regression, the naive Bayes, the random forest, the SVM, the hidden Markov model, and the k-nearest neighbors. This study was extended in [98] using a similar CNN architecture with four convolutional layers and two fully connected layers, trained on the time-series data containing the actual noise retrieved from the Advanced LIGO detectors instead of the simulated Gaussian noise utilized in [97]. The study in [85] reported an extension of the deep learning-based technique proposed in [97] to allow its application with a detection network formed by three GW detectors.

The study in [91] showed that the sensitivity of the conventionally applied matched filtering search could be closely reproduced by utilizing the deep CNN trained on the raw strain time series. This study considered the binary classification of the simulated BBH merger signals embedded in the synthetic additive Gaussian noise. A detailed analysis of the CNN-based binary classification approach to the detection of the BBH signals in the noisy strain time series was provided in [96], discussing its advantages, limitations, and potential challenges. Moreover, this study also proposed the extension of the binary classification approach in order to be utilized as a trigger generator for the GW events in the long stretches of the streaming real-life strain data acquired at the Advanced LIGO and Advanced Virgo detectors. The proposed technique was based on the fully convolutional CNN architecture, which does not contain fully connected layers and thus makes no assumptions on the size of the input data. Moreover, the data from multiple detectors were used as channels of a multidimensional input to the CNN. This study was an extension of the technique initially proposed in [94].

Another study on the CNN-based detection of the BBH signals was presented in [216] using a similar approach as those described above but providing a preliminary analysis

of using higher-order multipole waveforms as template signals. In [267], some minor adjustments were made to the conventional CNN architectures used in previous studies, such as those in [97] and [98], by adding a sensing layer and setting its coefficients according to some template waveforms used in matched filtering-based detection technique. The study in [167] proposed using Bayesian neural networks [196] for the detection of the BBH merger events in the noisy strain time series by integrating the Bayesian approach into the convolutional, long short-term memory, fully connected deep neural network (CLDNN) classifier [225]. The CLDNN combines the CNN, the long short-term memory RNN [122], and the deep neural network into a single architecture. In [272], the deep learning ensemble was proposed to detect the BBH signals in the strain time-series data. The proposed approach consisted of two WaveNet-based neural network models [263] simultaneously processing the data from two Advanced LIGO detectors.

As seen from the above overview, the studies on the machine learning-based detection of the BBH GW signals have dominantly focused on utilizing the time-series representation of the noisy strain data. However, the application of the deep learning techniques to the two-dimensional time-frequency representations of the strain data has not been adequately addressed in the literature. Namely, only three recent papers providing preliminary studies on the use of the CNN-based classification of the two-dimensional transformations of the GW data containing BBH signals have been found during the extensive literature review. However, these do not use two-dimensional transforms from Cohen's class, utilized in this thesis. Namely, in [184], the ResNet-50 CNN architecture was applied to classify the two-dimensional time-frequency spectrogram images obtained by the Q-transform [43] of the data examples with the noisy CBC GW events and the data examples containing only background noise. Moreover, the study in [19] proposed using the modified ResNet CNN architecture for classifying the spectrograms obtained by the Q-transform of the strain data containing only background detector noise and the data containing the BBH GW signals embedded in the noise. Two cases were considered: classification of the spectrogram images obtained from one GW detector and classification of the RGB spectrogram images produced by combining the spectrograms from each of three GW detectors forming a detector network. In the second case, the spectrogram image from a single detector represented one color channel of the final RGB image. Finally, in [190], the CNN was used for the binary classification of the reduced-size time-frequency images generated by the Morlet wavelet transform [191] of the strain time series with injected BBH GW signals.

In addition to the previous discussion, an extensive literature review has not identified any method of GW detection that would be based on the deep learning techniques using Cohen's class of TFDs of the time-series strain data as input.

Hence, this, as a novel approach, is proposed and analyzed in the thesis. The next chapter presents the details of the developed method for the detection of GW signals in noise.

CHAPTER 5

PROPOSED METHOD FOR DETECTING GRAVITATIONAL-WAVE SIGNALS BASED ON DEEP LEARNING AND TIME-FREQUENCY DISTRIBUTIONS

This chapter presents the method for detecting GW signals in intensive noise developed within the research presented in this thesis, recently published by the author in an international peer-reviewed journal [172]. The proposed method based on Cohen's class of TFDs and the deep learning algorithms is discussed in detail, starting with the description of the experimental setup. Next, each step of the proposed method and the utilized experimental setup is described in detail, including the data generation, the calculation of TFDs from Cohen's class, the input dataset, and the deep learning models used for classification. Moreover, the evaluation metrics and the statistical significance test used to analyze and interpret the obtained results are defined and explained.

5.1 Experimental Setup

The experimental setup, whose flowchart is provided in Figure 5.1, includes three main stages: the data preparation, the training and testing of the deep learning models, and the evaluation of the deep learning model performances.

The data preparation stage of the proposed approach includes data generation and

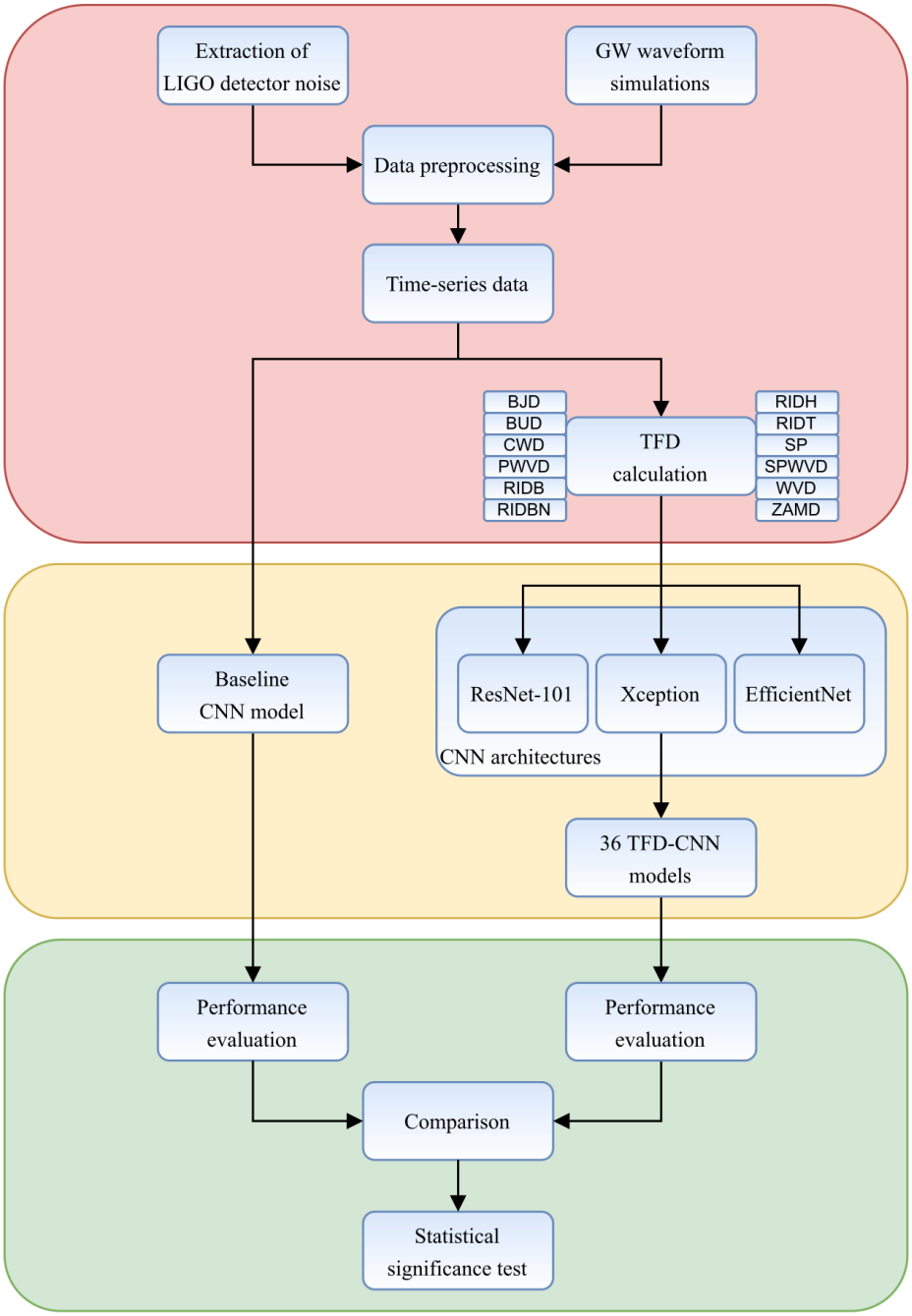


Figure 5.1 Experimental setup.

calculation of TFDs. The data generation procedure involves retrieving the real-life measurements from the LIGO detectors and the extensive GW simulations. The real-life LIGO data are used as the background noise in the generated data examples, while the synthetic GW signals obtained by the simulations are used as realistic models of the BBH merger waveforms. The generated dataset consists of 100 000 time-series data examples. After appropriate preprocessing, the time-series input data are obtained. Twelve different TFDs from Cohen's class are then calculated from the time-series data, resulting in 12

datasets, each containing 100 000 TFDs.

Next, the deep learning algorithms are applied to classify the TFDs, distinguishing those containing the GW signals in the background noise and those containing just noise. For this purpose, three different state-of-the-art deep CNN architectures (ResNet-101, Xception, and EfficientNet) are trained on 12 different TFD datasets, thus obtaining 36 different TFD-CNN combinations. The performance of each TFD-CNN model is evaluated on the test dataset and compared to the performance of the baseline model [98] trained on the original time-series input data. Finally, McNemar's statistical significance test is performed to verify the statistical significance of the obtained results.

The whole data generation procedure was conducted on the personal computer with the Intel Core i7-4720HQ CPU @ 2.60 GHz and 8 GB RAM. On the other hand, the TFD computations and the training and testing of the deep learning models were performed on the workstation with the Intel® Xeon® CPU E5-2620 v4 @ 2.10 GHz (two processors), 128 GB RAM, and three Nvidia® GeForce® RTX 2080 Ti GPUs.

In the following sections, each part of the proposed experimental setup is described in more detail.

5.2 Data Generation

Extensive and diverse datasets, representative of the studied phenomena, are required for the successful training and the proper evaluation of the deep learning models. Therefore, the dataset realistically modeling the physics of the BBH GW events was generated within this research. The data generation procedure consists of two key steps: collecting the actual data from the LIGO detectors and simulating the BBH merger waveforms to generate the synthetic GW signals. The approach to the GW dataset generation followed is similar to the one in [95, 96]. The utilized data generation procedure is visualized by the flowchart in Figure 5.2.

Many studies on GW detection use simulated Gaussian noise as background data. However, due to this simplified approach, in these cases, the background data do not contain characteristic detector glitches that represent some of the main challenges in the GW data analysis. Therefore, in this research, the real-life recordings from the LIGO detectors were used as the background noise, thus obtaining a more realistic model of the GW data recordings. The retrieved LIGO recordings were acquired during the O2 run [3]. The collected O2 data [164] were retrieved from the publicly available repositories at the Gravitational Wave Open Science Center (GWOSC) website [214].[1]

[1]"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) of the United Kingdom, the Max-Planck-Society (MPS), and the State of

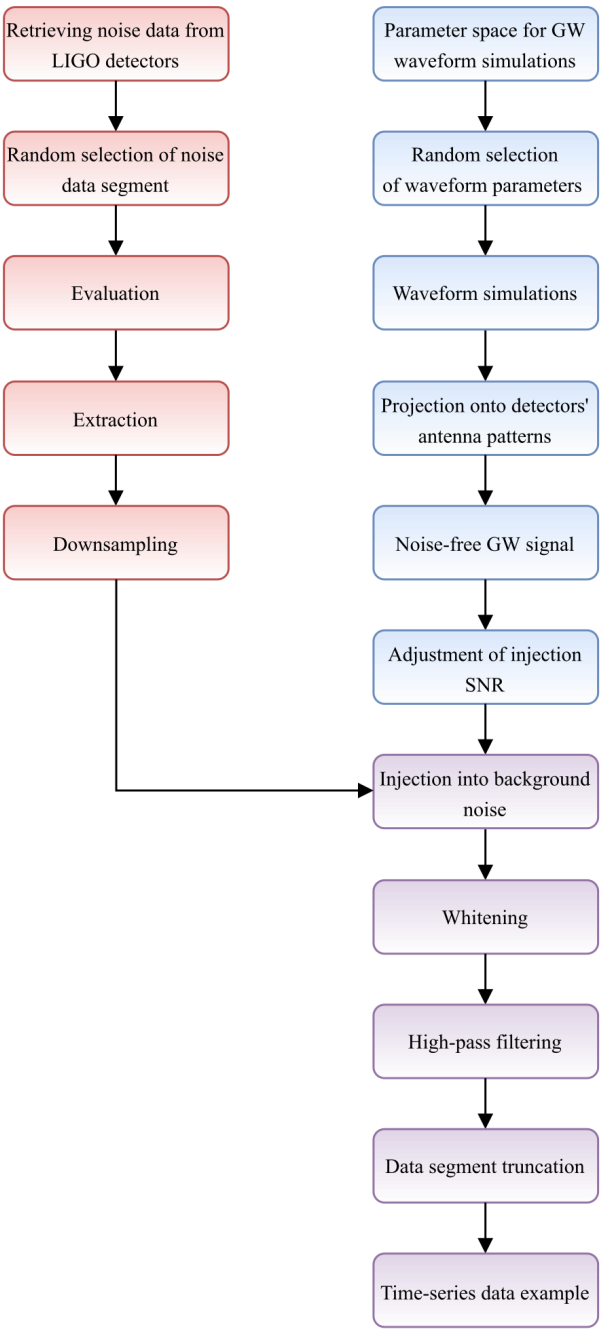


Figure 5.2 Data generation procedure.

For the purpose of this research, the data segments have to satisfy several criteria imposed on the retrieved LIGO data to be used as the background noise. The first criterion refers to the operating state of the LIGO detectors, where both the Hanford and Livingston detectors had to have data available during the considered time interval.

Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN), and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain."

Moreover, the considered data segment must satisfy all data quality requirements defined in LIGO for CBC searches. More precisely, the data segment must meet the minimum data quality level CBC-CAT3, as defined by the GWOSC. Furthermore, all data segments containing the confidently-detected real-life GW events are excluded based on the available GWTC. Finally, those data segments containing the hardware injections, i.e., the simulated transient signals emulating the real GW signals, injected into the LIGO measurement system for testing purposes, are also discarded. This way, it is ensured that the background data contain only the detector noise.

The background data selection procedure starts with randomly choosing a Global Positioning System (GPS) time value within the time interval defined by the start and end date of the O2 run. Next, the symmetric time interval of 16 s around the chosen GPS time is evaluated according to the above-defined criteria for the background data. In case the considered data segment meets the criteria, it is extracted for further processing. The original sampling frequency of the retrieved LIGO data is 4096 Hz. However, the extracted data segment is downsampled to the sampling frequency of 2048 Hz to reduce the computational cost and memory resources. According to the Nyquist-Shannon sampling theorem, the sampling frequency of 2048 Hz allows the reconstruction of frequencies up to 1024 Hz. Considering that BBH signals generally occur in the frequency range of up to a few hundred Hertz, the obtained Nyquist frequency of 1024 Hz allows their detection.

In parallel with the above-described extraction of the real-life LIGO data, extensive simulations of the BBH merger waveforms are conducted. This parallelization of the data generation procedure enables the reduction of the required execution time. The simulations of the BBH waveforms are performed using the LALSuite [163], and the PyCBC software package [198, 261]. LALSuite is a codebase containing all GW waveform simulation models and GW data analysis algorithms and pipelines used in the Advanced LIGO and Advanced Virgo analyses. PyCBC is a Python-based software package including algorithms for the detection of CBC GW events and the measurement of the parameters of their astrophysical sources. The algorithms provided by these two software packages are routinely used for the analysis of the GW data by the research groups within the LIGO and Virgo collaborations.

The simulations are conducted using the state-of-the-art effective-one-body (EOB) waveform model SEOBNRv4 in the time domain [37]. The utilized SEOBNRv4 model is suitable for simulations of the spinning, non-precessing BBH events. Parameter space is defined for the simulations according to the recommendations, and the characteristic parameter values of the detected real-life GW events [96]. The parameter values are independently and uniformly drawn at random from the defined distributions for each waveform simulation. There are three main groups of parameters: parameters describing the astrophysical source of the waveform, parameters describing the source's position and orientation in the sky, and a parameter determining the distance of the source. The parameters describing the characteristics of the source include masses and z-components

of the spins of the merging black holes. Moreover, the parameters describing the source's position and orientation in the sky include polarization, right ascension, declination, coalescence phase angle, and inclination. Finally, the distance of the source is modeled by injection SNR representing the desired network optimal matched-filter SNR (NOMF-SNR), whose definition will be given later in this section.

The masses of the two merging black holes are for the simulations independently and uniformly drawn from the range between 10 and 80 solar masses, whereas the z-components of the black hole spins are sampled from the range of $0 - 0.998$. Moreover, the polarization angle, representing one of the three Euler angles that connect the detector's reference frame with the radiation frame (the reference frame in which the GW propagates in the z-direction) [96], is drawn from the range of $0 - 2\pi$. Furthermore, the right ascension and declination parameters, which determine the source's position in the sky [96], are sampled jointly from the uniform distribution over the sky. In addition, the coalescence phase and the inclination, which represent the angles defining the detector's location in the sky as seen from the source reference frame whose z-axis is perpendicular to the plane with the two black holes orbiting each other [96], are taken together from the uniform distribution over the sphere. Finally, the desired injection SNR (the desired NOMF-SNR), related to the distance between the source and the detector [96], is taken at random from the range of $8 - 30$ dB. This range was chosen for the simulations based on the network matched-filter SNR (NMF-SNR) values of the real-life GW events observed in the Advanced LIGO and Advanced Virgo detectors during the O1, O2, and O3a runs whose data were available at the time of research done in this thesis.

The waveforms obtained by the LALSuite simulations are then windowed by the one-sided Tukey (tapered cosine) window [257] to suppress any potential amplitude discontinuities in the simulated waveforms. Each waveform consists of two time-series signals: h_+ and $h_\times$, where h_+ represents the + (plus), and $h_\times$ the $\times$ (cross) tensor polarization mode of the GW. These two signals are then projected onto the antenna patterns of the Advanced LIGO detectors based on the source location in the sky and the polarization angle using the available PyCBC functions. The antenna patterns define the directional sensitivity of the GW detector, i.e., the GW signal is detected with a different amplitude and phase in each detector due to their relative position and orientation [178]. Thus, noise-free detector signals are obtained.

The simulated noise-free GW signals are then injected into the selected real-life background noise to produce the data examples containing the GW signals in the noise. In contrast, the data examples containing only the background noise are obtained by skipping the signal injection step.

The simulated GW signals need to be appropriately scaled before being injected into the noise, so the desired injection SNR is obtained for each data example. The procedure for adjusting the injection SNR consists of several steps based on the calculation of the

MF-SNR. The MF-SNR is conventionally applied as a metric in all GW data analyses found in recent studies. For the matched-filter template $h_t(t)$ and the strain $s_{GW}(t)$ measured by the GW detector:

$$s_{GW}(t) = h_{GW}(t) + n(t), \tag{5.1}$$

where $h_{GW}(t)$ represents the GW signal and $n(t)$ the detector noise, the complex matched-filter output $z_{MF}(t)$ is defined as [17]

$$z_{MF}(t) = 4 \int_0^\infty \frac{S_{GW}(f) H_t^*(f)}{S_n(f)} e^{j2\pi ft} df, \tag{5.2}$$

where $S_{GW}(f)$ is the Fourier transform of the measured GW strain $s_{GW}(t)$, $H_t(f)$ is the Fourier transform of the filter template $h_t(t)$, and $S_n(f)$ is the estimated one-sided power spectral density (PSD) of the detector noise.

The MF-SNR $\rho_{MF}(t)$ is then obtained as [17]

$$\rho_{MF}(t) = \frac{|z_{MF}(t)|}{\sigma_{MF}}, \tag{5.3}$$

where σ_{MF} is the normalization constant representing the sensitivity of the detector. The normalization constant is calculated for each template $h_t(t)$ as [17]

$$\sigma_{MF}^2 = 4 \int_0^\infty \frac{|H_t(f)|^2}{S_n(f)} df. \tag{5.4}$$

The NMF-SNR ρ_{NMF} can then be obtained by combining the MF-SNR of each detector in the detector network as [68]

$$\rho_{NMF} = \sqrt{\sum_i \rho_{MF_i}^2}. \tag{5.5}$$

The optimal matched-filter SNR (OMF-SNR) represents the maximal MF-SNR obtained using the time-inverted GW signal itself as a filter template [68, 236]. The NOMF-SNR is then calculated based on the OMF-SNR values of each detector using the same principle as in (5.5).

The definitions provided above are used for adjusting the injection SNR in the data generation procedure. First, the selected noise segment and the simulated GW signal are added together. Next, the OMF-SNR of the injection is calculated for both LIGO detectors. The calculated OMF-SNR values are used to compute the NOMF-SNR. The GW signal is then scaled by the ratio of the desired injection SNR and the calculated NOMF-SNR. Finally, the scaled GW signal is injected into the background noise, thus ensuring the desired NOMF-SNR of the data example. The desired NOMF-SNR range of $8 - 30$ dB set during the simulations resulted in the OMF-SNR range of $0.10 - 30.46$ dB

for the data examples from the Livingston detector, whose data were employed to generate the dataset further used in the experimental procedure.

The generated time-series data examples are then preprocessed using the established whitening and filtering procedures. First, the strain time-series data are whitened by weighting their Fourier transforms with the inverse of the local estimate of the ASD (the square root of the PSD) of the detector noise [4]:

$$S_{GW,w}(f) = \frac{S_{GW}(f)}{\sqrt{S_n(f)}}, \tag{5.6}$$

where $S_{GW,w}(f)$ denotes the Fourier transform of the whitened strain time series $s_{GW,w}(t)$.

The whitening procedure ensures the equal significance of the data in each frequency bin by down-weighting the frequencies with high noise. By suppressing the extra noise at the low frequencies and the specific frequencies with the instrumental spectral lines, the whitening allows the weak GW signals to be detected more easily. The local PSD estimate is obtained by the Welch method [274] that divides the data into the overlapping windowed segments, calculates the Fourier transforms on these segments, and then averages the power spectrum obtained as the squared modulus of the Fourier transform of each segment. This study applies the Welch method to the 16 s long data examples with the Fourier transforms calculated on the overlapping 4 s long windowed segments, each spaced by 2 s.

The whitened data are then transformed back to the time domain ($s_{GW,w}(t)$) using the inverse Fourier transform and high-pass filtered at 20 Hz using the finite impulse response filter. The high-pass filter is used to remove the low-frequency artifacts that the simulations might have generated. Moreover, the LIGO detectors are not calibrated for frequencies below 10 Hz, and the frequencies below approximately 20 Hz are removed in GW data analyses due to high-amplitude noise [4].

Both edges of the data examples obtained after the above-described preprocessing are corrupted by calculating the Fourier transform included in the whitening procedure. Therefore, the data examples are truncated to discard the corrupted data at the edges. The 0.5 s length of the truncated data examples was chosen to reduce the required memory resources and computational cost of training the deep learning models. Nevertheless, the chosen data example length is sufficient for the proper detection of BBH signals as these signals are characterized by short duration, i.e., the most prominent part of the BBH signal representing the coalescence is expected to be contained within the chosen example length. Moreover, for each data example, the amplitude peak of each BBH signal is randomly placed within the $0.1 - 0.4$ s interval to eliminate any possibility of the deep learning classification models' overfitting due to the same location of the signals in the data examples.

The above-described data generation procedure finally resulted in 100 000 time-series

data examples, each with a length of 0.5 s (1024 samples). Half of the generated time-series data examples contain GW signals in the real-life noise, while the other half contains no GW signals, i.e., only the noise is present. Considering the standard SNR, defined as the ratio of the signal power to the noise power, the SNR values of the 0.5 s long raw, noisy time-series data examples with injected GW signals range from -123.46 to -2.27 dB. Figure 5.3 shows the histogram of the SNR values of the raw, noisy time-series data examples, i.e., the time-series data examples prior to applying the whitening and high-pass filtering procedures.

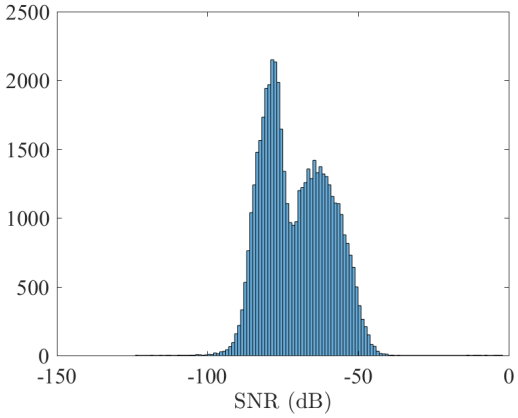
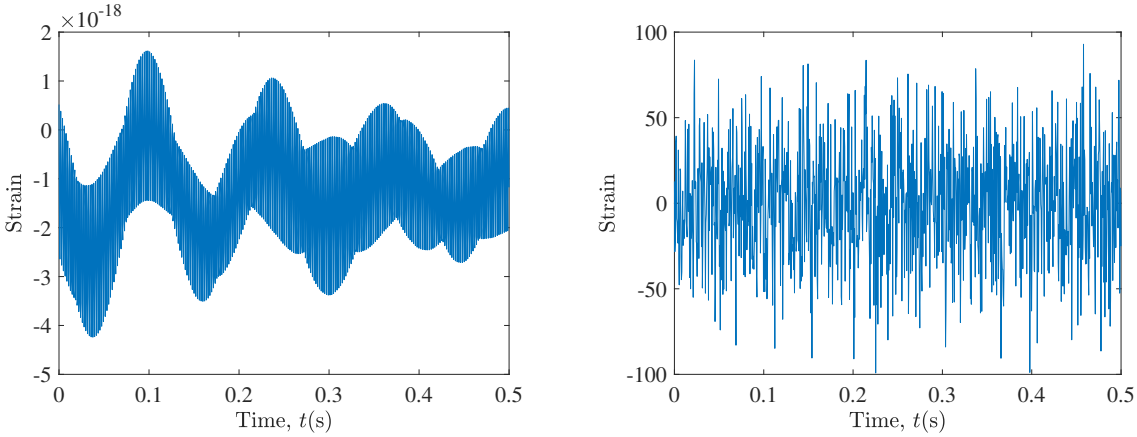


Figure 5.3 Histogram of the SNR values of the raw, noisy time-series data examples.

In order to illustrate the results of the data generation procedure, the following figures show examples of the time-series strain data containing only the real-life detector noise and those containing the GW signals injected into the noise. First, Figure 5.4 shows a randomly chosen time-series data example containing the noise only, including the plots of the raw time series, i.e., the time series obtained from the GW detector, and the whitened and high-pass filtered time-series.

Next, Figures 5.5, 5.6, and 5.7 show three time-series data examples obtained by setting the desired NOMF-SNR during the data generation procedure to the respective values of 8, 19, and 30 dB. These values were chosen for illustration purposes as they represent the minimum, mean, and maximum values of the $8-30$ dB range utilized for the data generation. The desired NOMF-SNR values of 8, 19, and 30 dB for these examples resulted in the single detector's OMF-SNR values of 6.55, 14.92, and 25.82 dB, respectively. Moreover, the respective SNR values of -85.24, -81.46, and -60.12 dB were obtained for these raw, noisy time-series data examples. Each provided figure contains the plots of the raw noise time series obtained from the GW detector, the noise-free GW signal obtained by the simulations, the GW signal injected into the noise, and the noisy GW signal after applying the whitening and high-pass filtering procedures.

As shown in Figures 5.5, 5.6, and 5.7, the GW signal amplitudes are several orders of magnitude less than the noise amplitudes. Thus, it is impossible to visually detect the

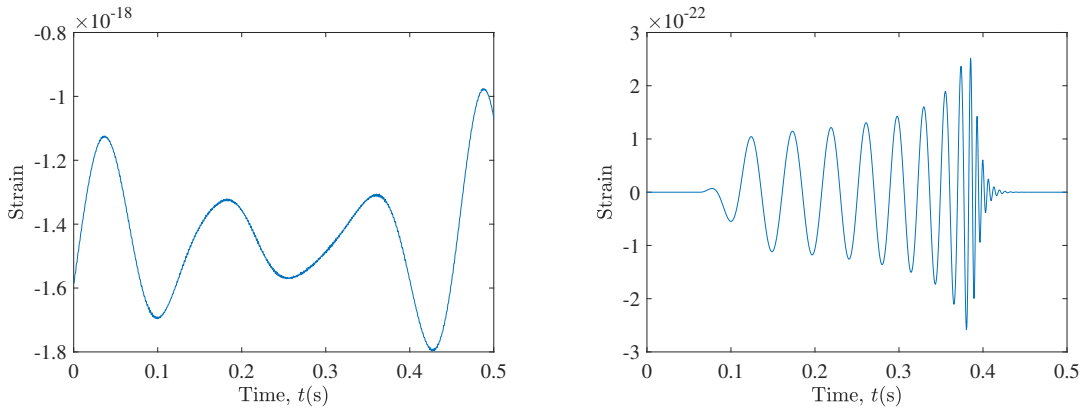


(a) (b)

Figure 5.4 Time-series data example containing only noise: (a) Raw noise; (b) Whitened and high-pass filtered noise.

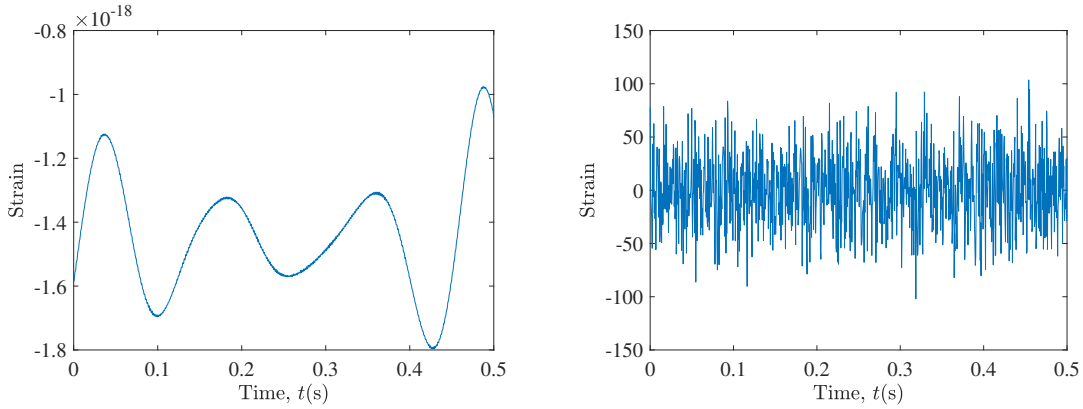
GW signal in the background noise from the plots of the raw time-series data obtained by the measurements, even at higher NOMF-SNR values. Moreover, it can be seen that, even after the whitening and high-pass filtering, the detection of GW signals in the noisy background using the time-series data, i.e., distinguishing between the time-series data examples containing GW signals from those containing only noise, represents a rather challenging task.

The time-series data examples obtained by the data generation procedure described in this section are also transformed into two-dimensional Cohen's class TFDs, as described in the next section.



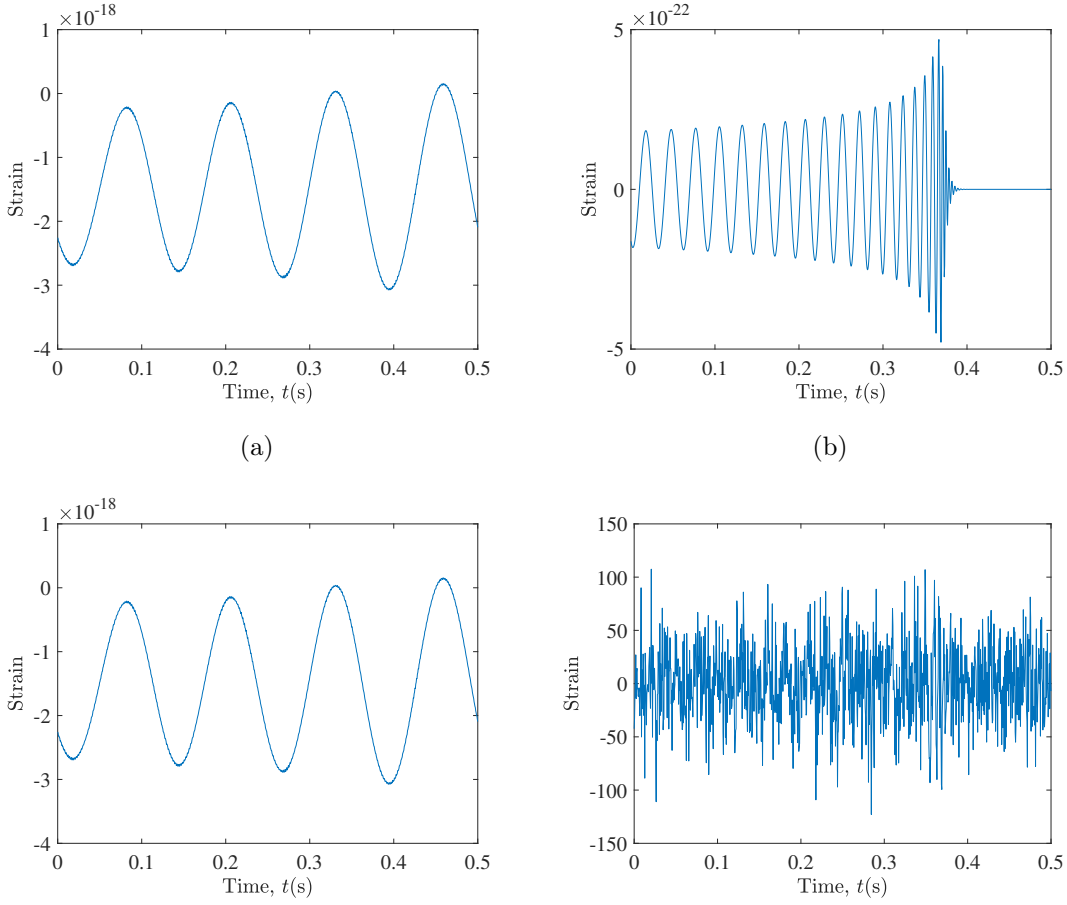
(a) (b)

Figure 5.5 Time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 8 dB (OMF-SNR = 6.55 dB, SNR = −85.24 dB): (a) Raw noise; (b) Noise-free GW signal; (c) Noisy GW signal; (d) Whitened and high-pass filtered noisy GW signal.



(c) (d)

Figure 5.5 (cont.) Time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 8 dB (OMF-SNR = 6.55 dB, SNR = −85.24 dB): (a) Raw noise; (b) Noise-free GW signal; (c) Noisy GW signal; (d) Whitened and high-pass filtered noisy GW signal.



(a) (b)

(c) (d)

Figure 5.6 Time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 19 dB (OMF-SNR = 14.92 dB, SNR = −81.46 dB): (a) Raw noise; (b) Noise-free GW signal; (c) Noisy GW signal; (d) Whitened and high-pass filtered noisy GW signal.

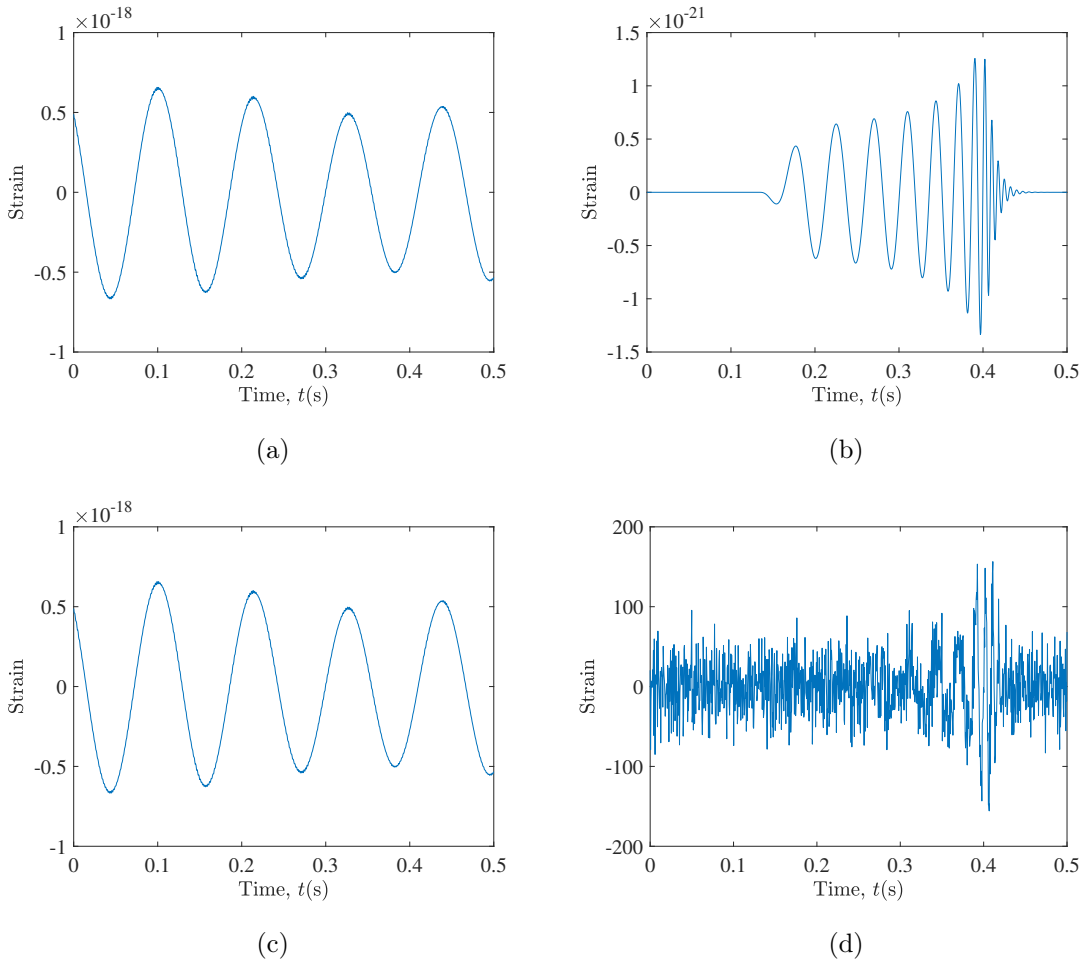


Figure 5.7 Time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 30 dB (OMF-SNR = 25.82 dB, SNR = −60.12 dB): (a) Raw noise; (b) Noise-free GW signal; (c) Noisy GW signal; (d) Whitened and high-pass filtered noisy GW signal.

5.3 Cohen's Class Time-Frequency Distributions

Twelve TFDs from Cohen's class, described in Chapter 2, have been applied to the 100 000 generated time-series data examples. First, the Hilbert transform is applied to these time series to obtain their analytic form. Next, the following TFDs are computed: BJD, BUD, CWD, PWVD, RIDB, RIDBN, RIDH, RIDT, SP, SPWVD, WVD, and ZAMD.

Thus, twelve different datasets are obtained, one for each TFD. Each dataset comprises 100 000 TFDs, resulting in a total of 1.2 million TFDs. Each TFD in the dataset contains information about the time-frequency representation of the input time-series data example, with the time range of $0 - 0.5$ s, and the frequency range of $0 - 1024$ Hz. For calculation of considered TFDs, Hamming windows were utilized. The TFD computations were implemented using the Time-Frequency Toolbox [25, 26] in MATLAB® R2017a.

Figure 5.8 shows the twelve TFDs obtained for the randomly chosen whitened time-series data example containing only noise, whose time-series form is shown in the previous section in Figure 5.4. Moreover, Figures 5.9, 5.10, and 5.11 show the TFDs calculated on the three whitened time-series data examples containing the GW signals embedded in the background noise. These data examples correspond to the desired NOMF-SNR values of 8, 19, and 30 dB, and their time-series forms are provided in the previous section in Figures 5.5, 5.6, and 5.7, respectively.

As shown in the provided figures, TFDs of Cohen's class provide improved intelligibility of the signal representation compared to the original time-series signal form, i.e., the characteristic chirp pattern generated by the GW signal can be detected more easily from the time-frequency representation than from the time domain, especially at higher NOMF-SNR values, such as those shown in Figures 5.10 and 5.11. Thus, the utilization of Cohen's class TFDs in combination with deep learning algorithms is expected to provide high classification performance. The improved performance is expected even for the data examples with low NOMF-SNR values, such as the one shown in Figure 5.9, where it is more challenging to detect the characteristic GW pattern in the background noise, i.e., distinguish the data examples containing GW signals from those containing just noise.

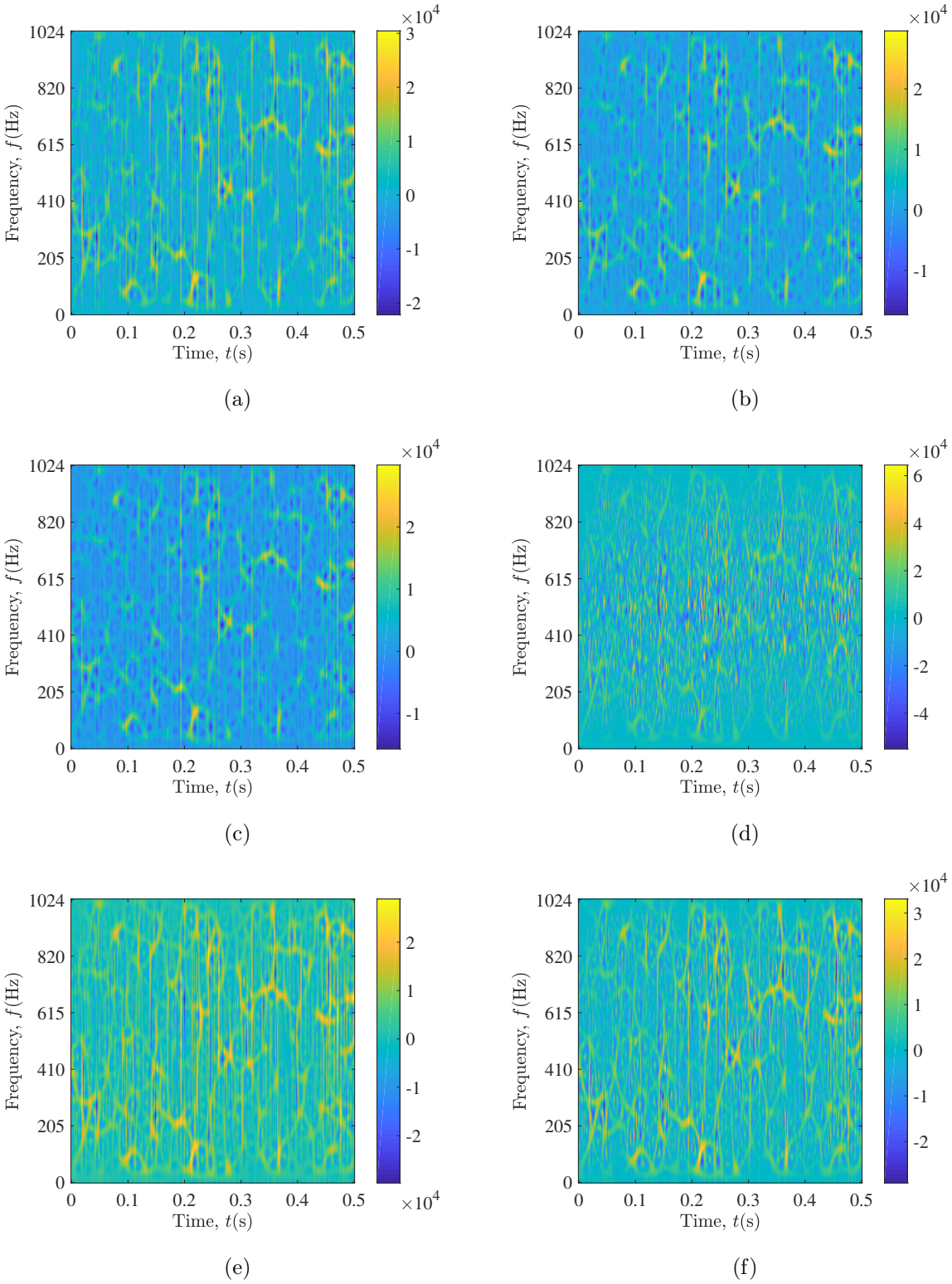


Figure 5.8 TFDs of the time-series data example containing only noise: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

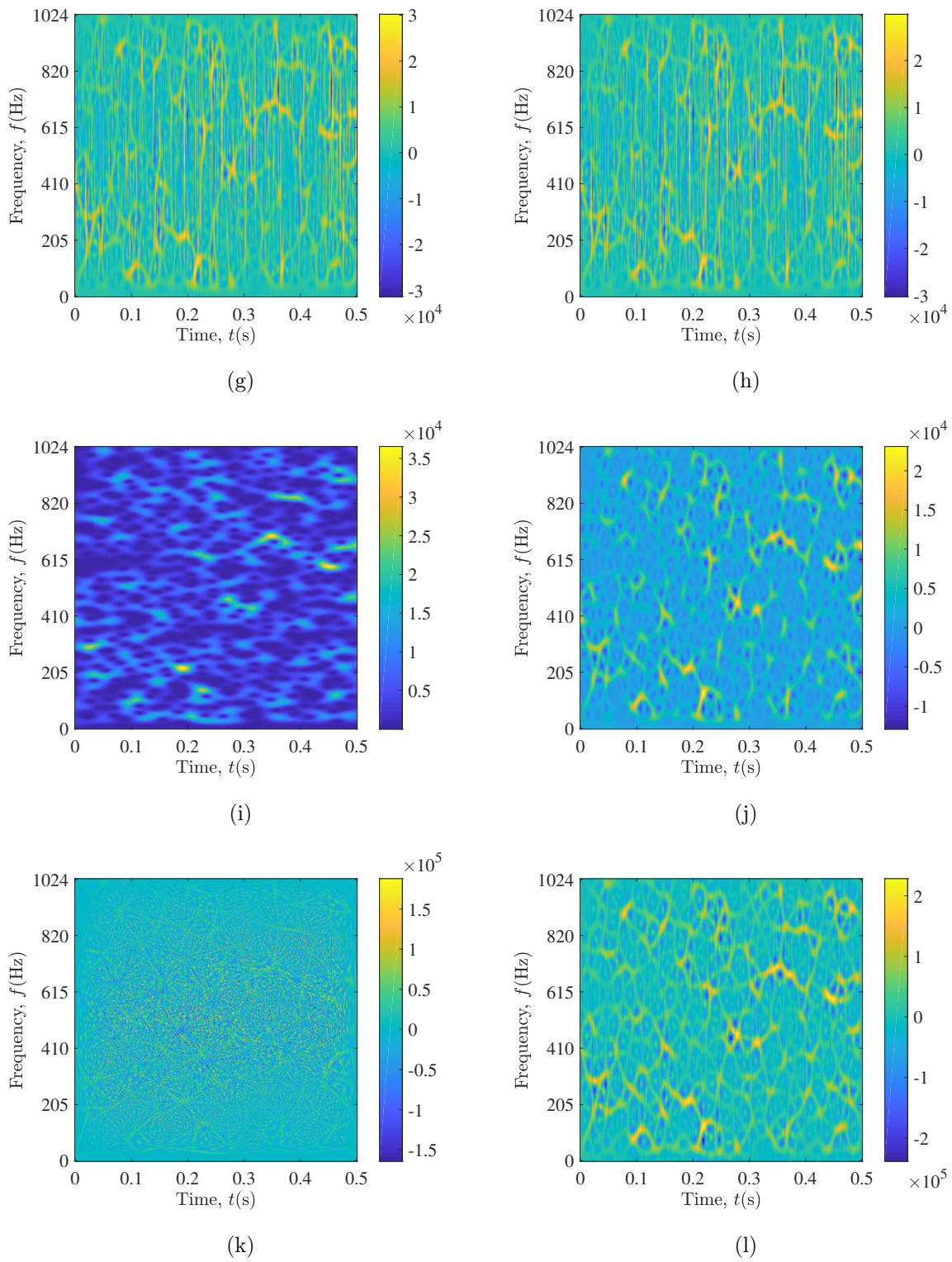


Figure 5.8 (cont.) TFDs of the time-series data example containing only noise: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

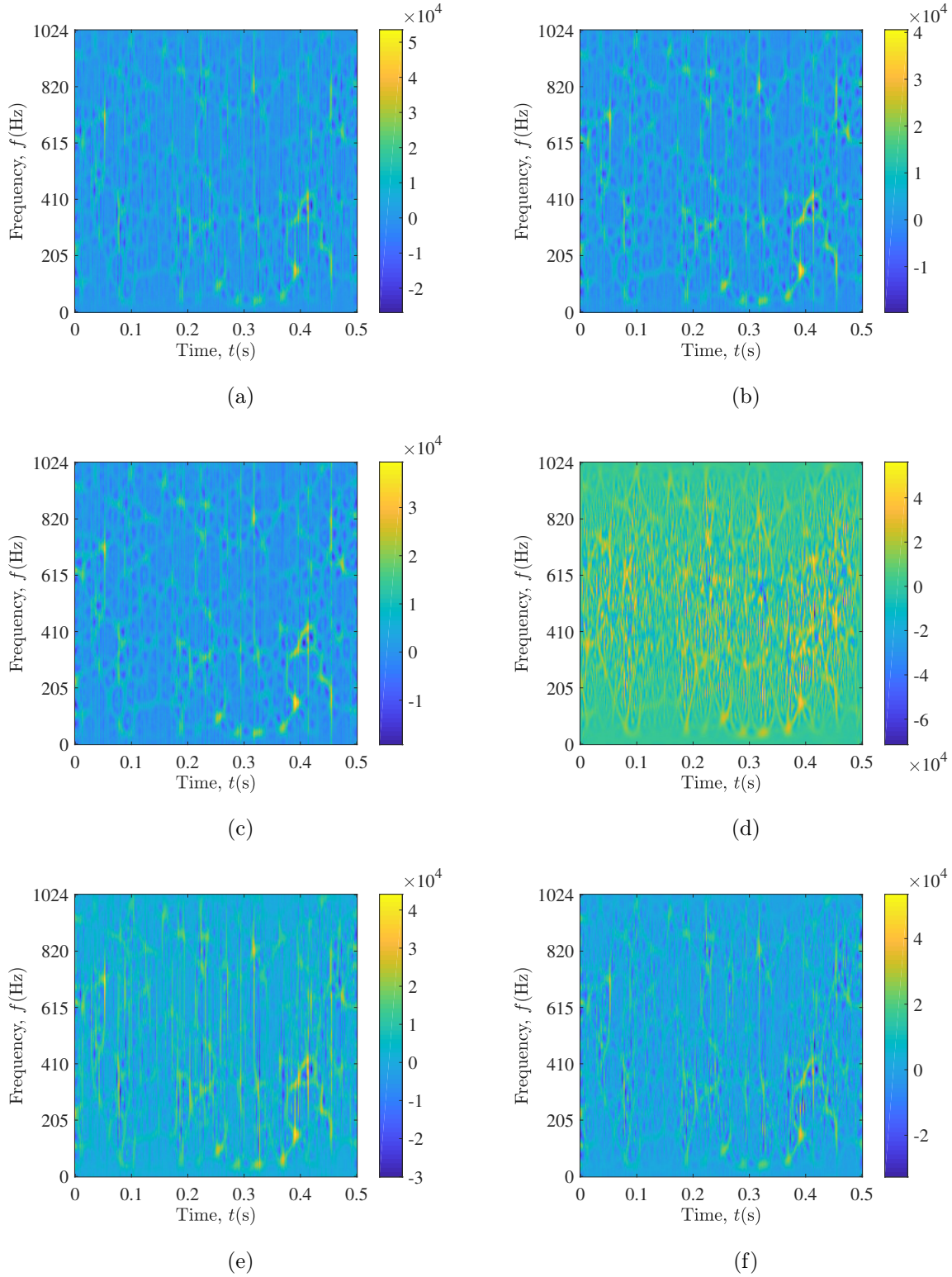


Figure 5.9 TFDs of the time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 8 dB (OMF-SNR = 6.55 dB, SNR = −85.24 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

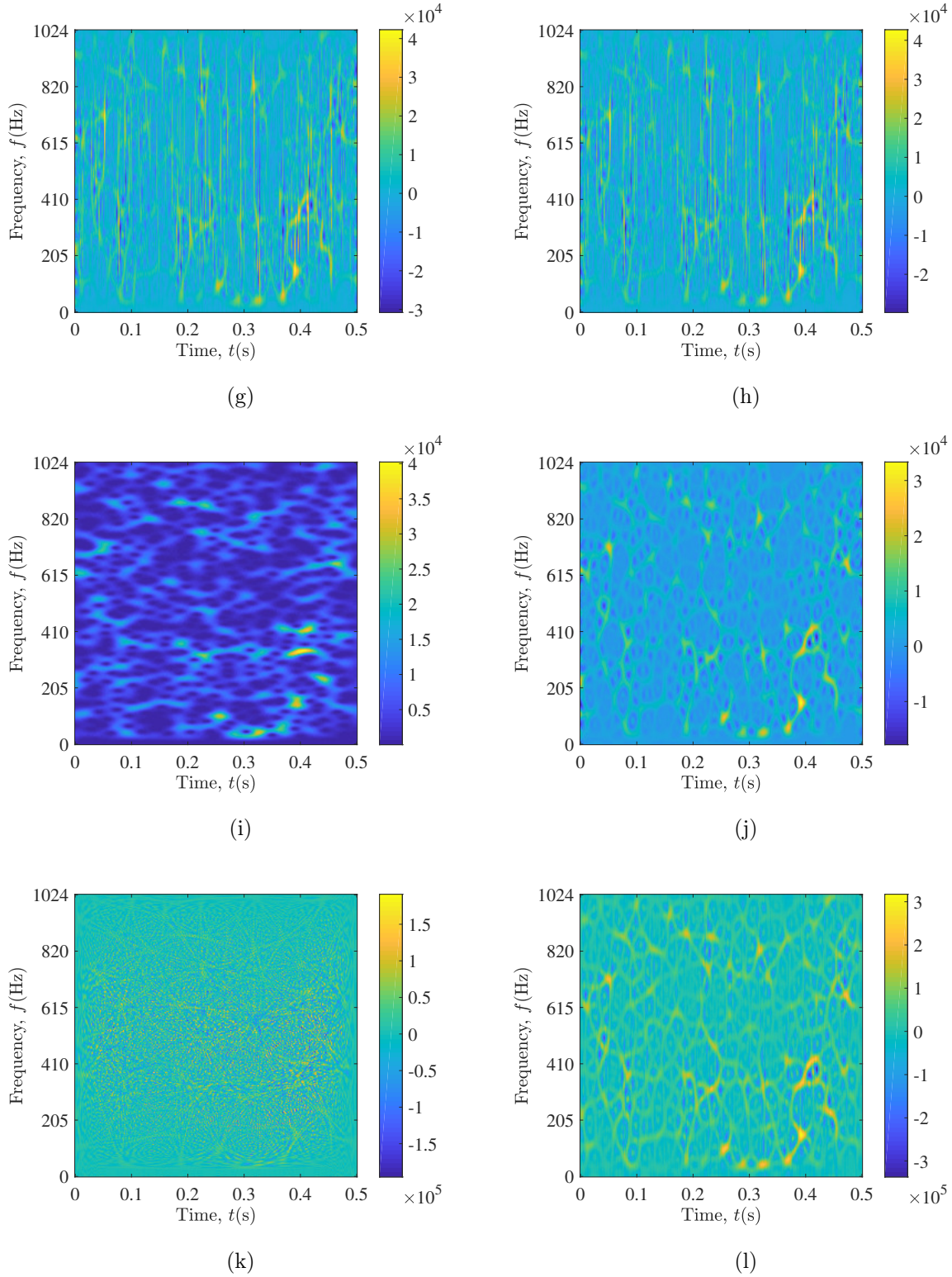


Figure 5.9 (cont.) TFDs of the time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 8 dB (OMF-SNR = 6.55 dB, SNR = −85.24 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

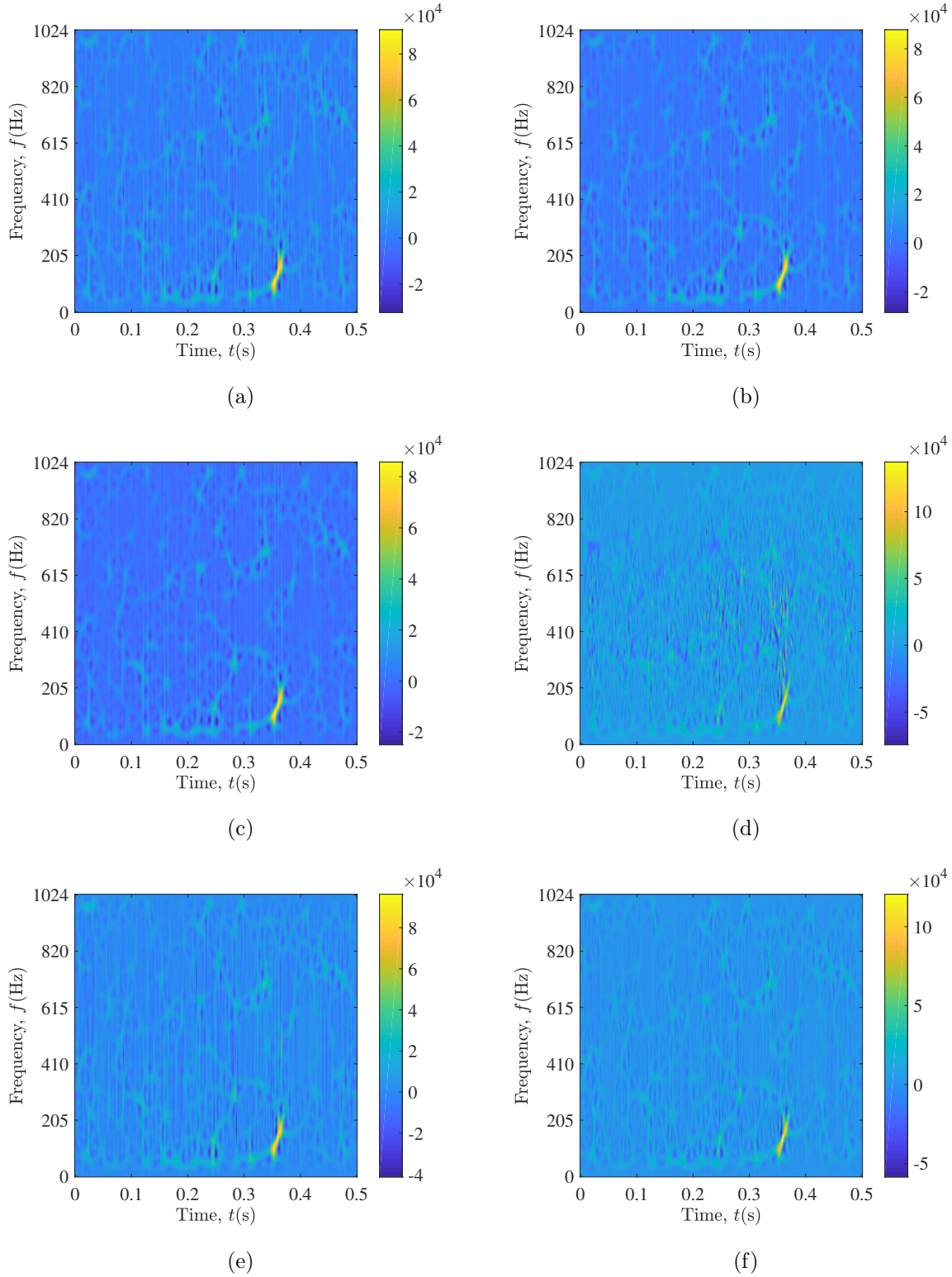


Figure 5.10 TFDs of the time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 19 dB (OMF-SNR = 14.92 dB, SNR = −81.46 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

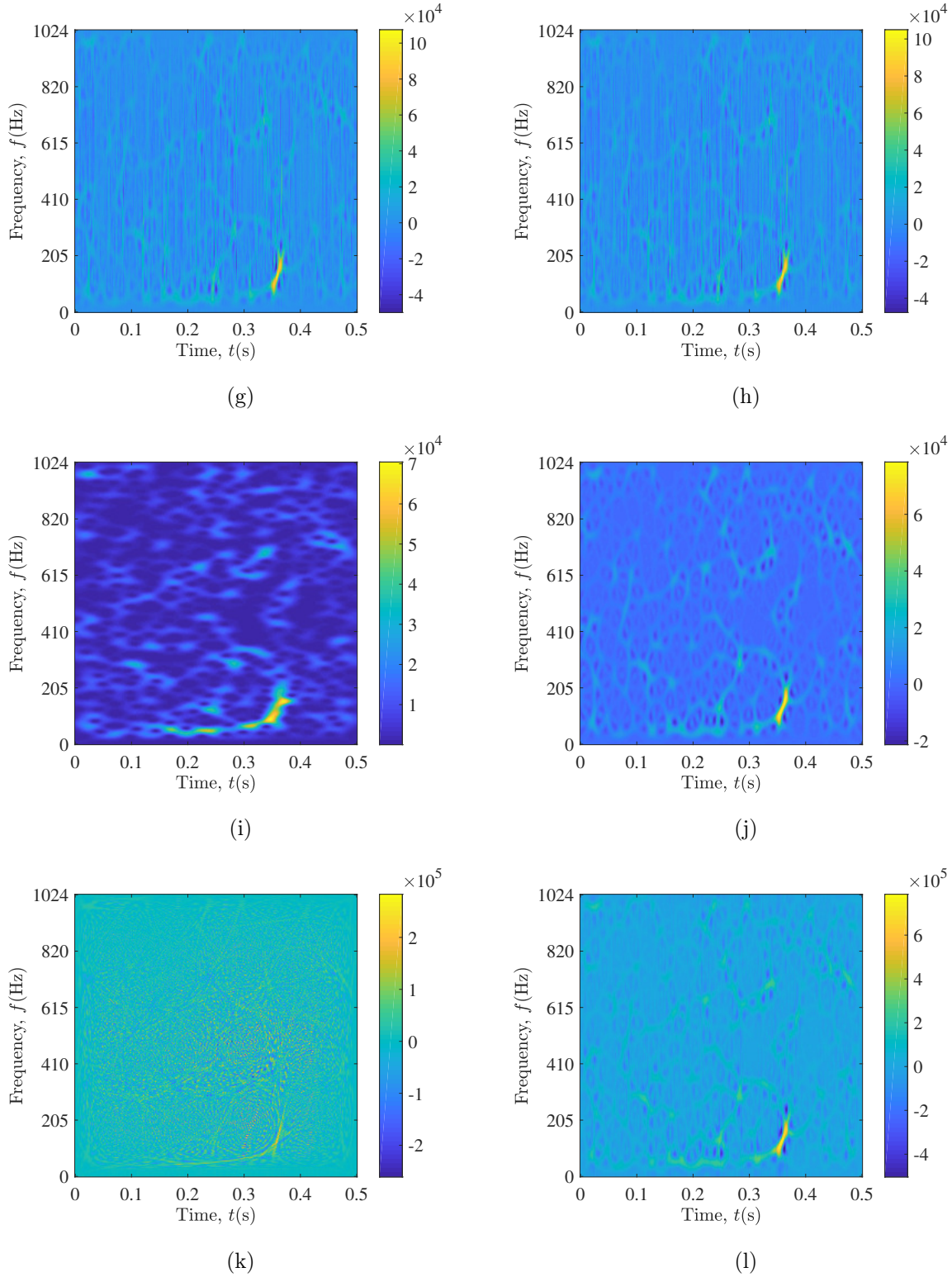


Figure 5.10 (cont.) TFDs of the time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 19 dB (OMF-SNR = 14.92 dB, SNR = −81.46 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

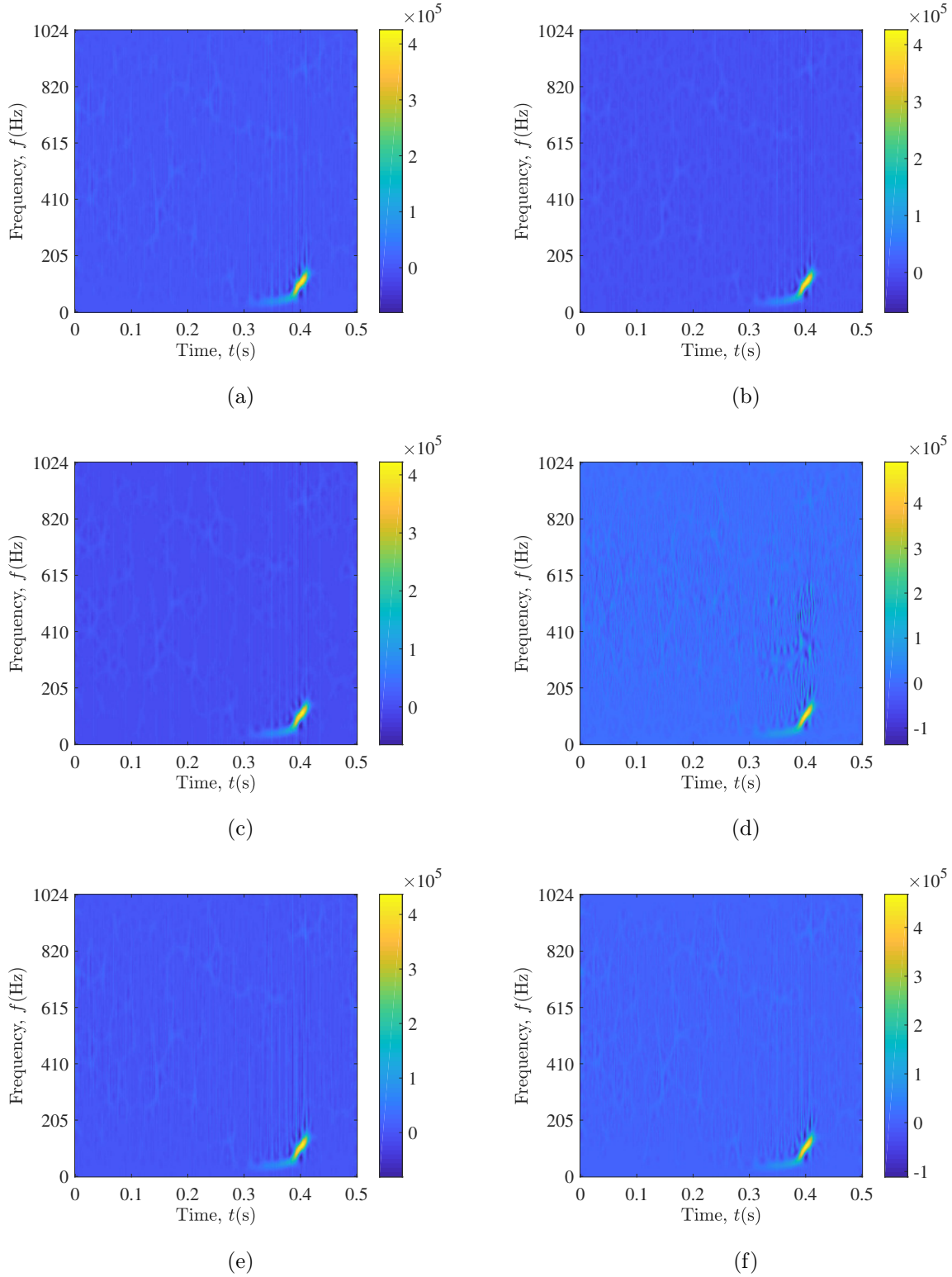


Figure 5.11 TFDs of the time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 30 dB (OMF-SNR = 25.82 dB, SNR = −60.12 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

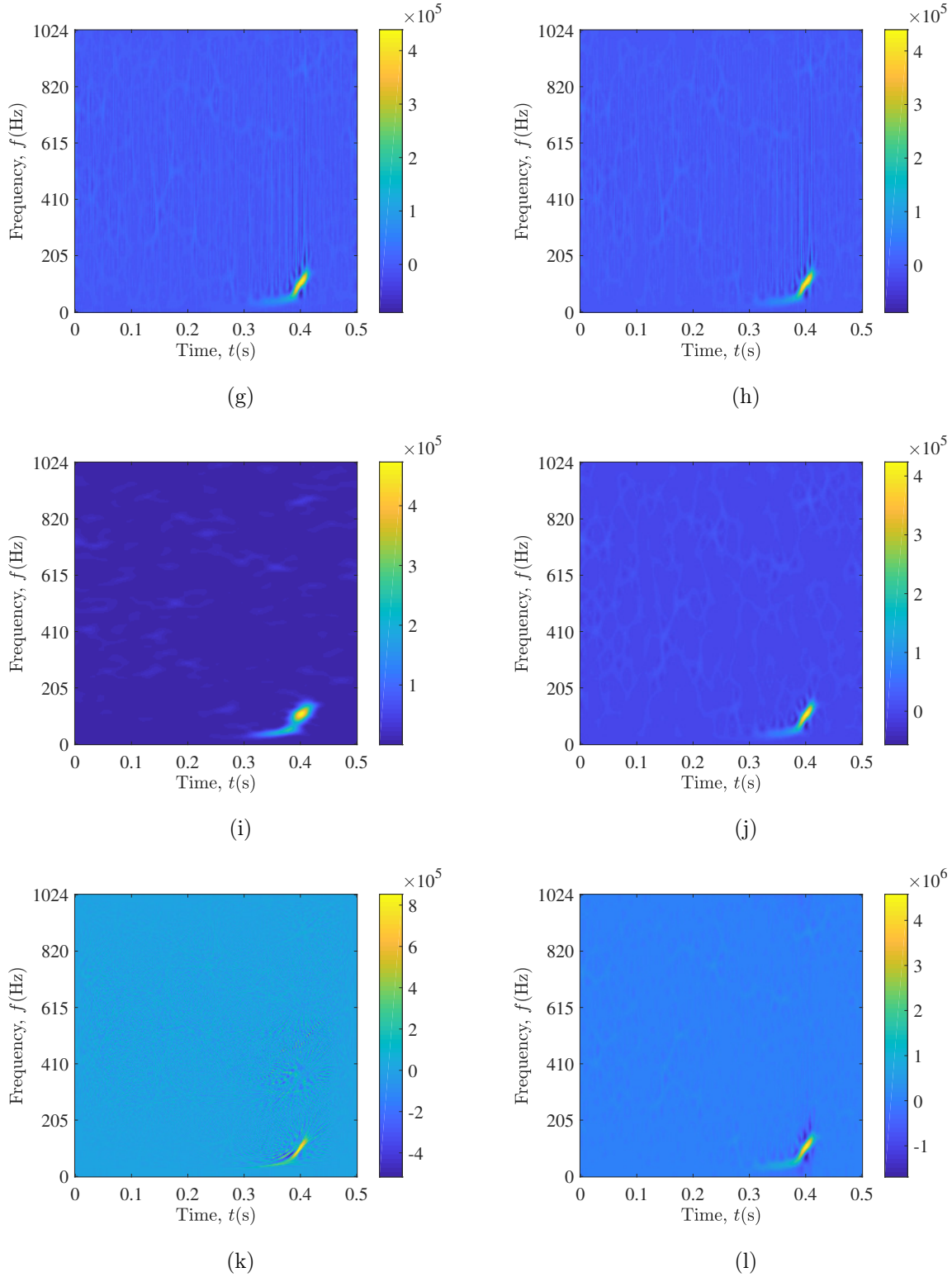


Figure 5.11 (cont.) TFDs of the time-series data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 30 dB (OMF-SNR = 25.82 dB, SNR = −60.12 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

5.4 Input Dataset

The time-series dataset contains 100 000 data examples of the 1024 samples length, where each example represents a unique realization of both the simulated GW signal and the real-life noise.

Moreover, each of the twelve obtained TFD datasets consists of 100 000 TFDs, stored as the 8-bit Portable Network Graphics (PNG) grayscale images with the 256×256 resolution. This image resolution was selected to reduce the required memory resources and adjust the image size to the size of the input layers of the utilized CNN architectures.

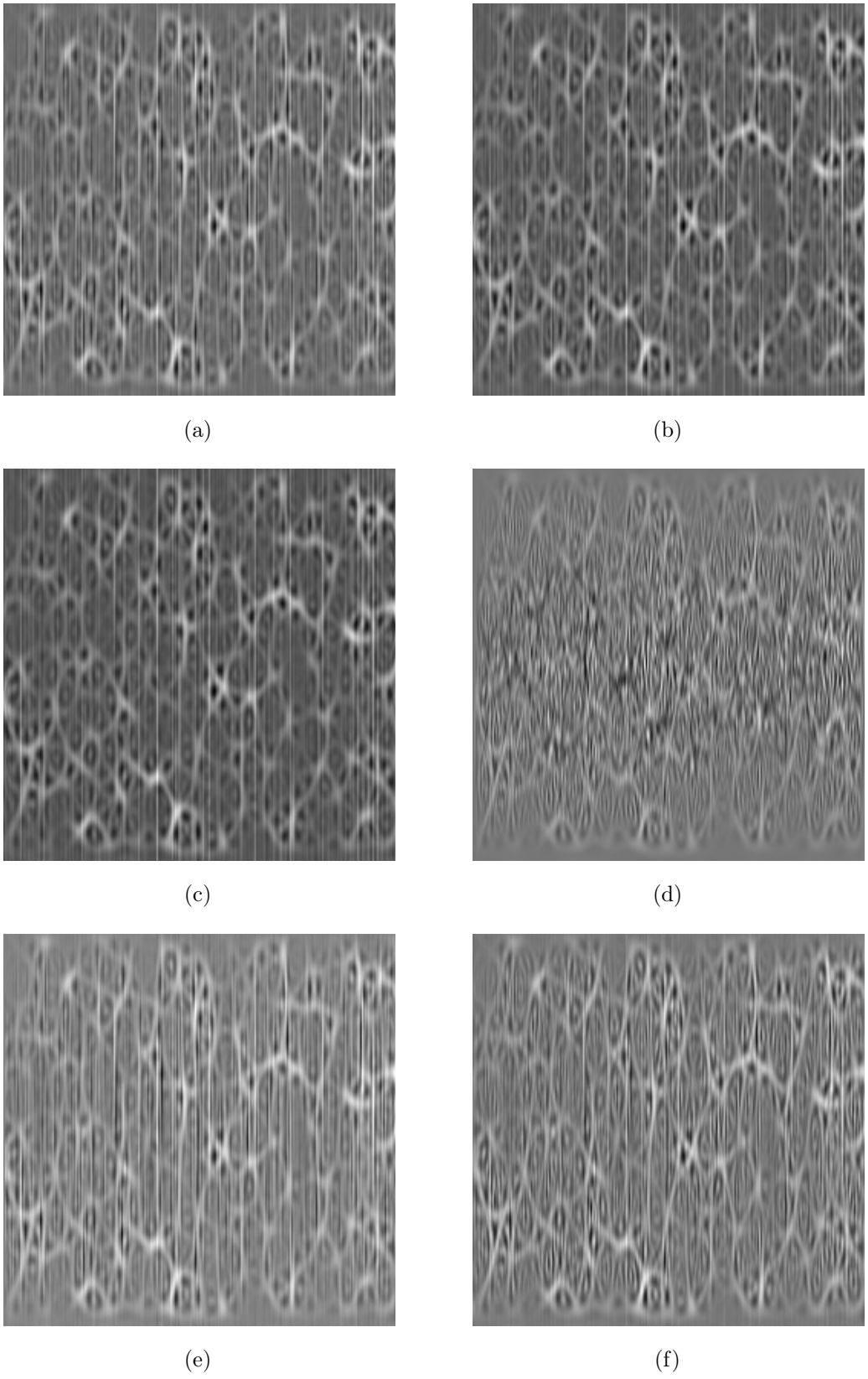
Each dataset is divided into three subsets: training, validation, and test dataset, where the training dataset includes 70%, the validation dataset 15%, and the test dataset 15% of the examples in the complete dataset. The training, validation, and test datasets keep the same ratio of the data examples with GW signals to those with only noise as in the complete dataset.

The time series and the TFD images are normalized before being used as input to the deep learning classification algorithms. Each data example is scaled independently to obtain the values in the range of $[0, 1]$:

$$I_n = \frac{I - I_{min}}{I_{max} - I_{min}}, \tag{5.7}$$

where I_n is the normalized value, I is the value being normalized, whereas I_{min} and I_{max} represent the minimum and maximum data example values, respectively.

Figures 5.12 and 5.13 provide the examples of the images used as input to the deep learning models. These images show the TFDs of the previously considered data examples, where Figure 5.12 provides the TFD images obtained from the data example containing only noise, and Figure 5.13 those obtained from the data example containing the GW signal embedded in the background noise at the NOMF-SNR of 19 dB. The original TFDs of these data examples are shown in the previous section in Figures 5.8 and 5.10.

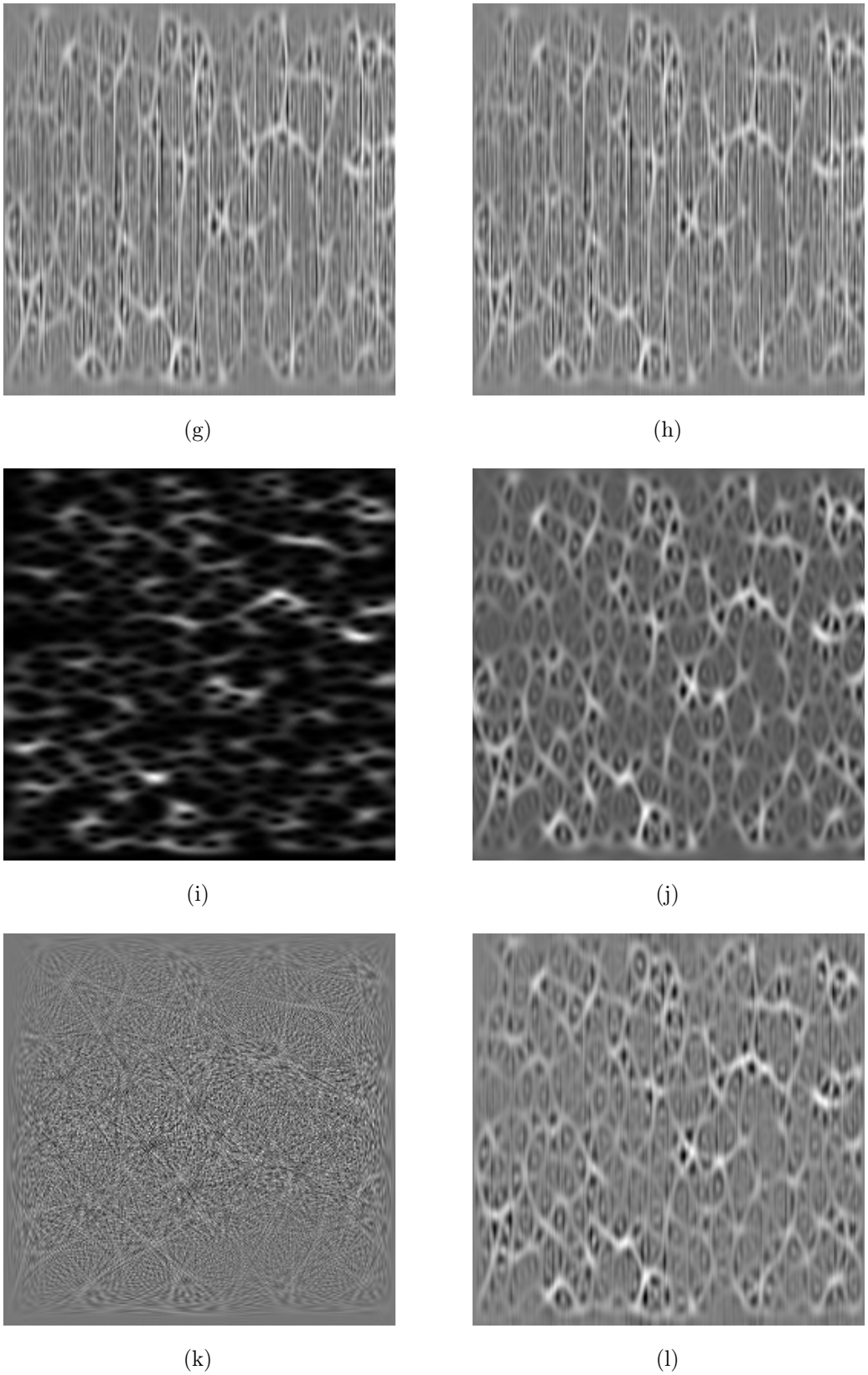


(a) (b)

(c) (d)

(e) (f)

Figure 5.12 Examples of the input images showing the TFDs of the data example containing only noise: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

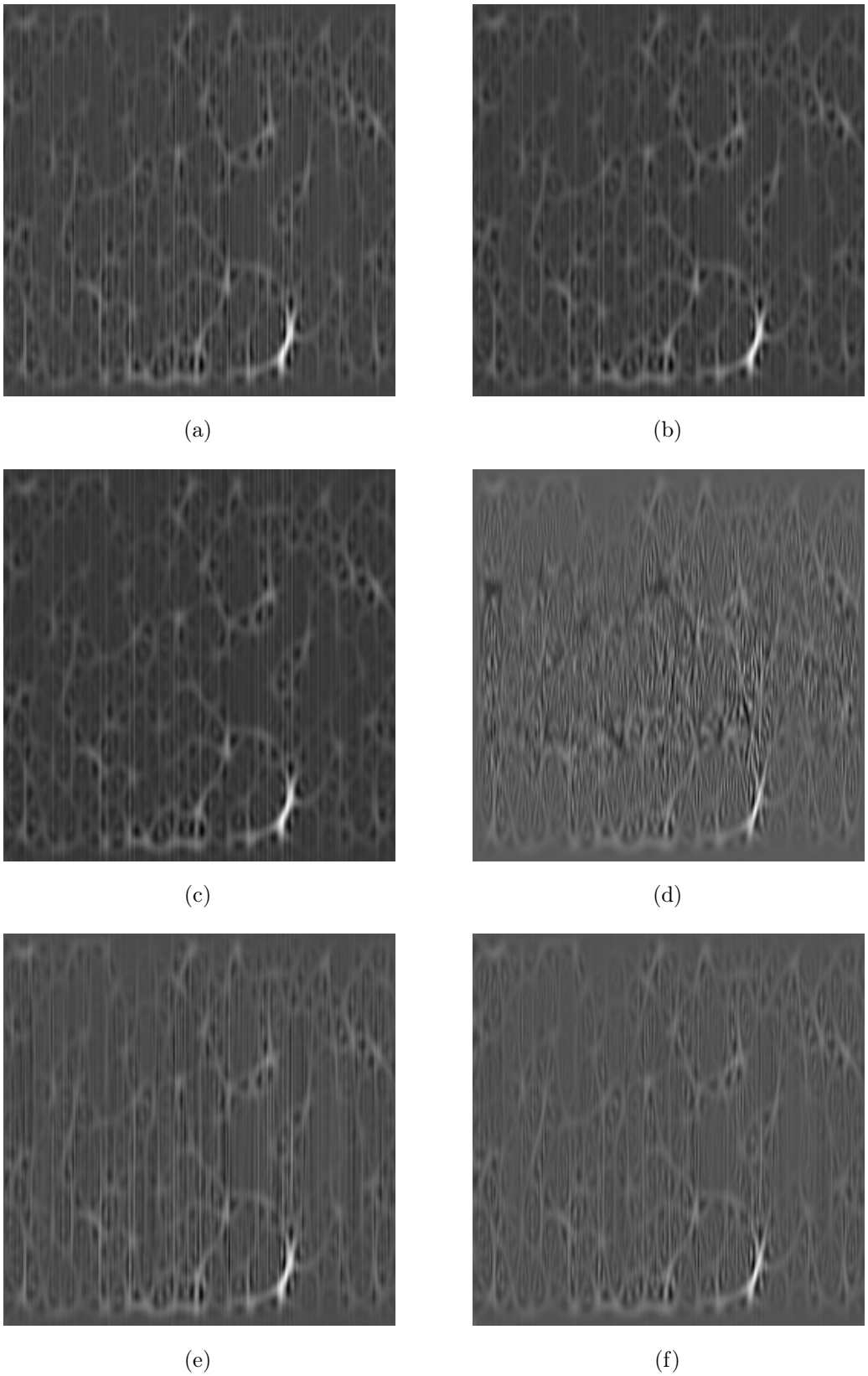


(g) (h)

(i) (j)

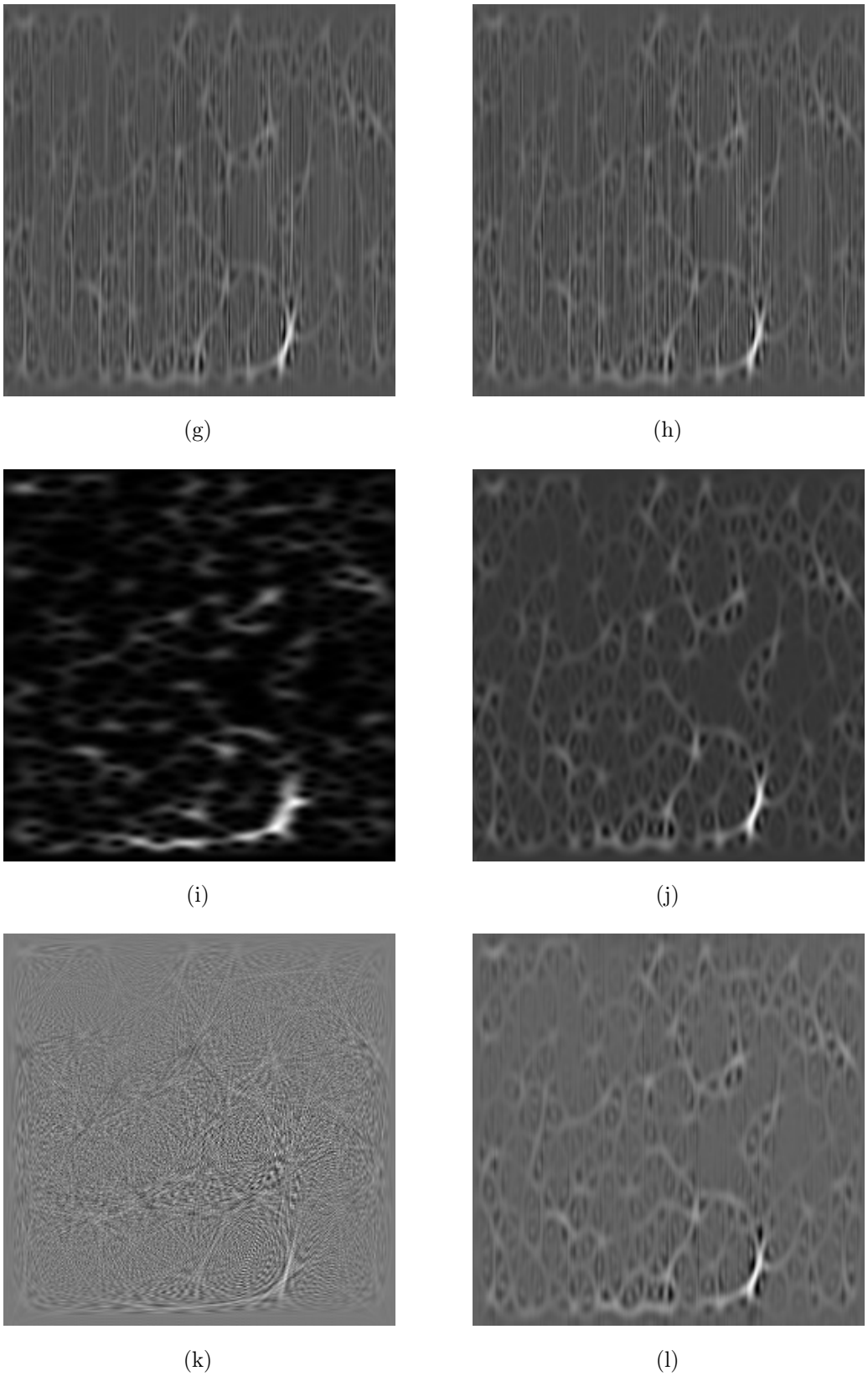
(k) (l)

Figure 5.12 (cont.) Examples of the input images showing the TFDs of the data example containing only noise: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.



(a) (b) (c) (d) (e) (f)

Figure 5.13 Examples of the input images showing the TFDs of the data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 19 dB (OMF-SNR = 14.92 dB, SNR = −81.46 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.



(g) (h)

(i) (j)

(k) (l)

Figure 5.13 (cont.) Examples of the input images showing the TFDs of the data example containing the GW signal in the noise, obtained for the desired NOMF-SNR of 19 dB (OMF-SNR = 14.92 dB, SNR = −81.46 dB): (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD.

5.5 Deep Learning Classification Models

This section presents the baseline deep learning model and the two-dimensional CNN-based deep learning models utilized within this research.

5.5.1 Baseline Model

For a baseline model, a deep learning model based on the one-dimensional CNN is used to classify the time-series data examples representing either noisy, non-stationary GW signals or pure noise. This model serves as a reference point to be used in comparison with the method proposed in this thesis. The baseline model, as an adapted version of the deep learning model proposed in [98], represents the state-of-the-art method for the deep learning-based detection of GW signals in noise, with high classification performances and the corresponding paper having the highest number of citations in the field of deep learning-based GW detection.

The model described in the original paper had the input time-series vector with a length of 8192 samples. On the other hand, in this thesis, the input time-series vector of 1024 samples is used. Therefore, the model adaptation consists of modifying network layer sizes by reducing the sizes of convolution kernels to reflect the used input vector with eight times fewer samples. The baseline model used in this thesis is depicted in Figure 5.14. As seen in Figure 5.14, the baseline model consists of four one-dimensional convolutional layers, each followed by a max-pooling layer and a ReLU activation function. Moreover, the model also contains two fully connected layers and an output layer. The fully connected layers contain 128 and 64 units, respectively, with the ReLU activation functions, while the output layer consists of one unit with a sigmoid activation function.

The model in the original paper used the convolution kernels of the sizes 16, 16, 16, and 32, respectively. The baseline model in this thesis uses the kernels whose sizes have been reduced by a factor of eight, thus obtaining the kernel sizes of 2, 2, 2, and 4, respectively. The other model parameters are kept the same as in the original model in [98]. Thus, the kernel size of 4 is used for all pooling layers. Moreover, the baseline model uses the unit stride for the convolutional layers and the stride of 4 for the pooling layers. Furthermore, the four convolutional layers have 64, 128, 256, and 512 kernels (channels), respectively. The dilations for the corresponding convolutional layers are set to 1, 2, 2, and 2, respectively. The sizes of the fully connected layers have also remained the same as they were in the original model. Additionally, the output layer has been adapted to the one neuron with a sigmoid activation function to obtain the probability of the input time-series vector representing the GW signal embedded in the noise.

The cross-entropy loss function [47] was used during the training of the baseline model. The cross-entropy loss, also called the log-loss, is suitable for binary classification problems

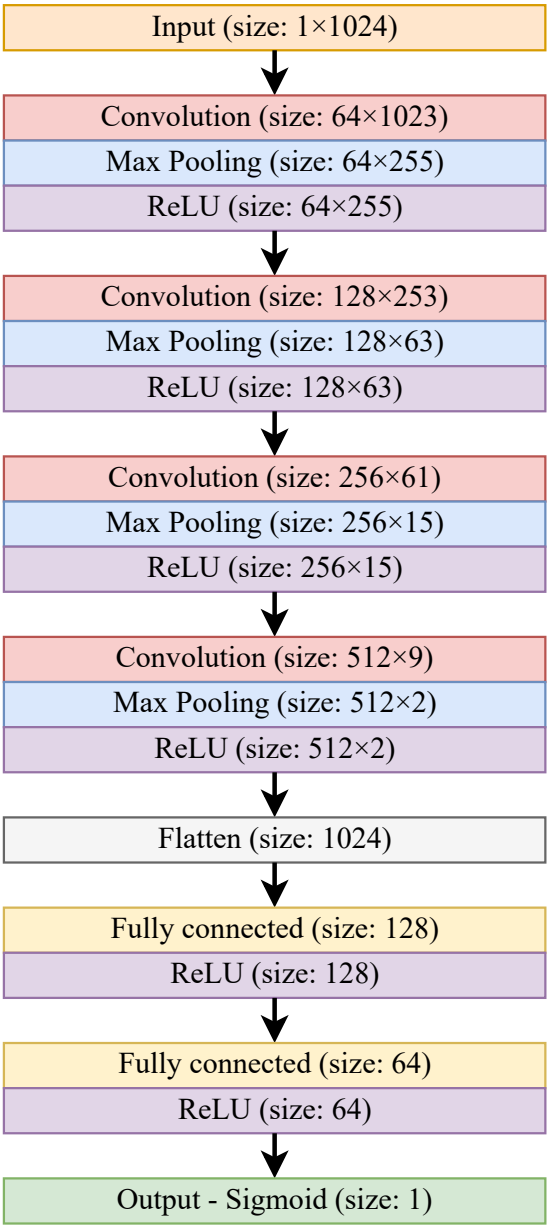


Figure 5.14 Baseline model for the deep learning classification of the time-series GW data examples.

and is based on the cross-entropy, i.e., the negative log-likelihood, between the training data and the predictions provided by the model. The binary cross-entropy cost function is for one output neuron that provides a probability value between 0 and 1 defined as [47, 197]

$$J = -\frac{1}{N_x}\sum_{i=1}^{N_x}\left(y_i \ln\left(a_i\right) + \left(1-y_i\right)\ln\left(1-a_i\right)\right), \tag{5.8}$$

where N_x is the total number of training data examples, x is the training input, y is the desired output, and a is the output predicted by the model.

Moreover, the Adam optimizer [143] was utilized for training the baseline model. Adam is an adaptive learning rate optimization algorithm whose name comes from the adaptive

moment estimation [103, 143]. Namely, Adam calculates adaptive learning rates for the individual parameters based on the estimates of the first and second moments of the gradients [143]. It combines the advantageous properties of the two other optimization algorithms [103, 143]: AdaGrad [76] and RMSProp [250]. The Adam algorithm updates the exponential moving averages of the gradient and the squared gradient that represent the estimates of the first moment (the mean) and the second raw moment (the uncentered variance) of the gradient, respectively [143]. The exponential decay rates are controlled by two hyperparameters [143]. The algorithm also introduces the correction terms to prevent the initialization bias [143].

Furthermore, the learning rate was set to $\epsilon = 1 \times 10^{-5}$, while the batch size of 32 was chosen. This learning rate value was selected as optimal for the optimizer based on the evaluation of the baseline model on the validation dataset, where the set of values $\epsilon \in \{1 \times 10^{-1}, 1 \times 10^{-2}, \ldots, 1 \times 10^{-6}\}$ was considered.

5.5.2 Two-Dimensional Convolutional Neural Network Models

Next, the deep learning models based on the three different state-of-the-art two-dimensional CNN architectures are used in this thesis to classify the TFDs of the time-series GW data examples. The schematic overview of the proposed classification approach is provided in Figure 5.15. The inputs to the classification procedure are the TFD images which contain either the non-stationary GW signal in the noisy background or the noise alone. The three different CNN architectures perform the classification, and the output of the proposed procedure is the probability that the input TFD image contains the GW signal.

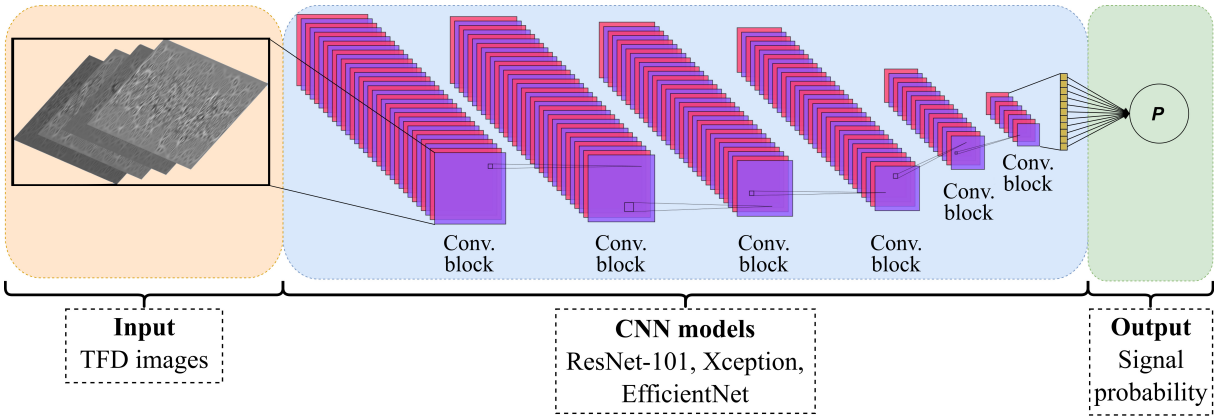


Figure 5.15 Deep learning classification of the TFDs of the time-series GW data based on the two-dimensional CNN architectures.

The CNN architectures considered in this thesis include ResNet-101 [114], Xception [58], and EfficientNet [246]. Except for the adaptations made to the last (prediction) layers, all three architectures are used in the same forms proposed in their respective papers.

Namely, each of these architectures in their original form contains 1000 softmax [169] units in the last layer for the classification of 1000 classes of images from the ImageNet dataset. In this thesis, the prediction layer of each considered CNN architecture was modified to include only one unit with a sigmoid activation function.

Moreover, the binary cross-entropy loss function was used during the training of all considered deep learning models based on the two-dimensional CNN architectures. For each model, the optimal learning rate was chosen based on the experimental evaluation on the validation dataset, where the following learning rates were considered: $\epsilon \in \{1 \times 10^{-1}, 1 \times 10^{-2}, \ldots, 1 \times 10^{-6}\}$. Furthermore, the transfer learning techniques utilizing the pretrained models were not used in this thesis, i.e., each considered deep learning model was trained from scratch.

5.5.3 ResNet-101

The first considered deep CNN architecture is ResNet-101 [114]. The ResNet class of CNN architectures allows the training of extremely deep CNN models with increased accuracy by introducing deep residual learning [114]. Residual learning helps to overcome the vanishing gradient problem commonly occurring during the training of such deep models [114]. Namely, the increased network depth, i.e., the increased number of network layers, is expected to allow learning of more complex concepts, thus leading to increased accuracy. However, it has been shown that as the network depth increases, not only does saturation occur in the increase of accuracy, but after a certain number of layers, the accuracy decreases. In addition, not only is there an increased error on the test dataset, but there is also an increase in the error on the training dataset, which means that this problem is not caused by overfitting [114, 115, 239].

Residual learning implies that stacked non-linear layers learn residual functions instead of directly learning the underlying mapping [114]. Residual learning is achieved using shortcut connections that directly propagate the layers' input to their output, where the element-wise addition is performed on each channel of two feature maps [114]. The layers' input and output must be of the same dimensions to perform the element-wise addition [114]. If considered feature maps are of different dimensions, i.e., have a different number of channels, shortcut connections performing a linear projection are used instead of identity connections [114]. In this case, linear projection is achieved by 1×1 convolutions, and shortcut connections are performed with a stride of 2 [114].

Figure 5.16 shows the examples of two types of residual blocks used in ResNet architectures. For each depicted convolutional layer in the residual block, the number of kernels and their dimensions are indicated, e.g., $3 \times 3, 128$ denotes the convolutional layer with 128 kernels of 3×3 dimensions. Figure 5.16(a) shows the standard residual block that consists of two layers with standard 3×3 convolutions and an intermediate non-linear

ReLU activation function [114]. The second ReLU activation is performed after adding the layers' input to their output using the shortcut connection [114].

On the other hand, in networks with more than 50 layers, the modified version of the residual block, whose example is shown in Figure 5.16(b), is used to reduce the computational cost [114]. This residual block contains three layers performing 1×1, 3×3, and 1×1 convolutions, respectively [114]. The first 1×1 convolutions are used to reduce the number of channels in the input, so the 3×3 convolutions are performed on the reduced number of channels, thus reducing the number of mathematical operations [114]. Finally, the second 1×1 convolutions again increase the number of channels to match the dimensions of the input and output feature maps [114]. Since the 3×3 convolutional layer has reduced dimensions, it is often called a bottleneck [114]. Therefore, this type of residual block is known as a bottleneck residual block [114].

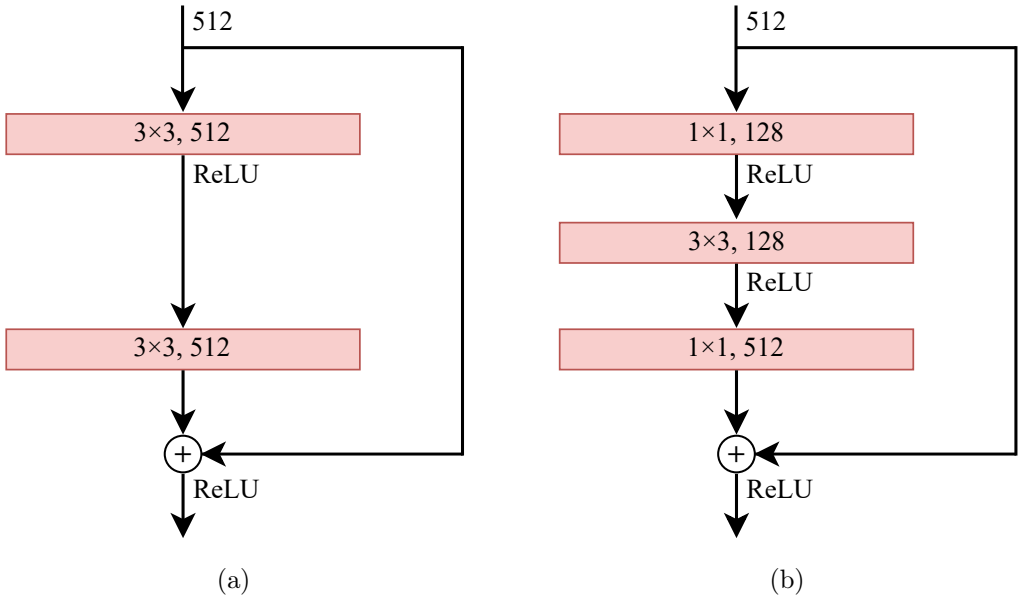


Figure 5.16 Residual blocks used in ResNet CNN architectures: (a) Standard residual block; (b) Bottleneck residual block.

The most commonly used ResNet architectures are ResNet-50, ResNet-101, and ResNet-152, where the number in the architecture name denotes the corresponding number of layers. By taking into account the available computational resources, the ResNet-101 architecture was chosen for this research. The schematic overview of the ResNet-101 architecture is shown in Figure 5.17.

The network begins with the layer performing 7×7 convolutions with 64 channels and a stride of 2, followed by the 3×3 max pooling layer with a stride of 2 [114]. Next, the four groups of residual blocks are used, where each group contains a different number of stacked bottleneck residual blocks, each performing 1×1, 3×3, and 1×1 convolutions [114]. These four groups comprise 3, 4, 23, and 3 bottleneck residual blocks, respectively [114]. The number of the output channels of the residual blocks in each group is 256, 512,

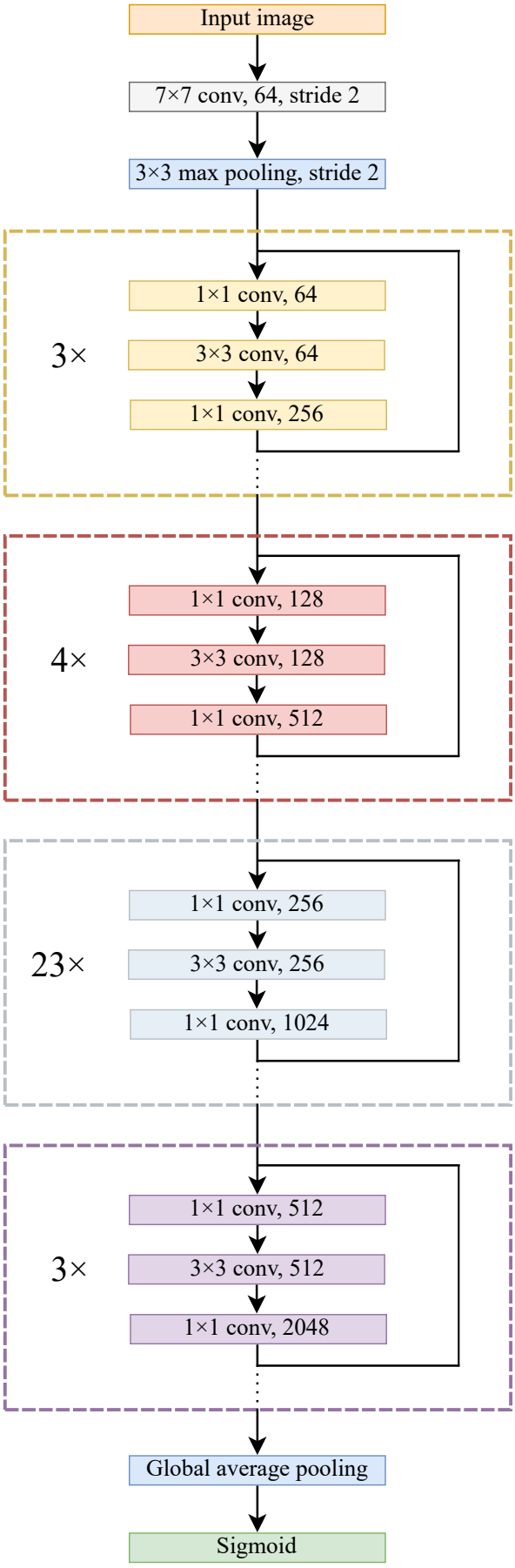


Figure 5.17 ResNet-101 CNN architecture.

1024, and 2048, respectively [114]. Moreover, the second, third, and fourth group's first 1×1 convolutional layer performs downsampling by a stride of 2 [114]. The corresponding three residual blocks utilize the projection shortcuts, while all other residual blocks use the identity shortcut connections [114]. Finally, the global average pooling is performed before the fully connected layer [114]. The research presented in this thesis replaced the fully connected layer with 1000 softmax units originally used in the ResNet-101 CNN architecture by a single sigmoid activation function.

For the purpose of this research, the utilized ResNet-101 architecture was trained using the Adam optimizer with the learning rate $\epsilon = 1 \times 10^{-6}$ and the batch size of 16.

5.5.4 Xception

The second considered deep CNN architecture is Xception [58]. The name Xception comes from the "Extreme Inception". Namely, the Xception CNN architecture is entirely based on the depthwise separable convolutions [58], which can be interpreted as an extreme version of the Inception module found in the Inception CNN architecture introduced in [243]. The depthwise separable convolutions [234] consist of the pointwise convolution, i.e., the 1×1 convolution, followed by the depthwise convolution [58]. The pointwise convolution projects the input channels onto a new channel space, i.e., it builds new features by calculating linear combinations of the input channels, and the depthwise convolution then performs a spatial convolution independently on each of these channels [58, 226]. Thus, the pointwise convolution maps the cross-channel correlations, while the depthwise convolution maps the spatial correlations of each channel provided by the pointwise convolution [58]. Therefore, the depthwise separable convolutions allow the separate mapping of the cross-channel and spatial correlations [58].

Standard convolution operation applies convolution kernel $K \in \mathbb{R}^{k \times k \times c_i \times c_j}$ to the $h_i \times w_i \times c_i$ input tensor T_i producing the $h_i \times w_i \times c_j$ output tensor T_j, where k denotes the width of the convolution kernel, c_i and c_j the number of channels of the input and output tensor, while h_i and w_i the height and width of the input tensor [226]. Thus, the computational cost of the standard convolution operation is $h_i \cdot w_i \cdot c_i \cdot c_j \cdot k \cdot k$ [226]. On the other hand, the depthwise separable convolutions have the total computational cost of $h_i \cdot w_i \cdot c_i \cdot (k^2 + c_j)$ [226]. Therefore, the depthwise separable convolutions reduce the computational cost by a factor of $k^2 c_j / (k^2 + c_j)$, which can be approximated by k^2, compared to the standard convolutions, accompanied by only a slight reduction in accuracy [124, 226].

The described concept of the depthwise separable convolutions used in the Xception CNN architecture is illustrated in Figure 5.18. This type of depthwise separable convolutions is a modified version of the commonly used depthwise separable convolutions that employ the reverse order of operations, i.e., the depthwise convolution is followed

by the pointwise convolution [58]. Moreover, in contrast to the Inception module, the depthwise separable convolutions utilized in the Xception architecture do not use an intermediate non-linear activation function (ReLU) after the pointwise convolution [58]. Using depthwise separable convolutions reduces the computational cost and model size while achieving high accuracy [58].

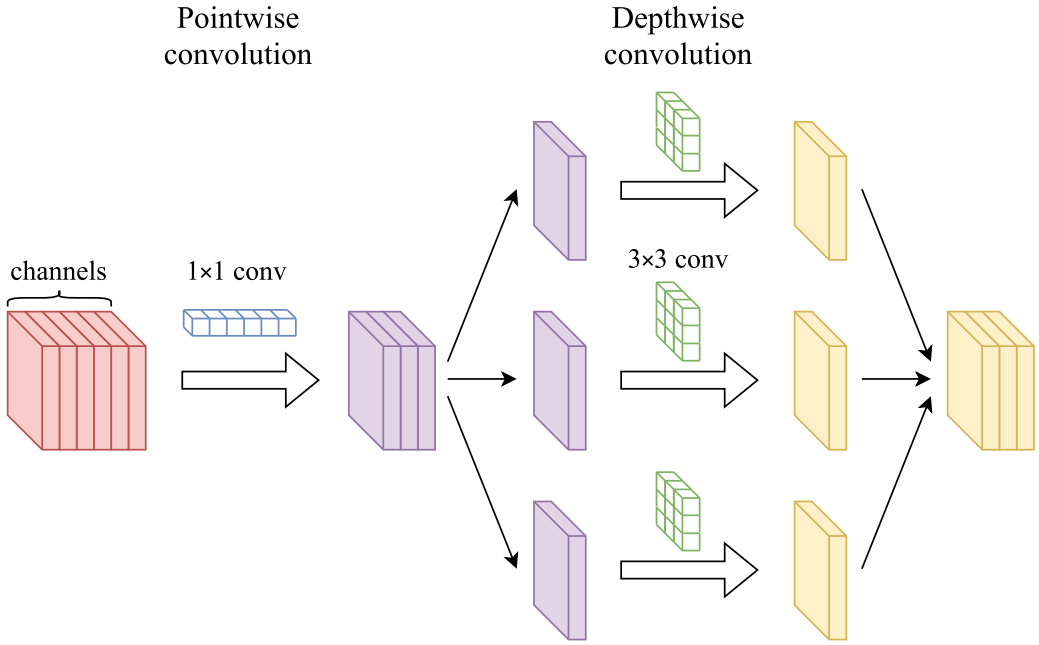


Figure 5.18 Depthwise separable convolutions.

In addition to the depthwise separable convolutions, the Xception architecture also utilizes the residual (shortcut) connections [58], previously described in the ResNet architecture. The overview of the Xception CNN architecture is shown in Figure 5.19. The Xception architecture consists of 36 convolutional layers stacked together and organized into 14 modules [58]. Each module, except the first and last, also has a linear residual connection [58]. Moreover, the final layer includes a sigmoid activation function for classification purposes. The fully connected layers may also be optionally inserted before the final layer [58].

As seen in Figure 5.19, the Xception architecture may be divided into three main parts: the entry, middle, and exit flow [58]. All convolutional layers, including the first two layers performing the standard convolutions and all other layers performing the depthwise separable convolutions, use the kernels of the 3×3 dimensions and are followed by the batch normalization [58, 131]. Moreover, all max-pooling layers operate on the 3×3 regions with a stride of 2 [58].

The entry flow consists of four modules, where the first module contains the standard convolutional layers and the other three contain the depthwise separable convolution layers using linear residual connections with a stride of 2 [58]. The first module comprises

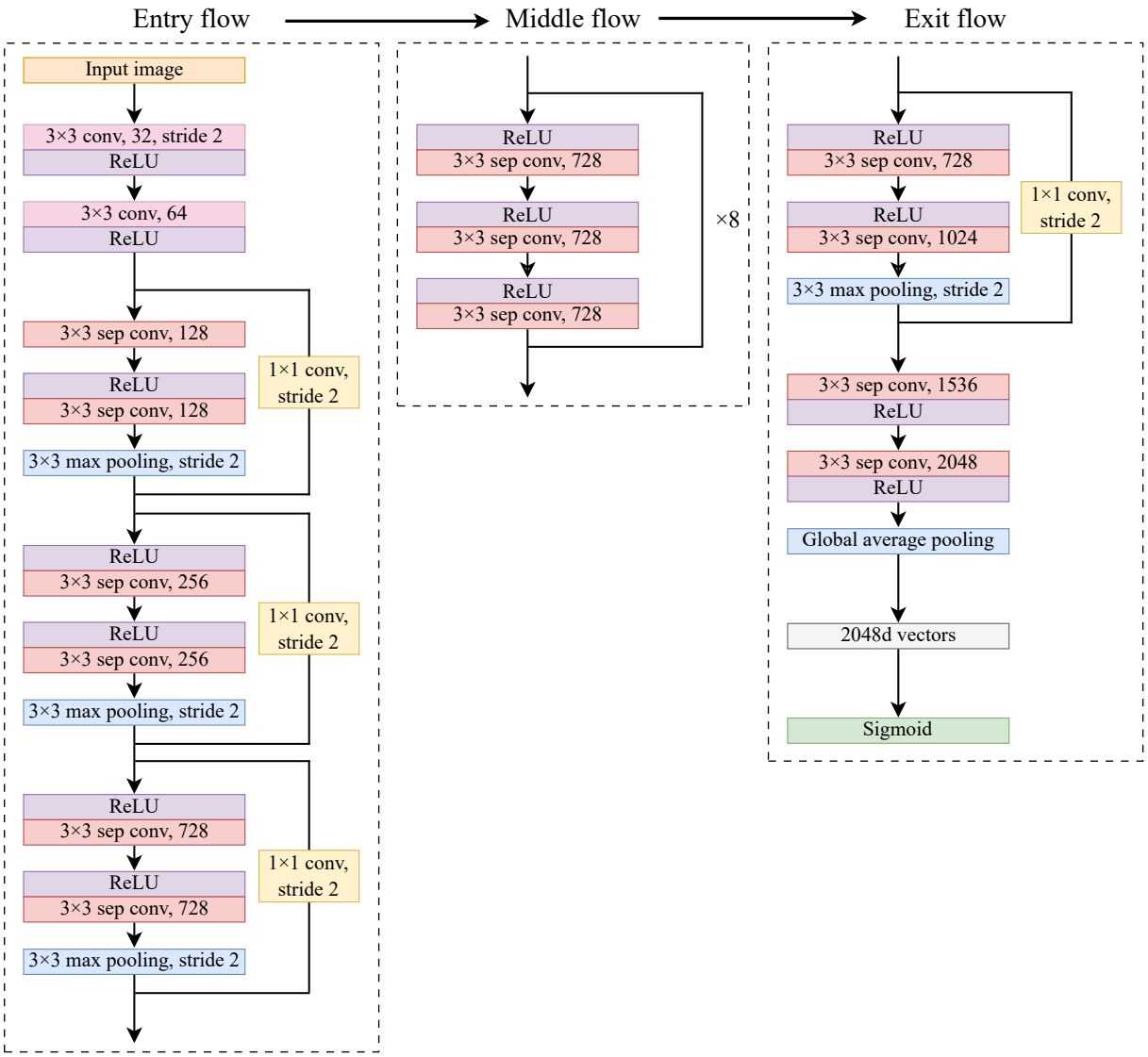


Figure 5.19 Xception CNN architecture.

two convolutional layers with 32 and 64 kernels (channels), respectively [58]. The ReLU activation function follows each convolutional layer in this module, and the first layer also performs the downsampling by a stride of 2 [58]. The second module contains two layers performing the depthwise separable convolutions with 128 kernels, where the first layer is followed by the ReLU activation and the second one by the max-pooling layer [58]. The third and the fourth module have the same structure, where the stack of two depthwise separable convolution layers, each preceded by a non-linear ReLU activation, is followed by the max-pooling layer [58]. The difference between these two modules is the number of kernels used, equal to 256 and 728 for the third and the fourth module, respectively [58].

Next, the middle flow contains eight repetitions of the same module with a residual connection and a stack of three depthwise separable convolution layers with 728 kernels, each preceded by a ReLU activation [58]. Finally, the exit flow includes two modules with the depthwise separable convolution layers, where the first module has a residual

connection, and the other does not [58]. The first module in this flow contains a stack of two depthwise separable convolution layers, each preceded by a ReLU activation, followed by the max-pooling layer [58]. These two convolutional layers use 728 and 1024 kernels, respectively [58]. The second module in the exit flow consists of two convolutional layers with 1536 and 2048 kernels, respectively, each followed by a ReLU activation function [58]. Finally, the global average pooling is applied to the output of the last module, resulting in 2048-dimensional vectors that are used as input to the last (output) layer containing a sigmoid activation function [58].

In this work, the Xception architecture was trained by applying the Adam optimizer with the learning rate $\epsilon = 1 \times 10^{-4}$ and the batch size of 32.

5.5.5 EfficientNet

The third considered deep CNN architecture is EfficientNet [246], representing the current state-of-the-art architecture for classification tasks. This class of CNNs efficiently deals with model scaling. Namely, CNNs are designed for specific computational resources [246]. If additional resources become available, CNNs are then scaled up to achieve higher accuracy [246]. There have been three main approaches in scaling up CNNs [246], i.e., three network dimensions are most commonly scaled up: network depth (number of layers) [114], network width (number of channels) [125, 282], and image resolution [127]. Deeper CNNs can learn more complex features but are more challenging to train due to the vanishing gradient problem [246, 282]. Wider CNNs can learn more detailed features and are less difficult to train [246, 282]. Very wide and shallow CNNs, on the other hand, have problems with learning higher-level features [246, 282]. CNNs using input images with higher resolution can learn more detailed features [166, 246]. Scaling up only one of the mentioned network dimensions increases accuracy, but this increase quickly saturates [246]. Moreover, manually scaling multiple dimensions is time-consuming and often results in sub-optimal performances [246, 287].

Therefore, the EfficientNet architectures introduce the compound scaling method that uniformly scales network depth, width, and resolution with a fixed ratio based on the compound coefficient ϕ [246]. Network depth, width, and resolution are scaled by coefficients $d = \alpha^\phi$, $w = \beta^\phi$, and $r = \gamma^\phi$, respectively, where $\alpha \geq 1$, $\beta \geq 1$, and $\gamma \geq 1$ are constant values selected by a small grid search on the original small model [246]. Compound coefficient ϕ defines the amount of additionally available computational resources [246]. The number of floating-point operations per second (FLOPS) of the convolution is proportional to d, w^2, and r^2 [246]. Thus, the compound scaling increases the total number of FLOPS by $(\alpha \cdot \beta^2 \cdot \gamma^2)^\phi$ [246]. For EfficientNets, the increase in FLOPS is constrained to 2^ϕ by $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ [246].

The compound scaling method can be used to scale up the existing CNN architectures

but is especially effective on the new base architecture, EfficientNet-B0, developed using a multi-objective neural architecture search by optimizing accuracy and FLOPS [245, 246]. The main building block of EfficientNet-B0 is mobile inverted bottleneck convolution (MBConv) [226, 245] with added Squeeze-and-Excitation optimization [126].

MBConv is an inverted residual module with linear bottleneck layers and depthwise separable convolutions introduced in the MobileNetV2 CNN architecture, specifically designed for mobile environments to reduce the required memory resources [226]. Namely, the input to the MBConv module is a compressed representation whose number of channels is increased by the pointwise convolution [226]. The depthwise convolution is then applied to this expanded representation, followed by another pointwise convolution that projects it back to the low-dimensional representation with the same number of channels as the input to the module [226]. In contrast to the standard residual blocks, the MBConv module uses inverted residuals where the shortcut connections are between bottleneck layers, i.e., layers with a reduced number of channels [226]. Moreover, the bottleneck layers in the MBConv module do not use non-linear activation functions [226]. The described concept of the MBConv module is depicted in Figure 5.20.

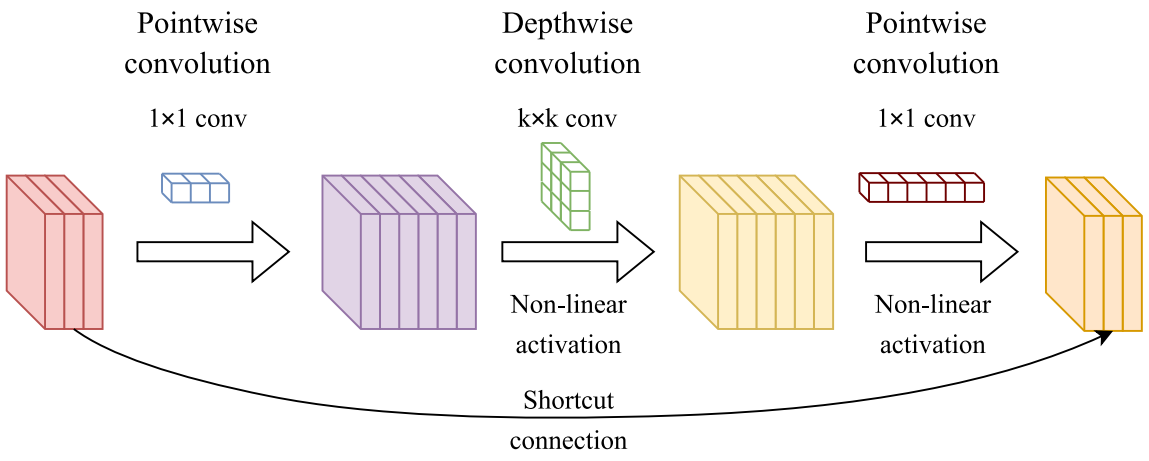


Figure 5.20 MBConv module.

The Squeeze-and-Excitation block, depicted in Figure 5.21, is introduced to model the interdependencies between the channels of the feature maps by performing the feature recalibration, i.e., the informative features are emphasized and the less informative ones are suppressed [126]. Features are recalibrated in two stages, called squeeze and excitation [126]. In the squeeze stage, global average pooling is applied to obtain statistics for each channel [126]. These channel descriptors contain aggregated global spatial information [126]. In the excitation stage, the vector with channel descriptors is used as input to two fully connected layers [126]. The first fully connected layer reduces the dimension by the reduction ratio r_E and uses the ReLU activation function, while the second fully connected

layer again increases the dimension by r_E and utilizes the sigmoid activation [126]. Finally, the obtained activations are used to rescale the input feature map by the channel-wise multiplication [126]. Thus, each channel is associated with the learned weight adapted from the channel descriptor [126].

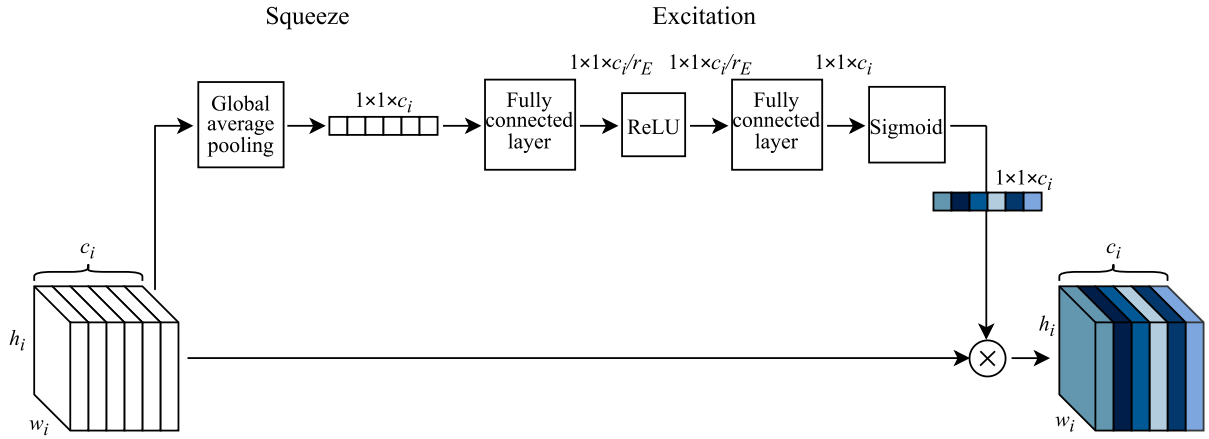


Figure 5.21 Squeeze-and-Excitation block.

The schematic overview of the EfficientNet-B0 architecture is shown in Figure 5.22. The architecture begins with the standard 3×3 convolutions with 32 output channels, followed by seven stages containing different MBConv modules [246]. Two types of MBConv modules are used: MBConv1 and MBConv6, where the number in the name denotes the expansion factor by which the input to the module is expanded in the intermediate expansion layer [246]. The second stage contains one MBConv1 module with 3×3 convolution kernels and 16 output channels, while the third stage uses two MBConv6 modules with 3×3 kernels and 24 channels [246]. Two MBConv6 modules with 5×5 kernels and 40 channels are used in the fourth stage, whereas the fifth stage consists of three MBConv6 modules with 3×3 kernels and 80 channels [246]. The sixth stage contains three MBConv6 modules with 5×5 kernels and 112 channels, the seventh stage four MBConv6 modules with 5×5 kernels and 192 channels, and the eighth stage one MBConv6 module with 3×3 kernels and 320 channels [246]. The standard 1×1 convolution block with 1280 channels follows the MBConv stages [246]. Finally, the global average pooling is performed, and the network ends with the sigmoid activation in the last layer [246].

The compound scaling is applied to the base EfficientNet-B0 CNN in two stages [246]. First, the compound coefficient is set to $\phi = 1$, and the coefficient values $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$ are selected as optimal by a small grid search taking into account previously described coefficient definitions and constraints [246]. Next, the obtained coefficients are used as constant values, and the EfficientNet-B0 CNN is scaled up by different compound coefficient values, thus obtaining CNN architectures EfficientNet-B1 to B7 [246]. EfficientNet CNNs achieve state-of-the-art classification accuracies while typically having an order of magnitude fewer FLOPS and parameters and running considerably

faster than the competitive architectures [246]. For the research presented in this thesis, the EfficientNet-B2 CNN architecture was used. The EfficientNet-B2 architecture was obtained by the appropriate compound scaling of the base EfficientNet-B0 architecture shown in Figure 5.22.

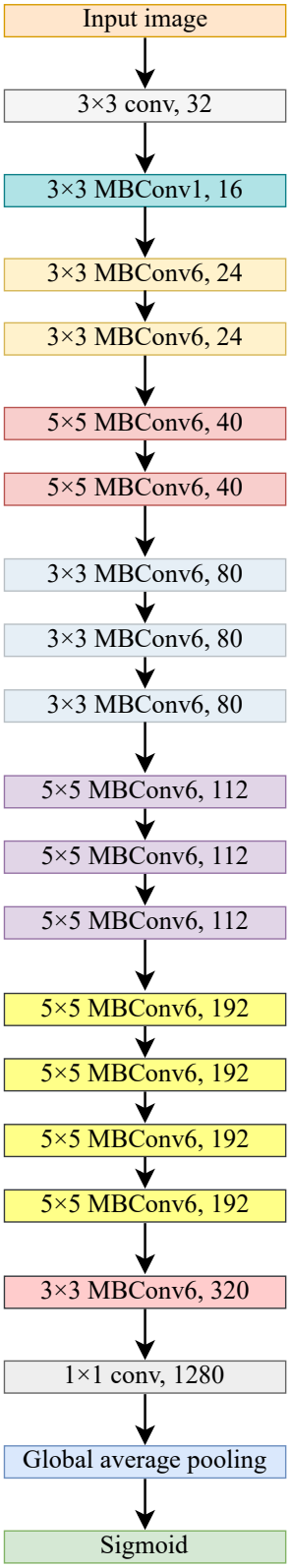


Figure 5.22 EfficientNet-B0 CNN architecture.

In this study, the deep learning models based on the EfficientNet-B2 CNN architecture were trained using the RMSProp optimizer [250] with the learning rate $\epsilon = 1 \times 10^{-4}$ and the batch size of 32. RMSProp is an optimization algorithm used for training deep learning models that adapts the learning rate of each model parameter by dividing it with the square root of the exponentially weighted moving average of the past squared gradient values [103, 250]. The exponential decay is controlled by a single hyperparameter [103, 250].

5.5.6 Training Parameters

The above-mentioned main parameters used for training the baseline CNN model and the deep learning models based on the three two-dimensional CNN architectures are summarized in Table 5.1. The provided parameters include loss function, optimizer algorithm, learning rate, and batch size.

Table 5.1 Parameters used for training the deep learning models.

CNN architecture	Parameters			
	Loss function	**Optimizer**	**Learning rate**	**Batch size**
Baseline 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

Several precautions were taken during the training of the described deep learning models to avoid potential underfitting or overfitting. The use of extensive and diverse datasets and high-capacity deep learning models prevented underfitting. Moreover, overfitting was avoided during the training by using the elbow method. Namely, the model with the lowest validation loss was chosen while simultaneously ensuring that the accuracy obtained on the validation dataset is similar to the one obtained on the training dataset.

The training and testing of the deep learning models included in the described experiments were conducted using TensorFlow 2.4 [2].

5.6 Evaluation Metrics

The classification performance of the considered baseline and TFD-CNN deep learning models was evaluated using several evaluation metrics on the test dataset, including 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) [46, 103]. Moreover, additional detailed insight into the obtained classification results

is provided using confusion matrices, ROC curves, and precision-recall curves for each considered classification model. The utilized evaluation metrics are defined and described below.

The classification accuracy is an evaluation metric that can be defined as the ratio between the number of correct predictions and the total number of predictions that the classification model has made. Namely, each data example from the test dataset is fed to the classification model as an input, and the model makes a prediction about that data example by assigning it with the appropriate label. The predicted labels are then compared to the known actual labels for all examples in the dataset, and the classification accuracy is calculated.

In this thesis, the classification accuracy was calculated for each tested deep learning classification model with its probability threshold set to the value that provides the optimal model performance. This probability threshold value was selected based on the evaluation of the deep learning model performed on the validation dataset.

Classification accuracy is the most commonly used metric to evaluate the performance of deep learning classification models. However, the classification accuracy is insufficient and may lead to incorrect conclusions about the performance of classification models in imbalanced classification problems, i.e., those problems in which there are not the same number of data examples belonging to each class under consideration [41].

The confusion matrix is an evaluation metric that enables the visualization of the classification model's performance by summarizing the predictions made by a classification model into a specific table layout [30]. Each row in the confusion matrix represents the examples in an actual class, whereas each matrix column represents the examples in a predicted class. The number in each confusion matrix cell represents the number of instances satisfying the predicted and the actual class combination, which is characteristic for that particular cell. Thus, the cells on the main diagonal of the confusion matrix, where the predicted and the actual class align, represent the predictions correctly made by the classification model. On the other hand, other matrix cells represent the incorrect predictions, where the classification model has confused two classes.

For a binary classification problem, the confusion matrix consists of two rows and two columns. One class may be denoted as positive (P), and the other class as negative (N). Therefore, regarding the classification model predictions, we distinguish between the following instances:

- True positives (TP) - examples correctly predicted belonging to the positive class
- True negatives (TN) - examples correctly predicted belonging to the negative class
- False positives (FP) - examples incorrectly predicted belonging to the positive class

- False negatives (FN) - examples incorrectly predicted belonging to the negative class

The confusion matrix as an evaluation metric provides a more detailed insight into the performance of the classification models. It is not focused only on the classification accuracy but also on the prediction performance for individual classes and different types of errors and their frequencies. Therefore, the confusion matrix also represents the basis for the other metrics used to evaluate classification models.

For the classification problem that is presented in this thesis and involves two classes, the data examples containing the GW signals in noise can be considered a positive class, while those examples with no GW signals (noise only examples) can be denoted as a negative class. Therefore, using the previously defined confusion matrix terms, the classification accuracy can be defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{5.9}$$

Moreover, the true positive rate (TPR), also called sensitivity, represents the ratio between the number of data examples correctly predicted as belonging to the positive class and the total number of data examples actually belonging to the positive class. The TPR can be computed as

$$TPR = \frac{TP}{TP + FN}. \tag{5.10}$$

On the other hand, the false positive rate (FPR) represents the ratio between the number of data examples incorrectly predicted as belonging to the positive class and the total number of data examples actually belonging to the negative class. The FPR is calculated as

$$FPR = \frac{FP}{FP + TN}. \tag{5.11}$$

Both the TPR and the FPR can take values from the range between 0 and 1. The TPR and the FPR values are obtained by evaluating the performance of the deep learning classification model at the different threshold values used to map the probability outputs of the model to the predicted class labels. The obtained TPR and FPR values plotted against each other represent another evaluation metric, called the ROC curve. The ROC curve of the evaluated classification model is often shown on the same plot as the curve representing the classification model with no skill. The no-skill model is represented by a diagonal line from the bottom left to the top right of the plot, with all points below this line denoting the classification performance worse than that of the model with no skill. The perfect classification model would have a ROC curve in the form of a point in the top left of the plot.

Additionally, the area under the ROC curve, called the ROC AUC, represents another evaluation metric that aggregates the obtained ROC curves into a single numerical result,

thus facilitating the comparison of various classification models. The ROC AUC is one of the most commonly used metrics to evaluate the performance of the classification models in binary classification problems. The ROC AUC can take values from the range $[0, 1]$ and can be used to analyze the classification model's efficiency at discriminating classes. The higher ROC AUC values are typical of classification models that perform well across a wide range of probability threshold values and provide good class separation. The ideal ROC AUC value of 1.0 would be obtained for the perfect classification model, while the no-skill classification model will have the ROC AUC value of 0.5.

Another metric used in this thesis to evaluate the performance of the classification models is recall. The recall is calculated in the same way as the TPR defined by (5.10) and indicates how accurately the positive class was predicted.

On the other hand, the precision, also called the positive predictive value (PPV), is an evaluation metric that measures the ratio of the number of the predicted examples that actually are part of the positive class to the number of examples assigned to the positive class by the classification model predictions. Thus, the precision is calculated as

$$Precision = \frac{TP}{TP + FP}. \tag{5.12}$$

Moreover, it is often desirable in classification problems to simultaneously balance both the recall and precision value to obtain both the satisfactory accuracy and the robustness of the model. This balance can be achieved by using the evaluation metric called the F1 score that combines the precision and the recall into a single metric value. The F1 score, having values in the range $[0, 1]$, is calculated as a harmonic mean of the precision and recall values:

$$F1\ score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \tag{5.13}$$

Alternatively, the F1 Score can also be calculated as

$$F1\ score = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{5.14}$$

Besides the ROC curves, the precision-recall curves are also often used to evaluate the performance of classification models, showing the precision and recall values plotted against each other. These values are obtained by evaluating the model performance at different probability threshold values. The precision-recall curve is often provided at the same plot as the curve of a no-skill classification model represented by a horizontal line with a precision proportional to the number of data examples belonging to the positive class (0.5 in the case of a balanced classification problem). On the other hand, the perfect classification model would be represented by a point in the top right of the plot.

Finally, the evaluation metric PR AUC can be calculated as the area under the precision-recall curve. High PR AUC values indicate high precision and high recall, and,

in the case of the perfect classification performance, the PR AUC value of 1.0 would be obtained.

The analysis of the obtained classification results using the previously described evaluation metrics is provided in Chapter 6.

5.7 Statistical Significance Test

The results provided by the considered deep learning classification models require an appropriate interpretation. When comparing multiple sets of results, such as comparing the results obtained by one classification model with those obtained by another model, statistical significance tests, also called statistical hypothesis tests, are used. The statistical significance tests apply statistical methods to calculate specific quantities whose values are used to confirm or reject the default assumption on the results, called the null hypothesis [45].

The quantities provided by the statistical significance tests are called p-values [45] and are used to assess whether the obtained results are statistically significant. The assessment is performed by comparing the p-value corresponding to a particular set of results to a predetermined threshold value called the significance level α [45]. The significance level is most commonly set at the value $\alpha = 0.05 = 5\%$ or $\alpha = 0.01 = 1\%$. The considered result is statistically significant for $p < \alpha$, i.e., the null hypothesis is rejected [45]. On the other hand, for $p > \alpha$, the null hypothesis cannot be rejected, which means that the result is not statistically significant [45].

Since the interpretation of the statistical significance test results is based on probability values, the outcome of the test is prone to errors. There are two types of possible errors: type I error (false positive) that represents incorrectly rejecting a true null hypothesis, and type II error (false negative) which denotes incorrectly failing to reject a false null hypothesis [45].

In this thesis, McNemar's statistical significance test was performed on the obtained classification results to check whether the differences between the results of the different deep learning classification models were statistically significant. McNemar's statistical test [84, 183] was selected as the most appropriate test type for this purpose as it has been shown to be the only statistical test with an acceptable type I error, i.e., the probability of falsely detecting a difference even though there is no difference, for algorithms that can only be performed once [72]. Since deep learning classification models are characterized by computationally intensive training procedures and large datasets, multiple repetitions of their training and evaluation are very time-consuming and impractical, leading to McNemar's test as the most suitable choice for the statistical significance analysis, allowing the comparison of the trained deep learning models on a single test dataset.

McNemar's test applied to examine the statistical significance of the differences in the results of two considered classification models uses a 2×2 contingency table [84]. Namely, both compared models were trained on the same training dataset and evaluated on the same test dataset examples. Then, the predictions made by these models are compared to the known actual values and evaluated as correct or incorrect. Finally, these observations are summarized in a table form, thus obtaining a contingency table [44, 84].

Table 5.2 shows the structure of the 2×2 contingency table used for McNemar's test of the statistical significance of the differences in the classification results of two deep learning models. The variable A denotes the number of instances in the test dataset that both Model 1 and Model 2 classified correctly, B is the number of instances that Model 1 got correct and Model 2 incorrect, C is the number of instances that Model 1 got wrong and Model 2 right, while D represents the number of instances that both Model 1 and Model 2 classified incorrectly.

Table 5.2 Structure of a contingency table used for McNemar's test of the statistical significance of the differences in the results provided by two deep learning classification models.

	Model 2 Correct	**Model 2 Incorrect**
Model 1 Correct	A	B
Model 1 Incorrect	C	D

McNemar's test is a paired non-parametric statistical significance test that examines the marginal homogeneity of the contingency table, thus checking whether two binary classification models disagree in the same manner [44]. McNemar's test statistic is calculated as [84, 183]

$$\chi^2 = \frac{(B-C)^2}{B+C}. \tag{5.15}$$

Furthermore, the expression in (5.15) is corrected for continuity as follows [78]:

$$\chi^2 = \frac{(|B-C|-1)^2}{B+C}. \tag{5.16}$$

Thus defined test statistic has a χ^2 distribution with one degree of freedom, assuming that each contingency table cell used for the calculation contains at least 25 instances [44].

McNemar's test statistic is not intended to provide information on the accuracy and error rates of the considered two classification models [44]. Instead, it gives information on the difference in the relative proportion of errors obtained by the models [44]. The null hypothesis of McNemar's test is that the predictions of the two considered models differ

to the same extent [44]. If the contingency table cells used to calculate the test statistic do not contain a similar number of instances, the models make different errors (errors on different examples of the test dataset) and also have a different relative proportion of errors on the test dataset [44]. Therefore, the null hypothesis can be rejected, and the test results are considered statistically significant [44]. If the opposite is true, the null hypothesis cannot be rejected [44].

Based on the above, in this thesis, the classification results provided by the baseline deep learning model were paired with the results provided by each TFD-CNN deep learning model in order to calculate McNemar's test statistics. The statistical significance level of $\alpha = 0.01 = 1\%$ was chosen as a commonly used value in the statistical significance analysis. In the experimental setup proposed in this thesis, each TFD of the input data examples is combined with three different deep learning architectures. Therefore, the Bonferroni correction [38, 62] was applied, dividing the selected significance level α by a factor of three and obtaining a new significance level of $\alpha = 0.00333 = 0.333\%$.

Namely, the Bonferroni correction is used in the case of multiple comparisons, with several statistical tests being conducted simultaneously [273]. The correction consists in reducing the significance level α to take into account the number of statistical tests that have been conducted. In this way, the incorrect rejection of a true null hypothesis can be avoided, i.e., the risk of a type I error can be decreased [21, 273]. The significance level for each statistical test is selected by dividing the original significance level α by the number of the conducted statistical tests [21].

The results obtained using McNemar's statistical test and the selected significance level are presented and discussed in Chapter 6. These statistical results indicate whether the considered TFD of the input time-series data examples enhances the information provided by the original time-series form when used for deep learning classification. The statistically significant results can be interpreted as a statistically significant improvement in the performance of the considered TFD-CNN deep learning models compared to the performance of the baseline model when used for the detection of non-stationary GW signals in noise.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter presents the results obtained by applying the method for detecting the non-stationary GW signals in noise, which is proposed in the previous chapter of this thesis. The presentation of the obtained results is accompanied by a detailed quantitative analysis and an appropriate discussion.

6.1 Model Testing

Each of the considered deep CNN architectures (ResNet-101, Xception, and EfficientNet) was trained on each of the 12 TFD datasets, where each training dataset contained 75 000 input data examples. Model validation was performed on 15 000 data examples, while model performance was evaluated on the test dataset of 15 000 data examples using the evaluation metrics described in the previous chapter.

The probability threshold values used by the baseline CNN model and each of the TFD-CNN models are listed in Table 6.1. These values were selected during the validation of the deep learning models by varying the thresholds in increments of 0.1, evaluating the model performance on the validation dataset for each threshold, and finally selecting the thresholds that yield the optimal model performance.

6.2 Accuracy

The classification accuracy values obtained by the baseline CNN model and each combination of the deep CNN architecture and the TFD of the input data examples are shown in Table 6.2. As seen in Table 6.2, a classification accuracy of 93.147% is achieved with the baseline model. On the other hand, all TFD-CNN combinations achieve very high classification accuracy values, showing similar classification performances. The lowest classification accuracy of the ResNet-101 architecture (96.540%) is obtained when used

Table 6.1 Probability threshold values used for evaluating the deep learning models on the test dataset.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	0.2	0.5	0.3
BUD	0.5	0.5	0.4
CWD	0.7	0.8	0.7
PWVD	0.7	0.4	0.8
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
SP	0.5	0.8	0.6
SPWVD	0.4	0.6	0.8
WVD	0.6	0.5	0.4
ZAMD	0.8	0.2	0.4
Baseline model		0.5	

with the WVD of the non-stationary GW data examples, while the highest accuracy (96.953%) is achieved when used with the CWD. The Xception CNN architecture provides the lowest accuracy value (96.773%) for the RIDB and the highest value (97.040%) for the WVD of the input data. The EfficientNet architecture provides classification accuracy values in the range between 96.567% and 97.100%, where these values are obtained using the ZAMD and the SP data representation, respectively.

The overall classification accuracy of the tested TFD-CNN models ranges from 96.540% (obtained by the WVD - ResNet-101 combination) to 97.100% (achieved by the SP data representation and the EfficientNet architecture). Therefore, these models' classification accuracy surpasses the baseline model's classification accuracy by 3.393% to 3.953%.

6.3 ROC AUC

Table 6.3 shows the ROC AUC values calculated for each evaluated deep learning model. As seen in Table 6.3, the ROC AUC of the baseline model is 0.96787. As for the TFD-CNN models, all combinations achieve very high ROC AUC values, thus confirming excellent classification performances. The ROC AUC values of the ResNet-101 architecture range from 0.98539 (obtained with the WVD) to 0.98810 (achieved with the RIDB). The Xception architecture achieves the lowest ROC AUC value (0.98618) when coupled with the RIDT of the input data and the highest value (0.98810) when combined with the SP data representation. EfficientNet provides the lowest ROC AUC value (0.98505) with the RIDT and the highest (0.98854) with the CWD.

Table 6.2 Classification accuracy values for evaluating the deep learning models on the test dataset. The highest accuracy values for each considered CNN architecture are marked in bold.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	0.96827	0.96800	0.96987
BUD	0.96833	0.96827	0.96893
CWD	**0.96953**	0.96840	0.96980
PWVD	0.96887	0.96967	0.96927
RIDB	0.96853	0.96773	0.96833
RIDBN	0.96927	0.96907	0.96800
RIDH	0.96787	0.96867	0.97000
RIDT	0.96800	0.96860	0.96853
SP	0.96913	0.96953	**0.97100**
SPWVD	0.96760	0.96987	0.96907
WVD	0.96540	**0.97040**	0.96820
ZAMD	0.96813	0.96820	0.96567
Baseline model		0.93147	

Overall, the evaluated TFD-CNN models show the ROC AUC values between 0.98505 (the RIDT - EfficientNet model) and 0.98854 (the CWD - EfficientNet model), thus outperforming the baseline model by 1.718% to 2.067% in terms of ROC AUC.

Table 6.3 ROC AUC values for evaluating the deep learning models on the test dataset. The highest ROC AUC values for each considered CNN architecture are marked in bold.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	0.98800	0.98708	0.98816
BUD	0.98801	0.98666	0.98686
CWD	0.98703	0.98732	**0.98854**
PWVD	0.98646	0.98729	0.98693
RIDB	**0.98810**	0.98734	0.98637
RIDBN	0.98798	0.98782	0.98754
RIDH	0.98625	0.98753	0.98805
RIDT	0.98711	0.98618	0.98505
SP	0.98727	**0.98810**	0.98823
SPWVD	0.98676	0.98802	0.98766
WVD	0.98539	0.98709	0.98569
ZAMD	0.98708	0.98761	0.98752
Baseline model		0.96787	

6.4 Recall

The recall values for the baseline model and each TFD-CNN model are given in Table 6.4. As seen in Table 6.4, the baseline CNN model has a recall of 88.853%, while all TFD-CNN combinations show high recall values and outperform the baseline model. The lowest (94.240%) and highest (95.533%) recall values for ResNet-101 are obtained with the RIDH and the BJD of the input data, respectively. The Xception CNN architecture has recall values ranging from 94.147% (obtained with the CWD) to 95.867% (achieved with the ZAMD). The recall values of EfficientNet cover the range between 94.280% (obtained with the RIDT) and 95.533% (obtained with the ZAMD).

The overall recall values of the TFD-CNN models range from 94.147% (obtained by the CWD of the input data and the Xception architecture) to 95.867% (achieved by the ZAMD and the Xception architecture). Thus, the recall values of the TFD-CNN models significantly surpass the recall value of the baseline model by 5.294% to 7.014%.

Table 6.4 Recall values for evaluating the deep learning models on the test dataset. The highest recall values for each considered CNN architecture are marked in bold.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	**0.95533**	0.94880	0.94907
BUD	0.95187	0.95147	0.94720
CWD	0.94547	0.94147	0.94787
PWVD	0.94427	0.94947	0.95053
RIDB	0.94493	0.95333	0.94440
RIDBN	0.95067	0.95200	0.94853
RIDH	0.94240	0.94667	0.94947
RIDT	0.94813	0.94973	0.94280
SP	0.94747	0.94453	0.94720
SPWVD	0.95000	0.95120	0.94813
WVD	0.94253	0.95187	0.94467
ZAMD	0.94787	**0.95867**	**0.95533**
Baseline model		0.88853	

6.5 Precision

Table 6.5 presents the precision values for the evaluated deep learning models. The results presented in Table 6.5 show that the baseline model provides a precision of 97.200%, while the TFD-CNN models offer higher precision values. The precision values of the ResNet-101 architecture are in the range from 98.070% (for the BJD) to 99.328% (for the CWD), the Xception architecture provides precision values ranging from 97.730% (for

the ZAMD) to 99.507% (for the CWD), while the EfficientNet architecture has precision values between 97.549% (for the ZAMD) and 99.454% (for the SP).

Considering the performances across all three CNN architectures, the precision values of the evaluated TFD-CNN models range from 97.549% (for the ZAMD - EfficientNet model) to 99.507% (for the CWD - Xception model). Thus, the precision values of the TFD-CNN models are 0.349% to 2.307% higher than the precision achieved by the baseline model.

Table 6.5 Precision values for evaluating the deep learning models on the test dataset. The highest precision values for each considered CNN architecture are marked in bold.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	0.98070	0.98669	0.99026
BUD	0.98428	0.98455	0.99024
CWD	**0.99328**	**0.99507**	0.99135
PWVD	0.99313	0.98944	0.98753
RIDB	0.99174	0.98160	0.99188
RIDBN	0.98740	0.98564	0.98696
RIDH	0.99298	0.99024	0.99013
RIDT	0.98736	0.98698	0.99396
SP	0.99038	0.99425	**0.99454**
SPWVD	0.98466	0.98809	0.98956
WVD	0.98770	0.98851	0.99133
ZAMD	0.98791	0.97730	0.97549
Baseline model		0.97200	

6.6 F1 Score

The F1 score values obtained by the baseline model and the TFD-CNN models are given in Table 6.6. The presented results show that the baseline model achieves an F1 score of 92.839%, which is surpassed by the F1 score values obtained by the TFD-CNN models. The ResNet-101 architecture offers the F1 score values from the range between 96.459% (for the WVD) and 96.878% (for the CWD), the Xception architecture provides the F1 score values between 96.726% (for the RIDB) and 96.984% (for the WVD), while the F1 score values obtained by EfficientNet are between 96.531% (for the ZAMD) and 97.029% (for the SP).

Overall, the evaluated TFD-CNN models provide F1 score values ranging from 96.459% (for the WVD - ResNet-101 combination) to 97.029% (for the SP - EfficientNet combination), thus outperforming the F1 score of the baseline model by 3.620% to 4.190%.

Table 6.6 F1 score values for evaluating the deep learning models on the test dataset. The highest F1 score values for each considered CNN architecture are marked in bold.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	0.96785	0.96737	0.96923
BUD	0.96780	0.96772	0.96824
CWD	**0.96878**	0.96753	0.96912
PWVD	0.96808	0.96904	0.96868
RIDB	0.96777	0.96726	0.96756
RIDBN	0.96868	0.96853	0.96736
RIDH	0.96703	0.96796	0.96937
RIDT	0.96735	0.96800	0.96770
SP	0.96845	0.96875	**0.97029**
SPWVD	0.96702	0.96929	0.96841
WVD	0.96459	**0.96984**	0.96743
ZAMD	0.96747	0.96789	0.96531
Baseline model		0.92839	

6.7 PR AUC

The PR AUC is the final metric used to evaluate the performance of the considered deep learning models, and the obtained values are summarized in Table 6.7. As seen in Table 6.7, the PR AUC value of the baseline model is 0.97720, while the TFD-CNN models exceed this value. The PR AUC values of the ResNet-101 architecture cover the range between 0.99002 (for the WVD) and 0.99163 (for the BUD), the Xception architecture achieves the PR AUC values between 0.99053 (for the RIDT) and 0.99165 (for the SP), while the PR AUC values obtained by EfficientNet range between 0.98989 (for the RIDT) and 0.99195 (for the CWD).

The results presented in Table 6.7 show that the overall PR AUC values obtained by the evaluated TFD-CNN models are in the range between 0.98989 (for the combination of the RIDT of the input data and the EfficientNet CNN architecture) and 0.99195 (for the combination of the CWD of the input data and the EfficientNet deep learning architecture). Therefore, these values surpass the PR AUC value obtained by the baseline model by 1.269% to 1.475%.

Table 6.7 PR AUC values for evaluating the deep learning models on the test dataset. The highest PR AUC values for each considered CNN architecture are marked in bold.

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	0.99159	0.99111	0.99172
BUD	**0.99163**	0.99076	0.99090
CWD	0.99118	0.99125	**0.99195**
PWVD	0.99083	0.99115	0.99090
RIDB	0.99162	0.99119	0.99068
RIDBN	0.99161	0.99148	0.99126
RIDH	0.99059	0.99128	0.99159
RIDT	0.99097	0.99053	0.98989
SP	0.99122	**0.99165**	0.99179
SPWVD	0.99089	0.99161	0.99141
WVD	0.99002	0.99115	0.99012
ZAMD	0.99098	0.99145	0.99127
Baseline model		0.97720	

6.8 Additional Performance Indicators

Even more detailed insight into the classification performance of the considered deep learning models is possible by analyzing the additional performance indicators, including confusion matrices, ROC curves, and precision-recall curves. In this chapter, these performance indicators are presented for the baseline deep learning model and nine selected TFD-CNN models: the CWD - ResNet-101, the RIDBN - ResNet-101, the SP - ResNet-101, the WVD - Xception, the SPWVD - Xception, the PWVD - Xception, the SP - EfficientNet, the RIDH - EfficientNet, and the BJD - EfficientNet model. These TFD-CNN models were selected for illustration purposes based on the achieved classification accuracy. Namely, three TFDs that provide the highest classification accuracy values for that architecture were chosen for each considered deep learning architecture.

Figure 6.1 shows the confusion matrices for the baseline model and the selected TFD-CNN models. In addition, the ROC curves obtained by evaluating the selected models are depicted in Figure 6.2. Finally, Figure 6.3 presents the precision-recall curves of the considered deep learning models. The performance indicators were obtained by evaluating the deep learning models on the test dataset.

The complete results are presented in Appendix A, which contains the confusion matrices, the ROC curves, and the precision-recall curves for each of the TFDs of the input data in combination with each considered CNN architecture (ResNet-101, Xception, and EfficientNet). The presented confusion matrices, ROC curves, and precision-recall curves confirm consistently high classification performances of the TFD-CNN models,

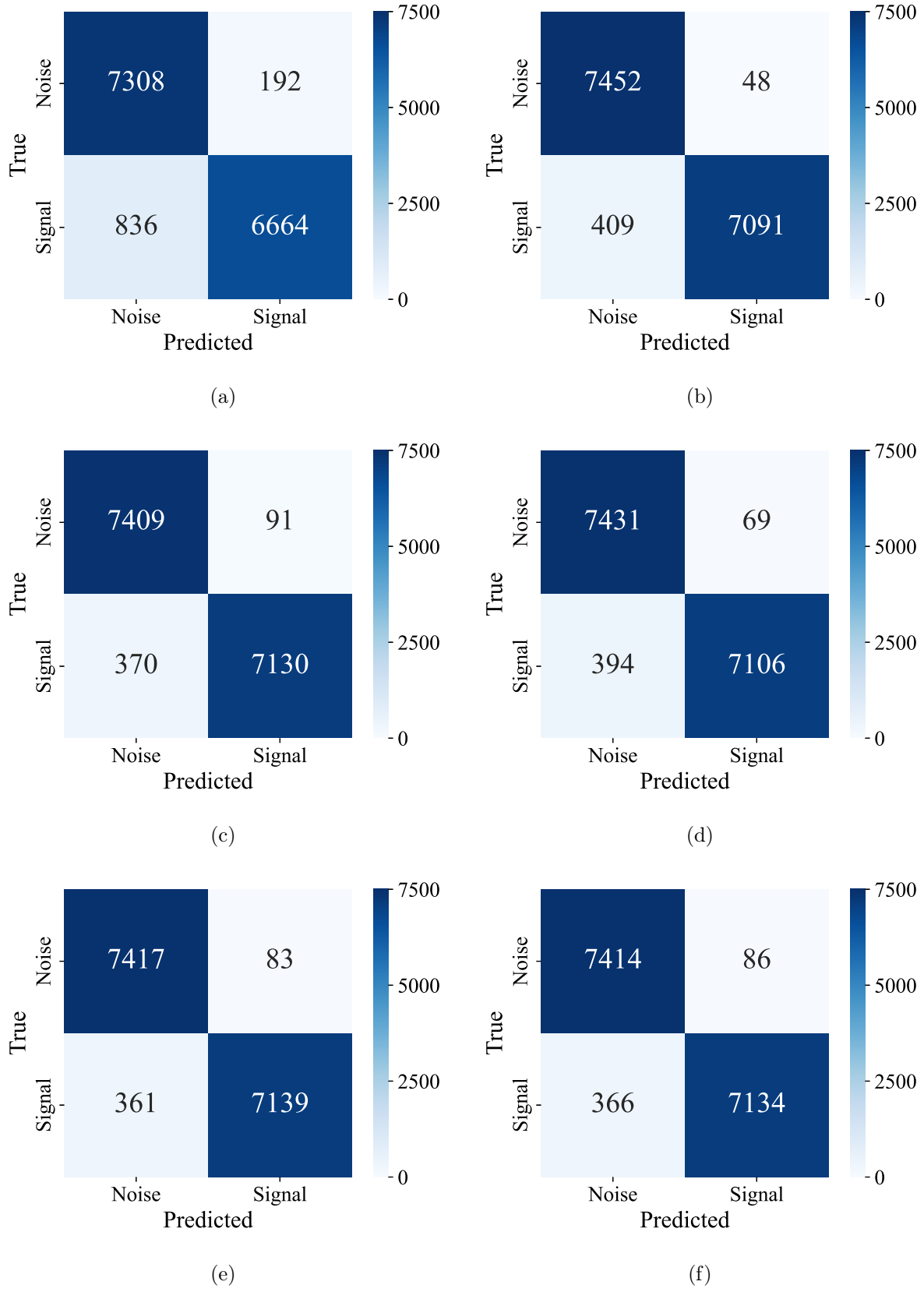


Figure 6.1 Confusion matrices for evaluating the selected deep learning models on the test dataset: (a) Baseline model; (b) CWD - ResNet-101 model; (c) RIDBN - ResNet-101 model; (d) SP - ResNet-101 model; (e) WVD - Xception model; (f) SPWVD - Xception model; (g) PWVD - Xception model; (h) SP - EfficientNet model; (i) RIDH - EfficientNet model; (j) BJD - EfficientNet model.

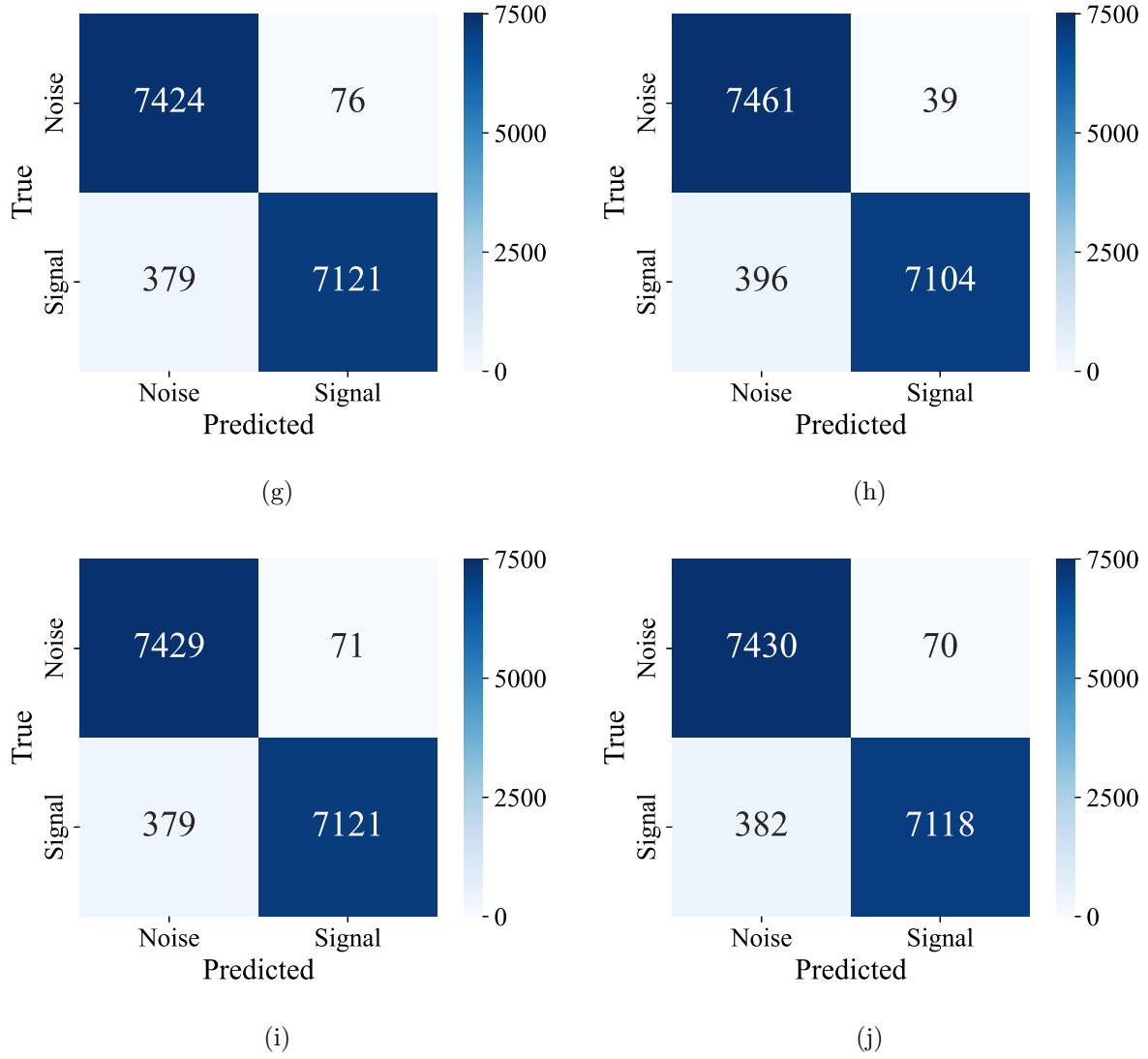


Figure 6.1 (cont.) Confusion matrices for evaluating the selected deep learning models on the test dataset: (a) Baseline model; (b) CWD - ResNet-101 model; (c) RIDBN - ResNet-101 model; (d) SP - ResNet-101 model; (e) WVD - Xception model; (f) SPWVD - Xception model; (g) PWVD - Xception model; (h) SP - EfficientNet model; (i) RIDH - EfficientNet model; (j) BJD - EfficientNet model.

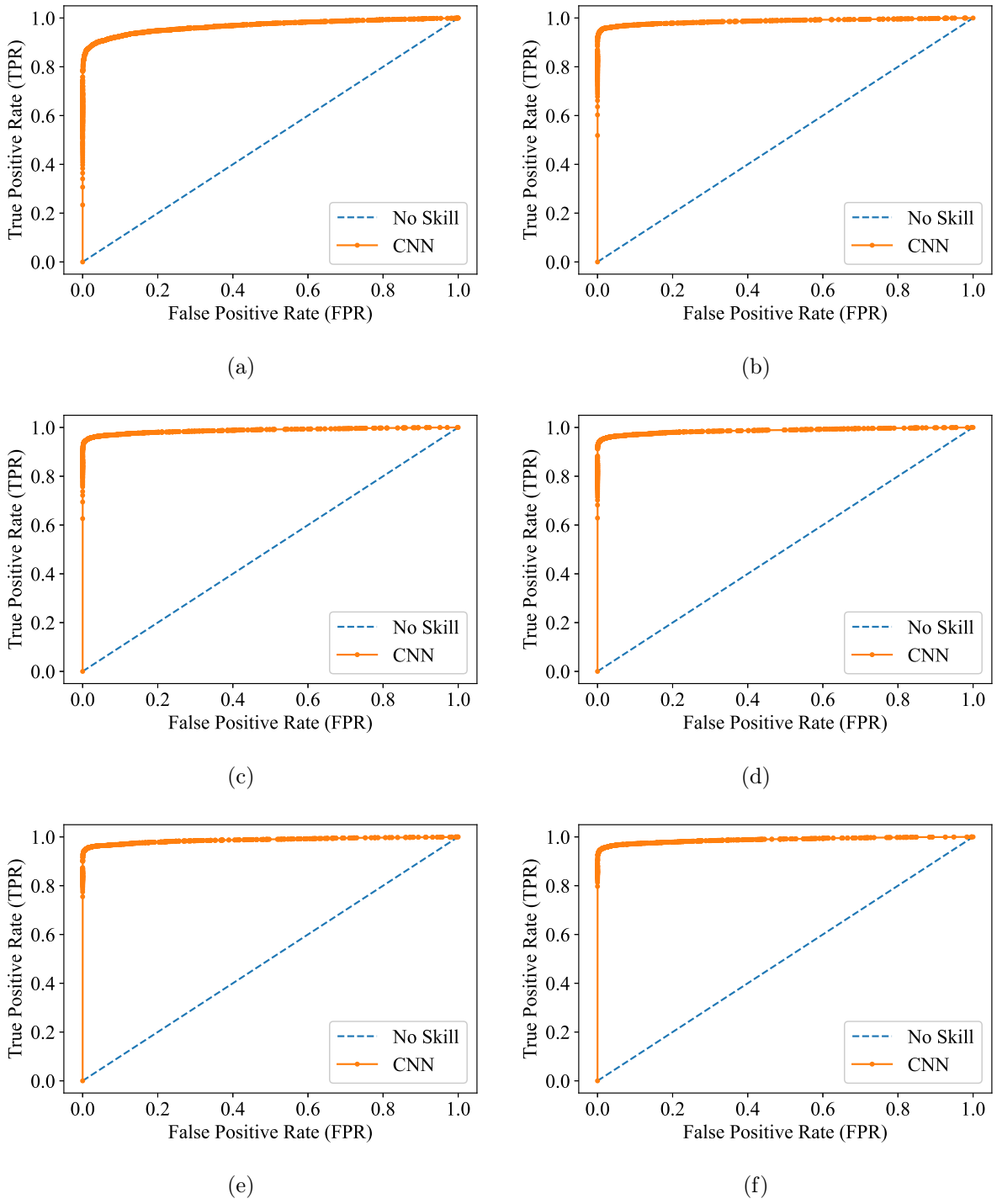
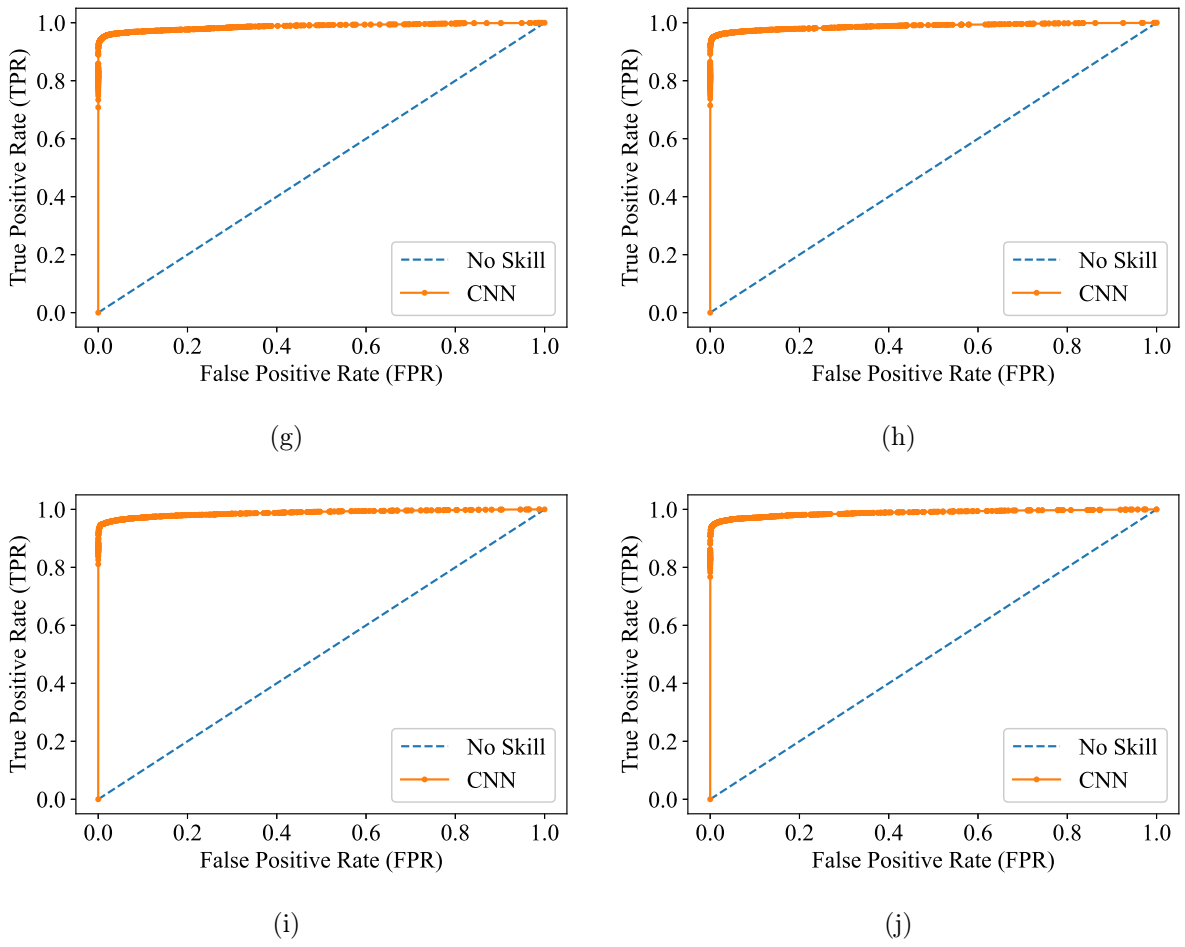


Figure 6.2 ROC curves for evaluating the selected deep learning models on the test dataset: (a) Baseline model; (b) CWD - ResNet-101 model; (c) RIDBN - ResNet-101 model; (d) SP - ResNet-101 model; (e) WVD - Xception model; (f) SPWVD - Xception model; (g) PWVD - Xception model; (h) SP - EfficientNet model; (i) RIDH - EfficientNet model; (j) BJD - EfficientNet model.



(g) (h)

(i) (j)

Figure 6.2 (cont.) ROC curves for evaluating the selected deep learning models on the test dataset: (a) Baseline model; (b) CWD - ResNet-101 model; (c) RIDBN - ResNet-101 model; (d) SP - ResNet-101 model; (e) WVD - Xception model; (f) SPWVD - Xception model; (g) PWVD - Xception model; (h) SP - EfficientNet model; (i) RIDH - EfficientNet model; (j) BJD - EfficientNet model.

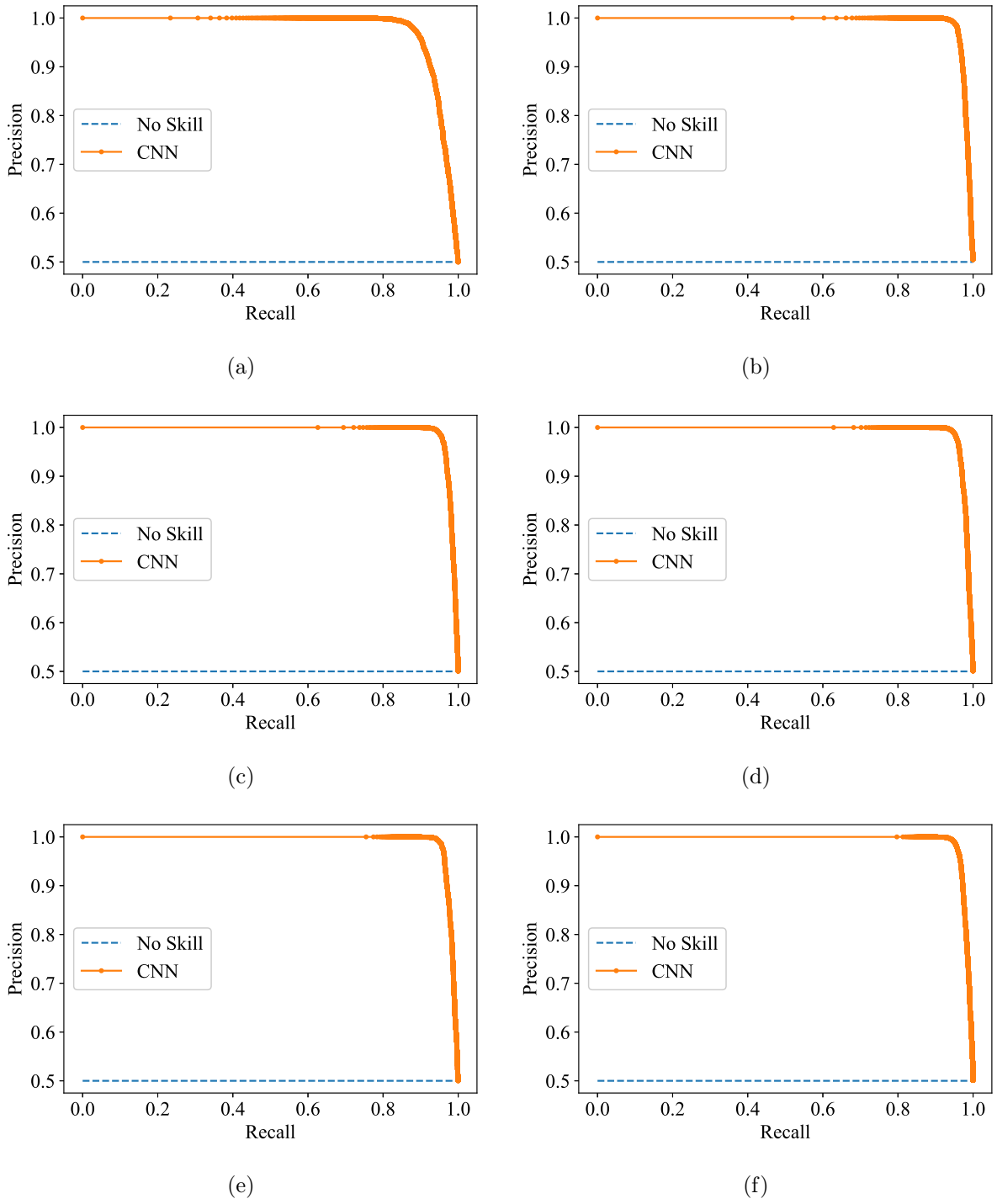


Figure 6.3 Precision-recall curves for evaluating the selected deep learning models on the test dataset: (a) Baseline model; (b) CWD - ResNet-101 model; (c) RIDBN - ResNet-101 model; (d) SP - ResNet-101 model; (e) WVD - Xception model; (f) SPWVD - Xception model; (g) PWVD - Xception model; (h) SP - EfficientNet model; (i) RIDH - EfficientNet model; (j) BJD - EfficientNet model.

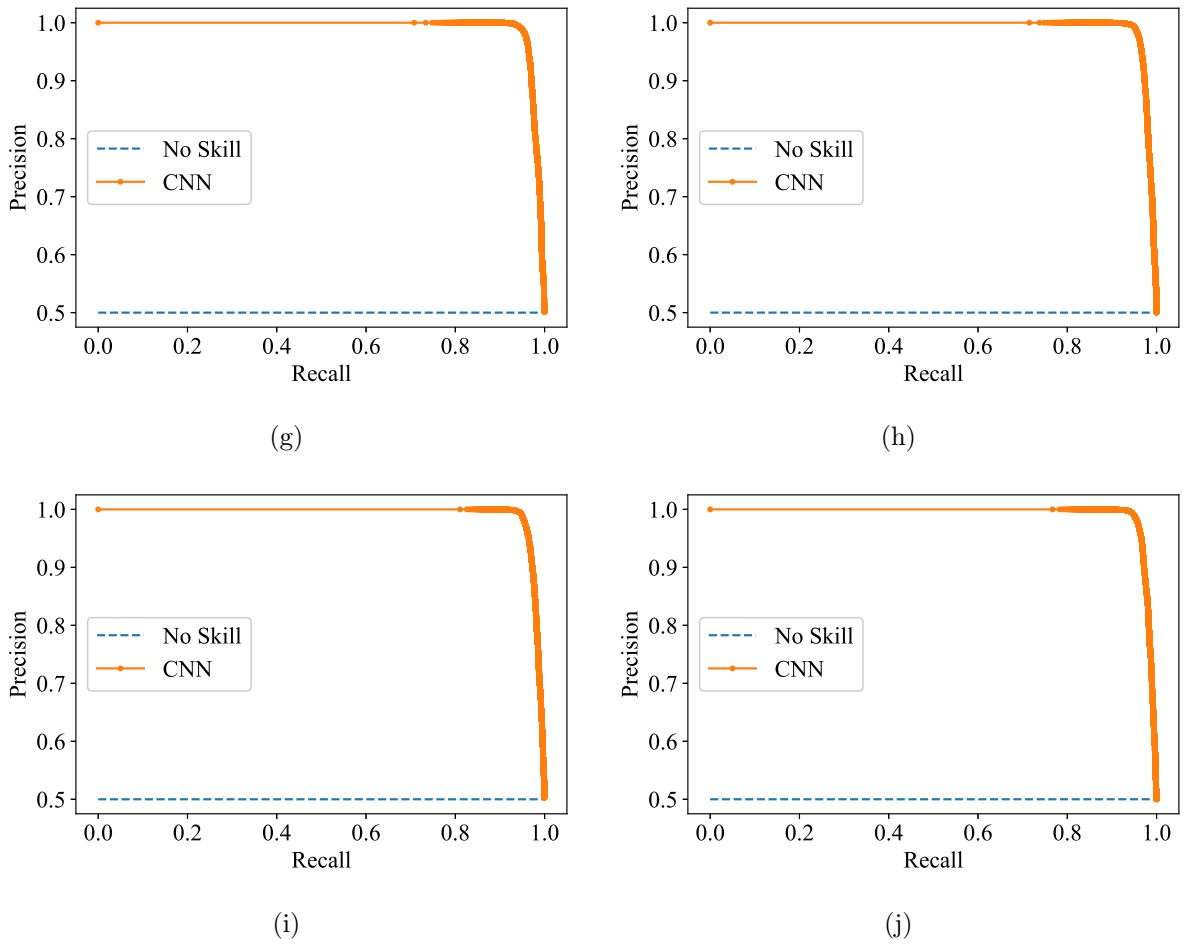


Figure 6.3 (cont.) Precision-recall curves for evaluating the selected deep learning models on the test dataset: (a) Baseline model; (b) CWD - ResNet-101 model; (c) RIDBN - ResNet-101 model; (d) SP - ResNet-101 model; (e) WVD - Xception model; (f) SPWVD - Xception model; (g) PWVD - Xception model; (h) SP - EfficientNet model; (i) RIDH - EfficientNet model; (j) BJD - EfficientNet model.

which surpass the performance of the baseline model.

6.9 Statistical Test Results

The final step in analyzing the obtained classification results consists of performing McNemar's statistical tests according to the procedure described in Section 5.7. This step is necessary to examine the statistical significance of the obtained results.

For illustration purposes, three contingency tables used in McNemar's statistical tests are selected and provided here. The tables were used to compare the baseline model with the three selected TFD-CNN models that provided the highest accuracy for each considered deep CNN architecture. Thus, Tables 6.8, 6.9, and 6.10 represent the contingency tables used to compare the baseline model with the CWD - ResNet-101 model, the WVD - Xception model, and the SP - EfficientNet model, respectively. The rest of the contingency tables are available in Appendix B.

Table 6.8 Contingency table for the baseline model and the CWD - ResNet-101 model.

	CWD - ResNet-101 model Correct	**CWD - ResNet-101 model Incorrect**
Baseline model Correct	13895	77
Baseline model Incorrect	648	380

Table 6.9 Contingency table for the baseline model and the WVD - Xception model.

	WVD - Xception model Correct	**WVD - Xception model Incorrect**
Baseline model Correct	13869	103
Baseline model Incorrect	687	341

Table 6.10 Contingency table for the baseline model and the SP - EfficientNet model.

	SP - EfficientNet model Correct	**SP - EfficientNet model Incorrect**
Baseline model Correct	13910	62
Baseline model Incorrect	655	373

The p-values obtained by McNemar's statistical test for each combination of the TFD of the input data and the CNN architecture are summarized in Table 6.11. As seen in Table 6.11, each calculated p-value is lower than the specified significance level ($p < 0.00333$). Indeed, the calculated p-values are very close to zero (considering the numerical solver's limitations), indicating high statistical significance. Therefore, the null hypothesis of McNemar's statistical test can be rejected, and it can be concluded that the differences between the classification results of the TFD-CNN models and those of the baseline deep learning model are statistically significant.

Table 6.11 p-Values obtained for each considered TFD-CNN model by McNemar's statistical test ($\alpha = 0.00333$).

	CNN architecture		
TFD	**ResNet-101**	**Xception**	**EfficientNet**
BJD	5.78×10^{-81}	2.05×10^{-85}	2.31×10^{-96}
BUD	4.19×10^{-84}	6.59×10^{-84}	5.14×10^{-93}
CWD	1.83×10^{-99}	1.14×10^{-98}	1.93×10^{-98}
PWVD	7.41×10^{-97}	9.51×10^{-97}	1.16×10^{-91}
RIDB	8.81×10^{-95}	3.06×10^{-79}	1.19×10^{-92}
RIDBN	3.98×10^{-92}	8.20×10^{-89}	2.05×10^{-85}
RIDH	5.68×10^{-91}	5.84×10^{-92}	8.66×10^{-96}
RIDT	2.18×10^{-87}	3.41×10^{-89}	3.72×10^{-97}
SP	1.26×10^{-93}	9.85×10^{-100}	2.61×10^{-108}
SPWVD	8.38×10^{-84}	3.02×10^{-93}	1.15×10^{-93}
WVD	5.71×10^{-79}	1.44×10^{-95}	9.73×10^{-93}
ZAMD	4.04×10^{-87}	2.17×10^{-78}	1.43×10^{-67}

6.10 Results Summary

To sum up the elaboration of the achieved results, the conducted comparative analysis of the obtained classification results indicates that the application of the advanced Cohen's class TFDs to the noisy, non-stationary input time-series GW data significantly improves the classification performance achieved by the TFD-CNN-based deep learning algorithms compared to the baseline CNN model trained only on the original time series. The TFD-CNN models show improved classification performance in terms of all considered evaluation metrics, i.e., the proposed method outperforms the baseline model by 3.393% to 3.953% in terms of classification accuracy, 1.718% to 2.067% in terms of ROC AUC, 5.294% to 7.014% in terms of recall, 0.349% to 2.307% in terms of precision, 3.620% to 4.190% in terms of F1 score, and 1.269% to 1.475% in terms of PR AUC.

Furthermore, the enhanced performance of the proposed method is also demonstrated by analyzing the additional performance indicators, including the confusion matrices,

ROC curves, and precision-recall curves. Finally, the obtained differences in the classification performances of the proposed method and the baseline model are confirmed to be statistically significant by McNemar's statistical test.

Each considered Cohen's class TFD of noisy, non-stationary input data yields high values of the evaluation metrics when used as input to each considered deep CNN architecture. Moreover, the obtained values of the evaluation metrics are also very similar to those obtained using the SP representation of the considered data. Namely, the SP data representation is commonly used for time-frequency signal analysis in many fields dealing with non-stationary signals. Therefore, the results presented in this chapter indicate that the alternative TFDs of Cohen's class are also a viable and robust solution that can successfully improve the performance of deep learning algorithms for classification between data examples with non-stationary signals in intensive noise and those containing only noise.

Finally, the increased accuracy of the deep learning classification algorithms contributes to improved detection of non-stationary signals embedded in the noisy background. In this thesis, the proposed detection method is demonstrated in the example of detecting GW signals. It has been shown that the proposed approach can be efficiently applied in GW research to further improve the detection of GW events. Moreover, the proposed detection approach is general and has great potential to be successfully applied to detect various noisy, non-stationary signals in other fields and practical applications.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

This thesis proposes a method for detecting non-stationary GW signals in intensive noise based on deep learning algorithms and Cohen's class of TFDs. In order to validate the proposed approach, the experimental setup was developed, including data preparation, training and testing deep learning models, and performance evaluation. During the data generation procedure, the real-life recordings were retrieved from LIGO detectors, and the extensive simulations of the GW waveforms were conducted, thus obtaining a diverse time-series dataset, allowing testing the proposed method in intensive real-life, non-stationary, non-Gaussian, and non-white noise, with the SNR values between -123.46 and -2.27 dB.

After preprocessing the time-series data, 12 TFDs from Cohen's class, including BJD, BUD, CWD, PWVD, RIDB, RIDBN, RIDH, RIDT, SP, SPWVD, WVD, and ZAMD, were calculated. Thus obtained 12 TFD datasets (a total of 1.2 million TFD images) were used as input to the deep learning classification algorithms based on three state-of-the-art two-dimensional CNN architectures – ResNet-101, Xception, and EfficientNet. Each of 36 obtained TFD-CNN models was trained and evaluated on the corresponding dataset. The obtained results show excellent classification performance of the proposed approach, with classification accuracy between 96.540% and 97.100%, ROC AUC between 0.98505 and 0.98854, recall between 94.147% and 95.867%, precision between 97.549% and 99.507%, F1 score between 96.459% and 97.029%, and PR AUC between 0.98989 and 0.99195.

Moreover, the performance of the proposed approach was compared to the performance of the baseline deep learning model trained on the original time-series data. The comparison has shown that the proposed approach outperforms the baseline model by 3.393% to 3.953%, 1.718% to 2.067%, 5.294% to 7.014%, 0.349% to 2.307%, 3.620% to 4.190%, and 1.269% to

1.475% in terms of classification accuracy, ROC AUC, recall, precision, F1 score, and PR AUC, respectively. The additional analysis of the obtained confusion matrices, ROC curves, and precision-recall curves further supports the superior performance of the proposed method, whose statistical significance was confirmed by McNemar's statistical test.

Therefore, the results obtained by the proposed approach suggest that applying alternative TFDs from Cohen's class coupled with deep learning algorithms can improve the classification of non-stationary time-series signals in intensive noise. Namely, these quadratic, high-resolution TFDs provide more-informative signal representation with improved intelligibility compared to the time series, which deep CNN architectures can efficiently utilize.

The proposed method for detecting non-stationary signals in noise is in this thesis demonstrated on the example of detecting GW signals. The obtained high performance on this challenging task proves its applicability in GW data analysis, which can further contribute to the improved GW detection rates. However, the application of the proposed method is not limited to GW data only, i.e., it can be easily extended to other fields of non-stationary signal analysis.

7.2 Future Research Directions

As mentioned above, the proposed approach has great potential to be implemented in different practical applications requiring the classification of noisy, non-stationary signals. These applications include but are not limited to analysis of seismic, speech, radar, EEG, and ECG signals, to name a few. However, the application of the proposed approach in these fields remains a potential focus of future research.

Based on the benefits of introducing TFDs of Cohen's class to deep learning-based signal classification demonstrated in this thesis, these TFDs could also be utilized for data augmentation. Namely, in the case of datasets with reduced size, multiple TFDs could be calculated from the initially available time-series data, thus obtaining additional data for training and testing deep learning models. Moreover, the research presented in this thesis could be extended to using multiple TFDs of the same input time-series signal as different channels of the deep learning models' input. Furthermore, an ensemble of deep learning models could also be utilized to classify Cohen's class TFDs of the input time series to investigate whether even higher classification performances could be achieved than those obtained by a single deep learning model.

Finally, future research might include additional preprocessing techniques prior to the TFD calculation and applying the proposed approach (such as the locally-adaptive, data-driven denoising techniques applied to the noisy, non-stationary signals) and studying their effects on the deep learning classification performance.

BIBLIOGRAPHY

[1] J. Aasi *et al.*, "Advanced LIGO," *Classical and Quantum Gravity*, vol. 32, no. 7, Mar. 2015, Art. no. 074001, doi: 10.1088/0264-9381/32/7/074001.

[2] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[3] B. P. Abbott *et al.*, "GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs," *Physical Review X*, vol. 9, Sep. 2019, Art. no. 031040, doi: 10.1103/PhysRevX.9.031040.

[4] ——, "A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals," *Classical and Quantum Gravity*, vol. 37, no. 5, Feb. 2020, Art. no. 055002, doi: 10.1088/1361-6382/ab685e.

[5] ——, "Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914," *Classical and Quantum Gravity*, vol. 33, no. 13, Jun. 2016, Art. no. 134001, doi: 10.1088/0264-9381/33/13/134001.

[6] ——, "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, no. 6, Feb. 2016, Art. no. 061102, doi: 10.1103/PhysRevLett.116.061102.

[7] ——, "Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA," *Living Reviews in Relativity*, vol. 23, Sep. 2020, art. no. 3, doi: 10.1007/s41114-020-00026-9.

[8] R. Abbott *et al.*, "GWTC-2: Compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run," *Physical Review X*, vol. 11, no. 2, Jun. 2021, Art. no. 021053, doi: 10.1103/PhysRevX.11.021053.

[9] ——, "GWTC-2.1: Deep extended catalog of compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run," *arXiv preprint arXiv:2108.01045*, 2021.

[10] ——, "GWTC-3: Compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run," *arXiv preprint arXiv:2111.03606*, 2021.

[11] F. Acernese *et al.*, "Advanced Virgo: a second-generation interferometric gravitational wave detector," *Classical and Quantum Gravity*, vol. 32, no. 2, Dec. 2014, art. no. 024001, doi: 10.1088/0264-9381/32/2/024001.

[12] ——, "Advanced Virgo status," in *Journal of Physics: Conference Series*, vol. 1342, 2020, Art. no. 012010, doi: 10.1088/1742-6596/1342/1/012010.

[13] M. H. Ackroyd, "Short-time spectra and time-frequency energy distributions," *The Journal of the Acoustical Society of America*, vol. 50, no. 5A, pp. 1229–1231, Nov. 1971, doi: 10.1121/1.1912761.

[14] T. Adams *et al.*, "Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era," *Classical and Quantum Gravity*, vol. 33, no. 17, Aug. 2016, Art. no. 175012, doi: 10.1088/0264-9381/33/17/175012.

[15] T. Akutsu *et al.*, "KAGRA: 2.5 generation interferometric gravitational wave detector," *Nature Astronomy*, vol. 3, pp. 35–40, Jan. 2019, doi: 10.1038/s41550-018-0658-y.

[16] B. Allen, "χ^2 time-frequency discriminator for gravitational wave detection," *Physical Review D*, vol. 71, no. 6, Mar. 2005, Art. no. 062001, doi: 10.1103/PhysRevD.71.062001.

[17] 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," *Physical Review D*, vol. 85, no. 12, Jun. 2012, Art. no. 122006, doi: 10.1103/PhysRevD.85.122006.

[18] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete Fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, Jun. 1977, doi: 10.1109/TASSP.1977.1162950.

[19] J. D. Álvares *et al.*, "Exploring gravitational-wave detection and parameter inference using deep learning methods," *Classical and Quantum Gravity*, vol. 38, no. 15, Jul. 2021, Art. no. 155010, doi: 10.1088/1361-6382/ac0455.

[20] B. Ambikapathy, K. Kirshnamurthy, and R. Venkatesan, "Assessment of electromyograms using genetic algorithm and artificial neural networks," *Evolutionary Intelligence*, vol. 14, pp. 261–271, Jun. 2021, doi: 10.1007/s12065-018-0174-0.

[21] R. A. Armstrong, "When to use the Bonferroni correction," *Ophthalmic and Physiological Optics*, vol. 34, no. 5, pp. 502–508, Apr. 2014, doi: 10.1111/opo.12131.

[22] P. Astone *et al.*, "New method to observe gravitational waves emitted by core collapse supernovae," *Physical Review D*, vol. 98, no. 12, Dec. 2018, Art. no. 122002, doi: 10.1103/PhysRevD.98.122002.

[23] F. Aubin *et al.*, "The MBTA pipeline for detecting compact binary coalescences in the third LIGO–Virgo observing run," *Classical and Quantum Gravity*, vol. 38, no. 9, Apr. 2021, Art. no. 095004, doi: 10.1088/1361-6382/abe913.

[24] F. Auger, "Some simple parameter determination rules for the generalized Choi-Williams and Butterworth distributions," *IEEE Signal Processing Letters*, vol. 1, no. 1, pp. 9–11, Jan. 1994, doi: 10.1109/97.295313.

[25] F. Auger, P. Flandrin, P. Gonçalvès, and O. Lemoine, *Time-Frequency Toolbox*. France/USA: CNRS/Rice University, 1996.

[26] ——, *Time-Frequency Toolbox: Reference Guide*. France/USA: CNRS/Rice University, 1996.

[27] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *2017 International Conference on Platform Technology and Service (PlatCon 2017)*. Busan, South Korea: IEEE, Feb. 2017, pp. 1–5, doi: 10.1109/PlatCon.2017.7883728.

[28] S. Bahaadini *et al.*, "Machine learning for Gravity Spy: Glitch classification and dataset," *Information Sciences*, vol. 444, pp. 172–186, May 2018, doi: 10.1016/j.ins.2018.02.068.

[29] G. Baldini and R. Giuliani, "An assessment of the impact of wireless interferences on IoT emitter identification using time frequency representations and CNN," in *2019 Global IoT Summit (GIoTS 2019)*. Aarhus, Denmark: IEEE, Jun. 2019, pp. 1–6, doi: 10.1109/GIOTS.2019.8766385.

[30] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, Jun. 2004, doi: 10.1145/1007730.1007735.

[31] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural Networks: Tricks of the Trade*, 2nd ed., ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Germany: Springer, 2012, vol. 7700, pp. 437–478.

[32] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: 10.1109/TPAMI.2013.50.

[33] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of Machine Learning Research*, vol. 13, pp. 281–305, Feb. 2012.

[34] D. G. Blair, Ed., *The Detection of Gravitational Waves*. Cambridge, UK: Cambridge University Press, 1991, doi: 10.1017/CBO9780511600104.

[35] B. Boashash, "Note on the use of the Wigner distribution for time-frequency signal analysis," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 9, pp. 1518–1521, Sep. 1988, doi: 10.1109/29.90380.

[36] B. Boashash, Ed., *Time-Frequency Signal Analysis and Processing: A Comprehensive Reference*, 2nd ed., ser. EURASIP and Academic Press Series in Signal and Image Processing. London, UK: Academic Press, 2016.

[37] A. Bohé *et al.*, "Improved effective-one-body model of spinning, nonprecessing binary black holes for the era of gravitational-wave astrophysics with advanced detectors," *Physical Review D*, vol. 95, Feb. 2017, Art. no. 044028, doi: 10.1103/PhysRevD.95.044028.

[38] C. Bonferroni, "Teoria statistica delle classi e calcolo delle probabilita," *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, vol. 8, pp. 3–62, 1936.

[39] L. Boubchir, S. Al-Maadeed, and A. Bouridane, "On the use of time-frequency features for detecting and classifying epileptic seizure activities in non-stationary EEG signals," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*. Florence, Italy: IEEE, May 2014, pp. 5889–5893, doi: 10.1109/ICASSP.2014.6854733.

[40] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *27th International Conference on Machine Learning (ICML 2010)*, International Machine Learning Society (IMLS). Haifa, Israel: Omnipress, Jun. 2010, pp. 111–118.

[41] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–50, Aug. 2016, doi: 10.1145/2907070.

[42] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.

[43] J. C. Brown, "Calculation of a constant Q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, Jan. 1991, doi: 10.1121/1.400476.

[44] J. Brownlee, "How to calculate McNemar's test to compare two machine learning classifiers," Machine Learning Mastery, Aug. 2019, (accessed Aug. 6, 2021). [Online]. Available: https://machinelearningmastery.com/mcnemars-test-for-machine-learning/

[45] ——, *Statistical Methods for Machine Learning*, 2019.

[46] ——, "Tour of evaluation metrics for imbalanced classification," Machine Learning Mastery, May 2021, (accessed Aug. 6, 2021). [Online]. Available: https://machinelearningmastery.com/tour-of-evaluation-metrics-for-imbalanced-classification/

[47] A. Buja, W. Stuetzle, and Y. Shen, "Loss functions for binary class probability estimation and classification: Structure and applications," University of Washington, Tech. Rep., Nov. 2005.

[48] K. Cai, W. Cao, L. Aarniovuori, H. Pang, Y. Lin, and G. Li, "Classification of power quality disturbances using Wigner-Ville distribution and deep convolutional neural networks," *IEEE Access*, vol. 7, pp. 119 099–119 109, Aug. 2019, doi: 10.1109/ACCESS.2019.2937193.

[49] K. Cannon *et al.*, "Toward early-warning detection of gravitational waves from compact binary coalescence," *The Astrophysical Journal*, vol. 748, no. 2, Mar. 2012, Art. no. 136, doi: 10.1088/0004-637x/748/2/136.

[50] ——, "GstLAL: A software framework for gravitational wave discovery," *SoftwareX*, vol. 14, Jun. 2021, Art. no. 100680, doi: 10.1016/j.softx.2021.100680.

[51] S. Celin and K. Vasanth, "ECG signal classification using various machine learning techniques," *Journal of Medical Systems*, vol. 42, Oct. 2018, Art. no. 241, doi: 10.1007/s10916-018-1083-6.

[52] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, no. 1, pp. 89–97, Jan. 2004, doi: 10.1023/B:JMIV.0000011325.36760.1e.

[53] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*, ser. Radon Series on Computational and Applied Mathematics, M. Fornasier, Ed. Berlin/New York, Germany/USA: De Gruyter, 2010, vol. 9, pp. 263–340.

[54] M. L. Chan, I. S. Heng, and C. Messenger, "Detection and classification of supernova gravitational wave signals: A deep learning approach," *Physical Review D*, vol. 102, no. 4, Aug. 2020, Art. no. 043022, doi: 10.1103/PhysRevD.102.043022.

[55] S. J. Chang and J. B. Park, "Wire mismatch detection using a convolutional neural network and fault localization based on time-frequency-domain reflectometry," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 3, pp. 2102–2110, Mar. 2019, doi: 10.1109/TIE.2018.2835386.

[56] S. Chatterji, L. Blackburn, G. Martin, and E. Katsavounidis, "Multiresolution techniques for the detection of gravitational-wave bursts," *Classical and Quantum Gravity*, vol. 21, no. 20, pp. 1809–1818, Sep. 2004, doi: 10.1088/0264-9381/21/20/024.

[57] H. Choi and W. J. Williams, "Improved time-frequency representation of multicomponent signals using exponential kernels," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 6, pp. 862–871, Jun. 1989, doi: 10.1109/ASSP.1989.28057.

[58] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*. Honolulu, HI, USA: IEEE, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.

[59] A. J. K. Chua, C. R. Galley, and M. Vallisneri, "Reduced-order modeling with artificial neurons for gravitational-wave inference," *Physical Review Letters*, vol. 122, no. 21, May 2019, Art. no. 211101, doi: 10.1103/PhysRevLett.122.211101.

[60] A. J. K. Chua and M. Vallisneri, "Learning Bayesian posteriors with neural networks for gravitational-wave inference," *Physical Review Letters*, vol. 124, no. 4, Jan. 2020, Art. no. 041102, doi: 10.1103/PhysRevLett.124.041102.

[61] T. Claasen and W. Mecklenbräuker, "The Wigner distribution - A tool for time-frequency signal analysis, Parts I–III," *Philips Journal of Research*, vol. 35, pp. 217–250, 276–300, 372–389, 1980.

[62] B. Clarke, E. Fokoue, and H. H. Zhang, *Principles and Theory for Data Mining and Machine Learning*, ser. Springer Series in Statistics. New York, NY, USA: Springer Science & Business Media, 2009.

[63] L. Cohen, "Generalized phase-space distribution functions," *Journal of Mathematical Physics*, vol. 7, no. 5, pp. 781–786, May 1966, doi: 10.1063/1.1931206.

[64] ——, *Time-Frequency Analysis*. Upper Saddle River, NJ, USA: Prentice-Hall PTR, 1995.

[65] E. Cordero, M. de Gosson, and F. Nicola, "On the reduction of the interferences in the Born-Jordan distribution," *Applied and Computational Harmonic Analysis*, vol. 44, no. 2, pp. 230–245, Mar. 2018, doi: 10.1016/j.acha.2016.04.007.

[66] A. Costa and G. Boudreau-Bartels, "Design of time-frequency representations using a multiform, tiltable exponential kernel," *IEEE Transactions on Signal Processing*, vol. 43, no. 10, pp. 2283–2301, Oct. 1995, doi: 10.1109/78.469860.

[67] E. Cuoco *et al.*, "Enhancing gravitational-wave science with machine learning," *Machine Learning: Science and Technology*, vol. 2, no. 1, Dec. 2020, Art. no. 011002, doi: 10.1088/2632-2153/abb93a.

[68] C. Cutler and E. E. Flanagan, "Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral waveform?" *Physical Review D*, vol. 49, no. 6, pp. 2658–2697, Mar. 1994, doi: 10.1103/PhysRevD.49.2658.

[69] T. Dal Canton *et al.*, "Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors," *Physical Review D*, vol. 90, no. 8, Oct. 2014, Art. no. 082004, doi: 10.1103/PhysRevD.90.082004.

[70] G. S. Davies, T. Dent, M. Tápai, I. Harry, C. McIsaac, and A. H. Nitz, "Extending the PyCBC search for gravitational waves from compact binary mergers to a global network," *Physical Review D*, vol. 102, no. 2, Jul. 2020, Art. no. 022004, doi: 10.1103/PhysRevD.102.022004.

[71] T. Dent and J. Veitch, "Optimizing gravitational-wave searches for a population of coalescing binaries: Intrinsic parameters," *Physical Review D*, vol. 89, no. 6, Mar. 2014, Art. no. 062002, doi: 10.1103/PhysRevD.89.062002.

[72] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998, doi: 10.1162/089976698300017197.

[73] C. Dreissigacker and R. Prix, "Deep-learning continuous gravitational waves: Multiple detectors and realistic noise," *Physical Review D*, vol. 102, no. 2, Jul. 2020, Art. no. 022005, doi: 10.1103/PhysRevD.102.022005.

[74] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, "Deep-learning continuous gravitational waves," *Physical Review D*, vol. 100, no. 4, Aug. 2019, Art. no. 044009, doi: 10.1103/PhysRevD.100.044009.

[75] Y. Du, Q. Shao, Y. Liu, G. Sheng, and X. Jiang, "Detection of single line-to-ground fault using convolutional neural network and task decomposition framework in distribution systems," in *2018 Condition Monitoring and Diagnosis (CMD 2018)*. Perth, WA, Australia: IEEE, Sep. 2018, pp. 1–4, doi: 10.1109/CMD.2018.8535600.

[76] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, Jul. 2011.

[77] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2018.

[78] A. L. Edwards, "Note on the "correction for continuity" in testing the significance of the difference between correlated proportions," *Psychometrika*, vol. 13, no. 3, pp. 185–187, Sep. 1948, doi: 10.1007/BF02289261.

[79] A. Effler *et al.*, "Environmental influences on the LIGO gravitational wave detectors during the 6th science run," *Classical and Quantum Gravity*, vol. 32, no. 3, Jan. 2015, Art. no. 035017, doi: 10.1088/0264-9381/32/3/035017.

[80] A. Einstein and N. Rosen, "On gravitational waves," *Journal of the Franklin Institute*, vol. 223, no. 1, pp. 43–54, Jan. 1937, doi: 10.1016/S0016-0032(37)90583-0.

[81] A. Einstein, "Näherungsweise Integration der Feldgleichungen der Gravitation," *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pp. 99–108, 1916.

[82] ——, "Über Gravitationswellen," *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, pp. 154–167, 1918.

[83] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning?" in *13th International Conference on Artificial In-*

telligence and Statistics (AISTATS 2010), ser. Proceedings of Machine Learning Research, vol. 9. Sardinia, Italy: PMLR, May 2010, pp. 201–208.

[84] B. S. Everitt, *The Analysis of Contingency Tables*, 2nd ed., ser. Monographs on Statistics and Applied Probability. Boca Raton, FL, USA: Chapman & Hall/CRC, 2019.

[85] 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," *Science China Physics, Mechanics & Astronomy*, vol. 62, no. 6, pp. 1–8, Jun. 2019, doi: 10.1007/s11433-018-9321-7.

[86] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[87] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, Apr. 2006, doi: 10.1109/TPAMI.2006.79.

[88] P. Flandrin, "Some features of time-frequency representations of multicomponent signals," in *1984 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1984)*, vol. 9. San Diego, CA, USA: IEEE, Mar. 1984, pp. 266–269, doi: 10.1109/ICASSP.1984.1172741.

[89] P. Flandrin, *Time-Frequency/Time-Scale Analysis*, 1st ed., ser. Wavelet Analysis and Its Applications. San Diego, CA, USA: Academic Press, 1998, vol. 10.

[90] P. Flandrin and B. Escudié, "An interpretation of the Pseudo-Wigner-Ville distribution," *Signal Processing*, vol. 6, no. 1, pp. 27–36, 1984, doi: 10.1016/0165-1684(84)90048-3.

[91] H. Gabbard, M. Williams, F. Hayes, and C. Messenger, "Matching matched filtering with deep networks for gravitational-wave astronomy," *Physical Review Letters*, vol. 120, no. 14, Apr. 2018, Art. no. 141103, doi: 10.1103/PhysRevLett.120.141103.

[92] A. Gao, Y. Zhu, W. Cai, and Y. Zhang, "Pattern recognition of partial discharge based on VMD-CWD spectrum and optimized CNN with cross-layer feature fusion," *IEEE Access*, vol. 8, pp. 151 296–151 306, Aug. 2020, doi: 10.1109/ACCESS.2020.3017047.

[93] L. Gao, X. Zhang, J. Gao, and S. You, "Fusion image based radar signal feature extraction and modulation recognition," *IEEE Access*, vol. 7, pp. 13 135–13 148, Jan. 2019, doi: 10.1109/ACCESS.2019.2892526.

[94] T. Gebhard, N. Kilbertus, G. Parascandolo, I. Harry, and B. Schölkopf, "CONVWAVE: Searching for gravitational waves with fully convolutional neural nets," in *Workshop on Deep Learning for Physical Sciences (DLPS) at the 31st Conference on Neural Information Processing Systems (NIPS 2017)*. Long Beach, CA, USA: Neural Information Processing Systems (NIPS), Dec. 2017, pp. 1–6.

[95] T. D. Gebhard and N. Kilbertus, "timothygebhard/ggwd: Version 1.0," Apr. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2649359

[96] T. D. Gebhard, N. Kilbertus, I. Harry, and B. Schölkopf, "Convolutional neural networks: A magic bullet for gravitational-wave detection?" *Physical Review D*, vol. 100, no. 6, Sep. 2019, Art. no. 063015, doi: 10.1103/PhysRevD.100.063015.

[97] D. George and E. A. Huerta, "Deep neural networks to enable real-time multimessenger astrophysics," *Physical Review D*, vol. 97, no. 4, Feb. 2018, Art. no. 044039, doi: 10.1103/PhysRevD.97.044039.

[98] D. George and E. Huerta, "Deep learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data," *Physics Letters B*, vol. 778, pp. 64–70, Mar. 2018, doi: 10.1016/j.physletb.2017.12.053.

[99] D. George, H. Shen, and E. A. Huerta, "Classification and unsupervised clustering of LIGO data with Deep Transfer Learning," *Physical Review D*, vol. 97, no. 10, May 2018, Art. no. 101501, doi: 10.1103/PhysRevD.97.101501.

[100] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, ser. Proceedings of Machine Learning Research, vol. 9. Sardinia, Italy: PMLR, May 2010, pp. 249–256.

[101] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, ser. Proceedings of Machine Learning Research, vol. 15, Society for Artificial Intelligence and Statistics (SAIAS). Fort Lauderdale, FL, USA: PMLR, Apr. 2011, pp. 315–323.

[102] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, Apr. 2009, doi: 10.1137/080725891.

[103] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, ser. The Adaptive Computation and Machine Learning. Cambridge, MA, USA: MIT Press, 2016.

[104] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, May 2015.

[105] M. Granata *et al.*, "Mechanical loss in state-of-the-art amorphous optical coatings," *Physical Review D*, vol. 93, no. 1, Jan. 2016, Art. no. 012007, doi: 10.1103/PhysRevD.93.012007.

[106] D. Graupe, *Principles of Artificial Neural Networks*, 3rd ed., ser. Advanced Series in Circuits and Systems. Singapore: World Scientific, 2013, vol. 7.

[107] S. R. Green, C. Simpson, and J. Gair, "Gravitational-wave parameter estimation with autoregressive neural network flows," *Physical Review D*, vol. 102, no. 10, Nov. 2020, Art. no. 104057, doi: 10.1103/PhysRevD.102.104057.

[108] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, May 2018, doi: 10.1016/j.patcog.2017.10.013.

[109] Q. Guo, X. Yu, and G. Ruan, "LPI radar waveform recognition based on deep convolutional neural network transfer learning," *Symmetry*, vol. 11, no. 4, Apr. 2019, Art. no. 540, doi: 10.3390/sym11040540.

[110] Z. Guo, L. Durand, and H. C. Lee, "The time-frequency distributions of nonstationary signals based on a Bessel kernel," *IEEE Transactions on Signal Processing*, vol. 42, no. 7, pp. 1700–1707, Jul. 1994, doi: 10.1109/78.298277.

[111] S. K. Hadjidimitriou and L. J. Hadjileontiadis, "EEG-based classification of music appraisal responses using time-frequency analysis and familiarity ratings," *IEEE*

Transactions on Affective Computing, vol. 4, no. 2, pp. 161–172, Apr. 2013, doi: 10.1109/T-AFFC.2013.6.

[112] C. Hanna *et al.*, "Fast evaluation of multidetector consistency for real-time gravitational wave searches," *Physical Review D*, vol. 101, no. 2, Jan. 2020, Art. no. 022003, doi: 10.1103/PhysRevD.101.022003.

[113] M. Hatamzadeh, R. Hassannejad, and A. Sharifnezhad, "A new method of diagnosing athlete's anterior cruciate ligament health status using surface electromyography and deep convolutional neural network," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 1, pp. 65–76, Jan. 2020, doi: 10.1016/j.bbe.2019.05.009.

[114] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[115] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 5353–5360, doi: 10.1109/CVPR.2015.7299173.

[116] C. W. Helstrom, *Statistical Theory of Signal Detection*, 2nd ed., ser. International Series of Monographs in Electronics and Instrumentation. Oxford, UK: Pergamon Press, 1968, vol. 9.

[117] T. Hinderer *et al.*, "Effects of neutron-star dynamic tides on gravitational waveforms within the effective-one-body approach," *Physical Review Letters*, vol. 116, no. 18, May 2016, Art. no. 181101, doi: 10.1103/PhysRevLett.116.181101.

[118] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[119] F. Hlawatsch and F. Auger, Eds., *Time-Frequency Analysis: Concepts and Methods*, ser. Digital Signal and Image Processing Series. London, UK / Hoboken, NJ, USA: ISTE / John Wiley & Sons, 2013.

[120] F. Hlawatsch and P. Flandrin, "The interference structure of the Wigner distribution and related time-frequency signal representations," in *The Wigner Distribution – Theory and Applications in Signal Processing*, W. Mecklenbräuker and F. Hlawatsch, Eds. Amsterdam, Netherlands: Elsevier, 1997, pp. 59–133.

[121] 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 Processing*, vol. 43, no. 2, pp. 149–168, May 1995, doi: 10.1016/0165-1684(94)00150-X.

[122] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[123] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989, doi: 10.1016/0893-6080(89)90020-8.

[124] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[125] ——, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[126] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 7132–7141, doi: 10.1109/TPAMI.2019.2913372.

[127] Y. Huang *et al.*, "GPipe: Efficient training of giant neural networks using pipeline parallelism," in *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, ser. Advances in Neural Information Processing Systems, vol. 32, Neural Information Processing Systems (NIPS). Vancouver, BC, Canada: Curran Associates, Inc., Dec. 2019.

[128] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology*, vol. 160, no. 1, pp. 106–154, Jan. 1962, doi: 10.1113/jphysiol.1962.sp006837.

[129] T. Huynh-The, V.-S. Doan, C.-H. Hua, Q.-V. Pham, T.-V. Nguyen, and D.-S. Kim, "Accurate LPI radar waveform recognition with CWD-TFA for deep convolutional network," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1638–1642, Aug. 2021, doi: 10.1109/LWC.2021.3075880.

[130] A. Iess, E. Cuoco, F. Morawski, and J. Powell, "Core-collapse supernova gravitational-wave search and deep learning classification," *Machine Learning: Science and Technology*, vol. 1, no. 2, May 2020, Art. no. 025014, doi: 10.1088/2632-2153/ab7d31.

[131] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *32nd International Conference on Machine Learning (ICML 2015)*, ser. Proceedings of Machine Learning Research, vol. 37, International Machine Learning Society (IMLS). Lille, France: PMLR, Jul. 2015, pp. 448–456.

[132] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *2009 IEEE 12th International Conference on Computer Vision (ICCV)*. Kyoto, Japan: IEEE, Sep. 2009, pp. 2146–2153, doi: 10.1109/ICCV.2009.5459469.

[133] Jeffrey Kissel (LIGO Scientific Collaboration and Virgo Collaboration), "LIGO Document G1801950-v1: H1 Calibrated Sensitivity Spectra Jun 10 2017 (Representative Best of O2 – C02, With Cleaning/Subtraction)," Nov. 2018. [Online]. Available: https://dcc.ligo.org/LIGO-G1801950/public

[134] ——, "LIGO Document G1801952-v1: L1 Calibrated Sensitivity Spectra Aug 06 2017 (Representative Best of O2 – C02, With Cleaning/Subtraction)," Dec. 2018. [Online]. Available: https://dcc.ligo.org/LIGO-G1801952/public

[135] J. Jeong and W. J. Williams, "Kernel design for reduced interference distributions," *IEEE Transactions on Signal Processing*, vol. 40, no. 2, pp. 402–412, Feb. 1992, doi: 10.1109/78.124950.

[136] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.

[137] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Networks*, vol. 116, pp. 237–245, Aug. 2019, doi: 10.1016/j.neunet.2019.04.014.

[138] V. Katkovnik, "A new method for varying adaptive bandwidth selection," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2567–2571, Sep. 1999, doi: 10.1109/78.782208.

[139] V. Katkovnik, K. Egiazarian, and J. Astola, *Local Approximation Techniques in Signal and Image Processing*. Bellingham, WA, USA: SPIE Press, 2006, doi: 10.1117/3.660178.

[140] S. K. Khare and V. Bajaj, "Time-frequency representation and convolutional neural network-based emotion recognition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 2901–2909, Jul. 2021, doi: 10.1109/TNNLS.2020.3008938.

[141] S. K. Khare, V. Bajaj, and U. R. Acharya, "PDCNNet: An automatic framework for the detection of Parkinson's disease using EEG signals," *IEEE Sensors Journal*, vol. 21, no. 15, pp. 17 017–17 024, Aug. 2021, doi: 10.1109/JSEN.2021.3080135.

[142] ——, "SPWVD-CNN for automated detection of schizophrenia patients using EEG signals," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, Apr. 2021, Art. no. 2507409, doi: 10.1109/TIM.2021.3070608.

[143] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, May 2015.

[144] P. G. Krastev, "Real-time detection of gravitational waves from binary neutron stars using artificial neural networks," *Physics Letters B*, vol. 803, Apr. 2020, Art. no. 135330, doi: 10.1016/j.physletb.2020.135330.

[145] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *26th Conference on Neural Information Processing Systems (NIPS 2012)*, ser. Advances in Neural Information Processing Systems, vol. 25, Neural Information Processing Systems (NIPS). Stateline, NV, USA: Curran Associates, Inc., Dec. 2012.

[146] P. Kwee *et al.*, "Stabilized high-power laser system for the gravitational wave detector advanced LIGO," *Optics Express*, vol. 20, no. 10, pp. 10 617–10 634, Apr. 2012, doi: 10.1364/OE.20.010617.

[147] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," in *23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, Association for the Advancement of Artificial Intelligence (AAAI). Chicago. IL, USA: AAAI Press, Jul. 2008, pp. 646–651.

[148] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, doi: 10.1109/5.726791.

[149] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, 2nd ed., M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 2003, pp. 276–279.

[150] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[151] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*, 2nd ed., ser. Lecture Notes in Computer Science, G. Montavon, G. B. Orr, and K.-R. Müller, Eds. Berlin, Germany: Springer, 2012, vol. 7700, pp. 9–48.

[152] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," in *3rd Conference on Neural Information Processing Systems (NIPS 1989)*, ser. Advances in Neural Information Processing Systems, vol. 2, Neural Information Processing Systems (NIPS). Denver, CO, USA: Morgan-Kaufmann, Nov. 1989, pp. 396–404.

[153] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *2010 IEEE International Symposium on Circuits and Systems (ISCAS 2010)*. Paris, France: IEEE, May 2010, pp. 253–256, doi: 10.1109/ISCAS.2010.5537907.

[154] J. Lerga, M. Vrankic, and V. Sucic, "A signal denoising method based on the improved ICI rule," *IEEE Signal Processing Letters*, vol. 15, pp. 601–604, Oct. 2008, doi: 10.1109/LSP.2008.2001817.

[155] R. Li, C. Song, Y. Song, X. Hao, S. Yang, and X. Song, "Deep geometric convolutional network for automatic modulation classification," *Signal, Image and Video Processing*, vol. 14, pp. 1199–1205, Sep. 2020, doi: 10.1007/s11760-020-01641-3.

[156] LIGO Caltech, "LIGO Technology," LIGO: Laser Interferometer Gravitational-Wave Observatory, (accessed Aug. 7, 2021). [Online]. Available: https://www.ligo.caltech.edu/page/ligo-technology

[157] ——, "LIGO's Interferometer," LIGO: Laser Interferometer Gravitational-Wave Observatory, (accessed Aug. 7, 2021). [Online]. Available: https://www.ligo.caltech.edu/page/ligos-ifo

[158] ——, "Sources and Types of Gravitational Waves," LIGO: Laser Interferometer Gravitational-Wave Observatory, (accessed Aug. 15, 2021). [Online]. Available: https://www.ligo.caltech.edu/page/gw-sources

[159] ——, "What are Gravitational Waves?" LIGO: Laser Interferometer Gravitational-Wave Observatory, (accessed Aug. 15, 2021). [Online]. Available: https://www.ligo.caltech.edu/page/what-are-gw

[160] ——, "LIGO Hanford," LIGO: Laser Interferometer Gravitational-Wave Observatory, 2008, (accessed Nov. 7, 2021). [Online]. Available: https://www.ligo.caltech.edu/image/ligo20150731f

[161] ——, "LIGO Livingston," LIGO: Laser Interferometer Gravitational-Wave Observatory, 2015, (accessed Nov. 7, 2021). [Online]. Available: https://www.ligo.caltech.edu/image/ligo20150731c

[162] ——, "Virgo detector (aerial photo)," LIGO: Laser Interferometer Gravitational-Wave Observatory, 2017, (accessed Nov. 7, 2021). [Online]. Available: https://www.ligo.caltech.edu/image/ligo20170927b

[163] LIGO Scientific Collaboration, "LIGO Algorithm Library - LALSuite," free software, 2018. [Online]. Available: https://doi.org/10.7935/GT1W-FZ16

[164] ——, "The O2 Data Release," Feb. 2019. [Online]. Available: https://doi.org/10.7935/CA75-FM95

[165] LIGO Scientific Collaboration and Virgo Collaboration, "LIGO Document P1800374-v1: GWTC-1: Fig. 1," Dec. 2018. [Online]. Available: https://dcc.ligo.org/LIGO-P1800374/public

[166] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*. Honolulu, HI, USA: IEEE, Jul. 2017, pp. 2117–2125, doi: 10.1109/CVPR.2017.106.

[167] Y.-C. Lin and J.-H. P. Wu, "Detection of gravitational waves using Bayesian neural networks," *Physical Review D*, vol. 103, no. 6, Mar. 2021, Art. no. 063034, doi: 10.1103/PhysRevD.103.063034.

[168] L. Liu and X. Li, "Radar signal recognition based on triplet convolutional neural network," *EURASIP Journal on Advances in Signal Processing*, vol. 2021, Nov. 2021, Art. no. 112, doi: 10.1186/s13634-021-00821-8.

[169] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *33rd International Conference on Machine Learning (ICML 2016)*, ser. Proceedings of Machine Learning Research, vol. 48, International Machine Learning Society (IMLS). New York, NY, USA: PMLR, Jun. 2016, pp. 507–516.

[170] Z. Liu, Y. Shi, Y. Zeng, and Y. Gong, "Radar emitter signal detection with convolutional neural network," in *2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT 2019)*. Jinan, China: IEEE, Oct. 2019, pp. 48–51, doi: 10.1109/ICAIT.2019.8935926.

[171] Z. Liu, Q. Zhang, Z. Han, and G. Chen, "A new classification method for transient power quality combining spectral kurtosis with neural network," *Neurocomputing*, vol. 125, pp. 95–101, Feb. 2014, doi: 10.1016/j.neucom.2012.09.037.

[172] N. Lopac, F. Hržić, I. Petrijevčanin Vuksanović, and J. Lerga, "Detection of non-stationary GW signals in high noise from Cohen's class of time-frequency representations using deep learning," *IEEE Access*, vol. 10, pp. 2408–2428, Jan. 2022, doi: 10.1109/ACCESS.2021.3139850.

[173] N. Lopac, I. Jurdana, J. Lerga, and N. Wakabayashi, "Particle-swarm-optimization-enhanced radial-basis-function-kernel-based adaptive filtering applied to maritime data," *Journal of Marine Science and Engineering*, vol. 9, no. 4, Apr. 2021, Art. no. 439, doi: 10.3390/jmse9040439.

[174] 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, Dec. 2020, Art. no. 6920, doi: 10.3390/s20236920.

[175] N. Lopac, J. Lerga, and I. Jurdana, "On evolutionary metaheuristic optimization approaches in data-driven signal processing techniques," in *5th My First Conference (MFC 2021)*. Rijeka, Croatia: University of Rijeka, Faculty of Maritime Studies, Sep. 2021, p. 27.

[176] N. Lopac, J. Lerga, N. Saulig, L. Stanković, and M. Daković, "On optimal parameters for ICI-based adaptive filtering applied to the GWs in high noise," in *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech*

2021), University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture. Bol and Split, Croatia: IEEE, Sep. 2021, pp. 1–6, doi: 10.23919/SpliTech52315.2021.9566364.

[177] R. Lynch, S. Vitale, R. Essick, E. Katsavounidis, and F. Robinet, "Information-theoretic approach to the gravitational-wave burst detection problem," *Physical Review D*, vol. 95, no. 10, May 2017, Art. no. 104046, doi: 10.1103/PhysRevD.95.104046.

[178] M. Maggiore, *Gravitational Waves: Volume 1: Theory and Experiments*, 1st ed. Oxford, UK: Oxford University Press, 2008.

[179] ——, *Gravitational Waves: Volume 2: Astrophysics and Cosmology*, 1st ed. Oxford, UK: Oxford University Press, 2018.

[180] V. Man'ko and R. Vilela Mendes, "Non-commutative time-frequency tomography," *Physics Letters A*, vol. 263, no. 1, pp. 53–61, Nov. 1999, doi: 10.1016/S0375-9601(99)00688-X.

[181] D. V. Martynov *et al.*, "Sensitivity of the Advanced LIGO detectors at the beginning of gravitational wave astronomy," *Physical Review D*, vol. 93, Jun. 2016, Art. no. 112004, doi: 10.1103/PhysRevD.93.112004.

[182] F. Matichard *et al.*, "Seismic isolation of Advanced LIGO: Review of strategy, instrumentation and performance," *Classical and Quantum Gravity*, vol. 32, no. 18, Aug. 2015, Art. no. 185003, doi: 10.1088/0264-9381/32/18/185003.

[183] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, Jun. 1947, doi: 10.1007/BF02295996.

[184] 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," *Physical Review D*, vol. 103, no. 6, Mar. 2021, Art. no. 062004, doi: 10.1103/PhysRevD.103.062004.

[185] C. Messick *et al.*, "Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data," *Physical Review D*, vol. 95, no. 4, Feb. 2017, Art. no. 042001, doi: 10.1103/PhysRevD.95.042001.

[186] A. L. Miller *et al.*, "How effective is machine learning to detect long transient gravitational waves from neutron stars in a real search?" *Physical Review D*, vol. 100, no. 6, Sep. 2019, Art. no. 062005, doi: 10.1103/PhysRevD.100.062005.

[187] M. Minsky and S. A. Papert, *Perceptrons: An Introduction to Computational Geometry*, 2nd ed. Cambridge, MA, USA: MIT Press, 1988.

[188] T. M. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.

[189] A. Mjahad, A. Rosado-Muñoz, M. Bataller-Mompeán, J. Francés-Víllora, and J. Guerrero-Martínez, "Ventricular fibrillation and tachycardia detection from surface ECG using time-frequency representation images as input dataset for machine learning," *Computer Methods and Programs in Biomedicine*, vol. 141, pp. 119–127, Apr. 2017, doi: 10.1016/j.cmpb.2017.02.010.

[190] M. D. Morales, J. M. Antelis, C. Moreno, and A. I. Nesterov, "Deep learning for gravitational-wave data analysis: A resampling white-box approach," *Sensors*, vol. 21, no. 9, May 2021, Art. no. 3174, doi: 10.3390/s21093174.

[191] J. Morlet, G. Arens, E. Fourgeau, and D. Glard, "Wave propagation and sampling theory – Part I: Complex signal and scattering in multilayered media," *Geophysics*, vol. 47, no. 2, pp. 203–221, Feb. 1982, doi: 10.1190/1.1441328.

[192] N. Mukund, S. Abraham, S. Kandhasamy, S. Mitra, and N. S. Philip, "Transient classification in LIGO data using difference boosting neural network," *Physical Review D*, vol. 95, no. 10, May 2017, Art. no. 104059, doi: 10.1103/PhysRevD.95.104059.

[193] M. Musselman and D. Djurdjanovic, "Time-frequency distributions in the classification of epilepsy from EEG signals," *Expert Systems with Applications*, vol. 39, no. 13, pp. 11 413–11 422, Oct. 2012, doi: 10.1016/j.eswa.2012.04.023.

[194] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *27th International Conference on Machine Learning (ICML 2010)*, International Machine Learning Society (IMLS). Haifa, Israel: Omnipress, Jun. 2010, pp. 807–814.

[195] H. Nakano *et al.*, "Comparison of various methods to extract ringdown frequency from gravitational wave data," *Physical Review D*, vol. 99, no. 12, Jun. 2019, Art. no. 124032, doi: 10.1103/PhysRevD.99.124032.

[196] R. M. Neal, *Bayesian Learning for Neural Networks*, 1st ed., ser. Lecture Notes in Statistics. New York, NY, USA: Springer Science & Business Media, 1996, vol. 118.

[197] M. A. Nielsen, *Neural Networks and Deep Learning*. San Francisco, CA, USA: Determination Press, 2015, vol. 25.

[198] A. Nitz *et al.*, "gwastro/pycbc: PyCBC Release v1.13.6," Apr. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.2643618

[199] A. H. Nitz, T. Dal Canton, D. Davis, and S. Reyes, "Rapid detection of gravitational waves from compact binary mergers with PyCBC Live," *Physical Review D*, vol. 98, Jul. 2018, Art. no. 024050, doi: 10.1103/PhysRevD.98.024050.

[200] A. H. Nitz, T. Dent, T. D. Canton, S. Fairhurst, and D. A. Brown, "Detecting binary compact-object mergers with gravitational waves: Understanding and improving the sensitivity of the PyCBC search," *The Astrophysical Journal*, vol. 849, no. 2, Nov. 2017, Art. no. 118, doi: 10.3847/1538-4357/aa8f50.

[201] S. Oh and R. Marks, "Some properties of the generalized time frequency representation with cone-shaped kernel," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1735–1745, Jul. 1992, doi: 10.1109/78.143445.

[202] R. Ormiston, T. Nguyen, M. Coughlin, R. X. Adhikari, and E. Katsavounidis, "Noise reduction in gravitational-wave data via deep learning," *Physical Review Research*, vol. 2, no. 3, Jul. 2020, Art. no. 033066, doi: 10.1103/PhysRevResearch.2.033066.

[203] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb. 2018, doi: 10.1109/JSTSP.2018.2797022.

[204] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *23rd Conference on Neural Information Processing Systems (NIPS 2009)*, ser. Advances in Neural Information Processing Systems, vol. 22, Neural Information Processing Systems (NIPS). Vancouver, BC, Canada: Curran Associates, Inc., Dec. 2009.

[205] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, doi: 10.1109/TKDE.2009.191.

[206] A. Papandreou and G. F. Boudreaux-Bertels, "Generalization of the Choi-Williams distribution and the Butterworth distribution for time-frequency analysis," *IEEE Transactions on Signal Processing*, vol. 41, no. 1, pp. 463–472, Jan. 1993, doi: 10.1109/TSP.1993.193179.

[207] A. Papandreou-Suppappola, Ed., *Applications in Time-Frequency Signal Processing*, 1st ed., ser. The Electrical Engineering and Applied Signal Processing Series. Boca Raton, FL, USA: CRC Press, 2003.

[208] J. Powell *et al.*, "Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data," *Classical and Quantum Gravity*, vol. 34, no. 3, Jan. 2017, Art. no. 034002, doi: 10.1088/1361-6382/34/3/034002.

[209] J. Powell, D. Trifirò, E. Cuoco, I. S. Heng, and M. Cavaglià, "Classification methods for noise transients in advanced gravitational-wave detectors," *Classical and Quantum Gravity*, vol. 32, no. 21, Oct. 2015, Art. no. 215012, doi: 10.1088/0264-9381/32/21/215012.

[210] W. Qinghua, Z. Youyun, C. Lei, and Z. Yongsheng, "Fault diagnosis for diesel valve trains based on non-negative matrix factorization and neural network ensemble," *Mechanical Systems and Signal Processing*, vol. 23, no. 5, pp. 1683–1695, Jul. 2009, doi: 10.1016/j.ymssp.2008.12.004.

[211] Z. Qu, C. Hou, C. Hou, and W. Wang, "Radar signal intra-pulse modulation recognition based on convolutional neural network and deep Q-learning network," *IEEE Access*, vol. 8, pp. 49 125–49 136, Mar. 2020, doi: 10.1109/ACCESS.2020.2980363.

[212] Z. Qu, X. Mao, and Z. Deng, "Radar signal intra-pulse modulation recognition based on convolutional neural network," *IEEE Access*, vol. 6, pp. 43 874–43 884, Aug. 2018, doi: 10.1109/ACCESS.2018.2864347.

[213] Z. Qu, W. Wang, C. Hou, and C. Hou, "Radar signal intra-pulse modulation recognition based on convolutional denoising autoencoder and deep convolutional neural network," *IEEE Access*, vol. 7, pp. 112 339–112 347, Aug. 2019, doi: 10.1109/ACCESS.2019.2935247.

[214] R. Abbott et al. (LIGO Scientific Collaboration and Virgo Collaboration), "Open data from the first and second observing runs of Advanced LIGO and Advanced Virgo," *SoftwareX*, vol. 13, Jan. 2021, Art. no. 100658, doi: 10.1016/j.softx.2021.100658.

[215] M. Razzano and E. Cuoco, "Image-based deep learning for classification of noise transients in gravitational wave detectors," *Classical and Quantum Gravity*, vol. 35, no. 9, Apr. 2018, Art. no. 095016, doi: 10.1088/1361-6382/aab793.

[216] A. Rebei *et al.*, "Fusing numerical relativity and deep learning to detect higher-order multipole waveforms from eccentric binary black hole mergers," *Physical Review D*, vol. 100, no. 4, Aug. 2019, Art. no. 044025, doi: 10.1103/PhysRevD.100.044025.

[217] B. Richhariya and M. Tanveer, "EEG signal classification using universum support vector machine," *Expert Systems with Applications*, vol. 106, pp. 169–182, Sep. 2018, doi: 10.1016/j.eswa.2018.03.053.

[218] K. Riles, "Recent searches for continuous gravitational waves," *Modern Physics Letters A*, vol. 32, no. 39, Dec. 2017, Art. no. 1730035, doi: 10.1142/S021773231730035X.

[219] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.

[220] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[221] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992, doi: 10.1016/0167-2789(92)90242-F.

[222] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[223] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[224] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., ser. Pearson Series in Artificial Intelligence. Hoboken, NJ, USA: Pearson, 2020.

[225] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*. South Brisbane, QLD, Australia: IEEE, Apr. 2015, pp. 4580–4584, doi: 10.1109/ICASSP.2015.7178838.

[226] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[227] B. S. Sathyaprakash and B. F. Schutz, "Physics, astrophysics and cosmology with gravitational waves," *Living Reviews in Relativity*, vol. 12, no. 1, Dec. 2009, Art. no. 2, doi: 10.12942/lrr-2009-2.

[228] M. B. Schäfer, F. Ohme, and A. H. Nitz, "Detection of gravitational-wave signals from binary neutron star mergers using machine learning," *Physical Review D*, vol. 102, no. 6, Sep. 2020, Art. no. 063015, doi: 10.1103/PhysRevD.102.063015.

[229] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.

[230] E. Sejdic, I. Djurovic, and J. Jiang, "Time-frequency feature representation using energy concentration: An overview of recent advances," *Digital Signal Processing*, vol. 19, no. 1, pp. 153–183, Jan. 2009, doi: 10.1016/j.dsp.2007.12.004.

[231] H. Shen, D. George, E. A. Huerta, and Z. Zhao, "Denoising gravitational waves with enhanced deep recurrent denoising auto-encoders," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*. Brighton, UK: IEEE, May 2019, pp. 3237–3241, doi: 10.1109/ICASSP.2019.8683061.

[232] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, Jul. 2019, Art. no. 60, doi: 10.1186/s40537-019-0197-0.

[233] W. Si, C. Wan, and C. Zhang, "Towards an accurate radar waveform recognition algorithm based on dense CNN," *Multimedia Tools and Applications*, vol. 80, no. 2, pp. 1779–1792, Jan. 2021, doi: 10.1007/s11042-020-09490-5.

[234] L. Sifre, "Rigid-motion scattering for image classification," phdthesis, Ecole Polytechnique, CMAP, Paris, France, Oct. 2014.

[235] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, May 2015.

[236] R. J. E. Smith, I. Mandel, and A. Vecchio, "Studies of waveform requirements for intermediate mass-ratio coalescence searches with advanced gravitational-wave detectors," *Physical Review D*, vol. 88, no. 4, Aug. 2013, Art. no. 044010, doi: 10.1103/PhysRevD.88.044010.

[237] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *26th Conference on Neural Information Processing Systems (NIPS 2012)*, ser. Advances in Neural Information Processing Systems, vol. 25, Neural Information Processing Systems (NIPS). Stateline, NV, USA: Curran Associates, Inc., Dec. 2012.

[238] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[239] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *arXiv preprint arXiv:1505.00387*, 2015.

[240] L. Stankovic, M. Daković, and T. Thayaparan, *Time-Frequency Signal Analysis with Applications*. Boston, MA, USA: Artech House, 2013.

[241] W. Staszewski, K. Worden, and G. Tomlinson, "Time-frequency analysis in gearbox fault detection using the Wigner-Ville distribution and pattern recognition," *Mechanical Systems and Signal Processing*, vol. 11, no. 5, pp. 673–692, Sep. 1997, doi: 10.1006/mssp.1997.0102.

[242] C. Szegedy *et al.*, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations (ICLR 2014)*, Banff, AB, Canada, Apr. 2014.

[243] ——, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, MA, USA: IEEE, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[244] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence (AAAI-17)*. San Francisco, CA, USA: Association for the Advancement of Artificial Intelligence (AAAI), Feb. 2017, pp. 4278–4284.

[245] M. Tan *et al.*, "MnasNet: Platform-aware neural architecture search for mobile," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2019)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 2820–2828, doi: 10.1109/CVPR.2019.00293.

[246] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *36th International Conference on Machine Learning (ICML 2019)*, ser.

Proceedings of Machine Learning Research, vol. 97. Long Beach, CA, USA: PMLR, May 2019, pp. 6105–6114.

[247] L. Tang, Y. Jia, Y. Qian, S. Yi, and P. Yuan, "Human activity recognition based on mixed CNN with radar multi-spectrogram," *IEEE Sensors Journal*, vol. 21, no. 22, pp. 25 950–25 962, Oct. 2021, doi: 10.1109/JSEN.2021.3118836.

[248] J. E. Thompson *et al.*, "Modeling the gravitational wave signature of neutron star black hole coalescences," *Physical Review D*, vol. 101, no. 12, Jun. 2020, Art. no. 124059, doi: 10.1103/PhysRevD.101.124059.

[249] K. S. Thorne, "Gravitational radiation," in *Three Hundred Years of Gravitation*, 1st ed., S. W. Hawking and W. Israel, Eds. Cambridge, UK: Cambridge University Press, 1987, ch. 9, pp. 330–458.

[250] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSProp: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, 2012.

[251] D. C. Toledo-Pérez, J. Rodríguez-Reséndiz, R. A. Gómez-Loenzo, and J. Jauregui-Correa, "Support vector machine-based EMG signal classification techniques: A review," *Applied Sciences*, vol. 9, no. 20, Oct. 2019, Art. no. 4402, doi: 10.3390/app9204402.

[252] A. Torres-Forné, E. Cuoco, J. A. Font, and A. Marquina, "Application of dictionary learning to denoise LIGO's blip noise transients," *Physical Review D*, vol. 102, no. 2, Jul. 2020, Art. no. 023011, doi: 10.1103/PhysRevD.102.023011.

[253] 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," *Physical Review D*, vol. 98, no. 8, Oct. 2018, Art. no. 084013, doi: 10.1103/PhysRevD.98.084013.

[254] A. Torres-Forné, A. Marquina, J. A. Font, and J. M. Ibáñez, "Total-variation-based methods for gravitational wave denoising," *Physical Review D*, vol. 90, no. 8, Oct. 2014, Art. no. 084029, doi: 10.1103/PhysRevD.90.084029.

[255] ——, "Denoising of gravitational wave signals via dictionary learning algorithms," *Physical Review D*, vol. 94, no. 12, Dec. 2016, Art. no. 124040, doi: 10.1103/PhysRevD.94.124040.

[256] M. G. Tsipouras and D. I. Fotiadis, "Automatic arrhythmia detection based on time and time-frequency analysis of heart rate variability," *Computer Methods and Programs in Biomedicine*, vol. 74, no. 2, pp. 95–108, May 2004, doi: 10.1016/S0169-2607(03)00079-8.

[257] J. W. Tukey, "An introduction to the calculations of numerical spectrum analysis," in *Spectral Analysis of Time Series*, B. Harris, Ed. New York, NY, USA: Wiley, 1967, pp. 25–46.

[258] T. Tuncer, S. Dogan, and A. Subasi, "Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition," *Biomedical Signal Processing and Control*, vol. 58, Apr. 2020, Art. no. 101872, doi: 10.1016/j.bspc.2020.101872.

[259] A. T. Tzallas, M. G. Tsipouras, and D. I. Fotiadis, "Automatic seizure detection based on time-frequency analysis and artificial neural networks," *Computational Intelligence and Neuroscience*, vol. 2007, Dec. 2007, Art. no. 080510, doi: 10.1155/2007/80510.

[260] ——, "The use of time-frequency distributions for epileptic seizure detection in EEG recordings," in *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Lyon, France: IEEE, Aug. 2007, pp. 3–6, doi: 10.1109/IEMBS.2007.4352208.

[261] S. A. Usman *et al.*, "The PyCBC search for gravitational waves from compact binary coalescence," *Classical and Quantum Gravity*, vol. 33, no. 21, Oct. 2016, Art. no. 215004, doi: 10.1088/0264-9381/33/21/215004.

[262] G. Vajente *et al.*, "Machine-learning nonstationary noise out of gravitational-wave detectors," *Physical Review D*, vol. 101, no. 4, Feb. 2020, Art. no. 042003, doi: 10.1103/PhysRevD.101.042003.

[263] A. van den Oord *et al.*, "WaveNet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[264] J. Ville, "Theorie et application dela notion de signal analytique," *Câbles et transmissions*, vol. 2, no. 1, pp. 61–74, 1948.

[265] I. Volarić, "Signal concentration enhancement in the time-frequency domain using adaptive compressive sensing," phdthesis, University of Rijeka, Faculty of Engineering, Sep. 2017.

[266] C. Wang, J. Wang, and X. Zhang, "Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*. New Orleans, LA, USA: IEEE, Mar. 2017, pp. 2437–2441, doi: 10.1109/ICASSP.2017.7952594.

[267] H. Wang, S. Wu, Z. Cao, X. Liu, and J.-Y. Zhu, "Gravitational-wave signal recognition of LIGO data by deep learning," *Physical Review D*, vol. 101, no. 10, May 2020, Art. no. 104003, doi: 10.1103/PhysRevD.101.104003.

[268] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters*, vol. 119, pp. 3–11, Mar. 2019, doi: 10.1016/j.patrec.2018.02.010.

[269] T. Wang, J. Liang, and X. Liu, "Soil moisture retrieval algorithm based on TFA and CNN," *IEEE Access*, vol. 7, pp. 597–604, Dec. 2019, doi: 10.1109/ACCESS.2018.2885565.

[270] X. Wang, G. Huang, Z. Zhou, and J. Gao, "Radar emitter recognition based on the short time Fourier transform and convolutional neural networks," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI 2017)*. Shanghai, China: IEEE, Oct. 2017, pp. 1–5, doi: 10.1109/CISP-BMEI.2017.8302111.

[271] W. Wei and E. Huerta, "Gravitational wave denoising of binary black hole mergers with deep learning," *Physics Letters B*, vol. 800, Jan. 2020, Art. no. 135081, doi: 10.1016/j.physletb.2019.135081.

[272] W. Wei, A. Khan, E. Huerta, X. Huang, and M. Tian, "Deep learning ensemble for real-time gravitational wave detection of spinning binary black hole mergers," *Physics Letters B*, vol. 812, Jan. 2021, Art. no. 136029, doi: 10.1016/j.physletb.2020.136029.

[273] E. W. Weisstein, "Bonferroni correction," MathWorld – A Wolfram Web Resource, 2015, (accessed Aug. 6, 2021). [Online]. Available: https://mathworld.wolfram.com/BonferroniCorrection.html

[274] P. Welch, "The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, no. 2, pp. 70–73, Jun. 1967, doi: 10.1109/TAU.1967.1161901.

[275] W. J. Williams and J. Jeong, "Reduced interference time-frequency distributions," in *Time-Frequency Signal Analysis: Methods and Applications*, B. Boashash, Ed. Melbourne/New York: Longman-Cheshire/Wiley, 1992, ch. 3, pp. 74–97.

[276] Z. Wu, T. Lan, C. Yang, and Z. Nie, "A novel method to detect multiple arrhythmias based on time-frequency analysis and convolutional neural networks," *IEEE Access*, vol. 7, pp. 170 820–170 830, Nov. 2019, doi: 10.1109/ACCESS.2019.2956050.

[277] Y. Xia, N. Wulan, K. Wang, and H. Zhang, "Detecting atrial fibrillation by deep convolutional neural networks," *Computers in Biology and Medicine*, vol. 93, pp. 84–92, Feb. 2018, doi: 10.1016/j.compbiomed.2017.12.007.

[278] Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," *Constructive Approximation*, vol. 26, pp. 289–315, Apr. 2007, doi: 10.1007/s00365-006-0663-2.

[279] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *4th International Conference on Learning Representations (ICLR 2016)*, San Juan, Puerto Rico, May 2016.

[280] L. Yuan, "A time-frequency feature fusion algorithm based on neural network for HRRP," *Progress In Electromagnetics Research M*, vol. 55, pp. 63–71, Mar. 2017, doi: 10.2528/PIERM16123002.

[281] L. Yuan and J. Cao, "Patients' EEG data analysis via spectrogram image with a convolution neural network," in *9th KES International Conference on Intelligent Decision Technologies (KES-IDT 2017)*, ser. Smart Innovation, Systems and Technologies, vol. 72. Vilamoura, Portugal: Springer, Jun. 2017, pp. 13–21, doi: 10.1007/978-3-319-59421-7_2.

[282] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *27th British Machine Vision Conference (BMVC 2016)*, British Machine Vision Association (BMVA). York, UK: BMVA Press, Sep. 2016, pp. 87.1–87.12, doi: 10.5244/C.30.87.

[283] M. Zevin *et al.*, "Gravity Spy: integrating Advanced LIGO detector characterization, machine learning, and citizen science," *Classical and Quantum Gravity*, vol. 34, no. 6, Feb. 2017, Art. no. 064003, doi: 10.1088/1361-6382/aa5cea.

[284] M. Zhang, M. Diao, and L. Guo, "Convolutional neural networks for automatic cognitive radio waveform recognition," *IEEE Access*, vol. 5, pp. 11 074–11 082, Jun. 2017, doi: 10.1109/ACCESS.2017.2716191.

[285] M. Zhang, L. Liu, and M. Diao, "LPI radar waveform recognition based on time-frequency distribution," *Sensors*, vol. 16, no. 10, Oct. 2016, Art. no. 1682, doi: 10.3390/s16101682.

[286] Y. Zhao, L. E. Atlas, and R. J. Marks, "The use of cone-shaped kernels for generalized time-frequency representations of nonstationary signals," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1084–1091, Jul. 1990, doi: 10.1109/29.57537.

[287] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.

LIST OF ABBREVIATIONS

AI	artificial intelligence
ANN	artificial neural network
ASD	amplitude spectral density
BBH	binary black hole
BJD	Born-Jordan distribution
BNS	binary neutron star
BUD	Butterworth distribution
CBC	compact binary coalescence
CCSN	core-collapse supernova
CLDNN	convolutional, long short-term memory, fully connected deep neural network
CNN	convolutional neural network
CPU	central processing unit
CWD	Choi-Williams distribution
ECG	electrocardiogram
EEG	electroencephalogram
EGO	European Gravitational Observatory
EMG	electromyogram
EOB	effective-one-body
FLOPS	floating-point operations per second
GPS	Global Positioning System
GPU	graphics processing unit
GW	gravitational wave
GWOSC	Gravitational Wave Open Science Center
GWTC	Gravitational-Wave Transient Catalog
IAF	instantaneous autocorrelation function
ICI	intersection of confidence intervals
ILSVRC	ImageNet Large-Scale Visual Recognition Challenge
KAGRA	Kamioka Gravitational Wave Detector
LFM	linear frequency modulation
LIGO	Laser Interferometer Gravitational-Wave Observatory
LPA	local polynomial approximation
LPI	low-probability-of-intercept
MBTA	Multi-Band Template Analysis

MF-SNR	matched-filter SNR
MIT	Massachusetts Institute of Technology
NMF-SNR	network matched-filter signal-to-noise ratio
NOMF-SNR	network optimal matched-filter signal-to-noise ratio
NSBH	neutron star - black hole
O1	the first observing run of the Advanced LIGO detectors
O2	the second observing run of the Advanced LIGO and Advanced Virgo detectors
O3a	the first part of the third observing run of the Advanced LIGO and Advanced Virgo detectors
O3b	the second part of the third observing run of the Advanced LIGO and Advanced Virgo detectors
OMF-SNR	optimal matched-filter signal-to-noise ratio
PNG	Portable Network Graphics
PSD	power spectral density
PWVD	pseudo Wigner-Ville distribution
ReLU	rectified linear unit
RGB	red, green, blue
RICI	relative intersection of confidence intervals
RIDB	reduced-interference distribution with a kernel based on the first kind Bessel function
RIDBN	reduced-interference distribution with a kernel based on binomial coefficients
RIDH	reduced-interference distribution with a kernel based on the Hanning window
RIDT	reduced-interference distribution with a kernel based on the triangular window
RNN	recurrent neural network
ROC	receiver operating characteristic
ROC AUC	area under the receiver operating characteristic curve
SGD	stochastic gradient descent
SNR	signal-to-noise ratio
SP	spectrogram
SPWVD	smoothed pseudo Wigner-Ville distribution
STFT	short-time Fourier transform
SVM	support vector machine
TFD	time-frequency distribution
TV	total variation
WVD	Wigner-Ville distribution
ZAMD	Zhao-Atlas-Marks distribution

LIST OF SYMBOLS

Latin symbols:

a	output predicted by deep learning model
A	contingency table variable denoting the number of instances in the test dataset that both Model 1 and Model 2 classified correctly
$A_s(\nu, \tau)$	ambiguity function
b	bias of an artificial neuron
B	contingency table variable denoting the number of instances in the test dataset that Model 1 classified correctly and Model 2 incorrectly
$BJD_s(t, f)$	Born-Jordan distribution of $s(t)$
$BUD_s(t, f)$	Butterworth distribution of $s(t)$
c_i	number of channels of input tensor of convolution operation
c_j	number of channels of output tensor of convolution operation
C	contingency table variable denoting the number of instances in the test dataset that Model 1 classified incorrectly and Model 2 correctly
$CWD_s(t, f)$	Choi-Williams distribution of $s(t)$
d	convolutional neural network depth scaling coefficient
d_r	dilation rate of convolution kernel
D	contingency table variable denoting the number of instances in the test dataset that both Model 1 and Model 2 classified incorrectly
E_s	signal energy
f	frequency
f_0	frequency shift
$f_i(t)$	instantaneous frequency of the signal
$\mathcal{F}$	Fourier transform
FN	false negatives
FP	false positives
$g(t)$	time-smoothing window function
$h(t)$	window function
$h_\times(t)$	$\times$ (cross) tensor polarization mode of gravitational wave
$h_+(t)$	$+$ (plus) tensor polarization mode of gravitational wave
$h_{GW}(t)$	gravitational-wave strain amplitude
h_i	height of input tensor of convolution operation
$h_t(t)$	matched-filter template

$\mathcal{H}$	Hilbert transform
$H_t(f)$	Fourier transform of $h_t(t)$
I	data example value being normalized
I_{max}	maximum data example value
I_{min}	minimum data example value
I_n	normalized data example value
J	cost function for training deep learning model
k	width of convolution kernel
K	two-dimensional convolution kernel
$K_s(t,\tau)$	instantaneous autocorrelation function of $s(t)$
L	length of interferometer's arm
m	width of the image used as input to convolution operation
$n(t)$	gravitational-wave detector noise
N_x	total number of data examples for training deep learning model
p	p-value of statistical significance test
$PWVD_s(t,f)$	pseudo Wigner-Ville distribution of $s(t)$
r	convolutional neural network resolution scaling coefficient
$r(t)$	real signal
r_E	reduction ratio of the Squeeze-and-Excitation block
$R(f)$	Fourier transform of $r(t)$
$RIDB_s(t,f)$	reduced-interference distribution with a kernel based on the first kind Bessel function calculated for $s(t)$
$RIDBN_s(t,f)$	reduced-interference distribution with a kernel based on binomial coefficients calculated for $s(t)$
$RIDH_s(t,f)$	reduced-interference distribution with a kernel based on the Hanning window calculated for $s(t)$
$RIDT_s(t,f)$	reduced-interference distribution with a kernel based on the triangular window calculated for $s(t)$
$s(t)$	analytic associate of $r(t)$
$s^*(t)$	complex conjugate of $s(t)$
$s_{fs}(t)$	frequency-shifted signal $s(t)$
$s_{GW}(t)$	strain measured by gravitational-wave detector
$s_{GW,w}(t)$	whitened strain
$s_{ts}(t)$	time-shifted signal $s(t)$
$S(f)$	Fourier transform of $s(t)$
$S_{GW}(f)$	Fourier transform of $s_{GW}(t)$
$S_{GW,w}(f)$	Fourier transform of $s_{GW,w}(t)$
$S_n(f)$	noise power spectral density
$SP_s(t,f)$	spectrogram of $s(t)$
$SPWVD_s(t,f)$	smoothed pseudo Wigner-Ville distribution of $s(t)$
$STFT_s(t,f)$	short-time Fourier transform of $s(t)$
t	time
t_0	time shift
T_i	input tensor of convolution operation

T_j	output tensor of convolution operation
TN	true negatives
TP	true positives
w	convolutional neural network width scaling coefficient
$\mathbf{w}$	vector of real-valued weights of an artificial neuron
w_i	width of input tensor of convolution operation
$WVD_s(t, f)$	Wigner-Ville distribution of $s(t)$
x	input in deep learning model training
$\mathbf{x}$	input vector of an artificial neuron
X	two-dimensional input to convolution operation
y	desired output of deep learning model
Y	two-dimensional output of convolution operation
$z_{MF}(t)$	matched-filter output
$ZAMD_s(t, f)$	Zhao-Atlas-Marks distribution of $s(t)$

Greek symbols:

α	statistical significance level
ϵ	learning rate during training of deep learning models
$\theta(f)$	phase of the Fourier transform of the signal
ν	Doppler variable
$\rho_{MF}(t)$	matched-filter signal-to-noise ratio
$\rho_{NMF}(t)$	network matched-filter signal-to-noise ratio
$\rho_s(t, f)$	time-frequency distribution of $s(t)$
$\rho_{s_{fs}}(t, f)$	time-frequency distribution of $s_{fs}(t)$
$\rho_{s_{ts}}(t, f)$	time-frequency distribution of $s_{ts}(t)$
σ	width of the exponential kernel in the Choi-Williams distribution
σ_{MF}	normalization constant of matched-filter template
τ	lag variable
$\tau_d(f)$	time delay of the signal
$\tau_g(f)$	group delay of the signal
ϕ	compound coefficient
$\phi(t)$	instantaneous phase of the signal
χ^2	McNemar's test statistic

LIST OF FIGURES

LIST OF TABLES

APPENDICES

Appendix A

RESULTS OF THE TFD-CNN MODELS EVALUATION

This appendix presents more detailed evaluation results of the considered deep learning models on the test dataset, including confusion matrices, ROC curves, and precision-recall curves. The results are given for each combination of three considered CNN architectures and 12 TFDs of the input data.

Figure A.1 shows the confusion matrices for each of the TFDs of the input data combined with the ResNet-101 CNN architecture, Figure A.2 gives the confusion matrices obtained by the deep learning models using the TFDs and the Xception architecture, while Figure A.3 provides the confusion matrices for the EfficientNet architecture coupled with each considered TFD of the input data.

In addition, Figures A.4, A.5, and A.6 show the ROC curves for the deep learning models combining each considered TFD of the input data with the ResNet-101, Xception, and EfficientNet CNN architecture, respectively. Finally, the precision-recall curves of the deep learning models using the TFDs as input to the ResNet-101, Xception, and EfficientNet CNN architecture are given in Figures A.7, A.8, and A.9, respectively.

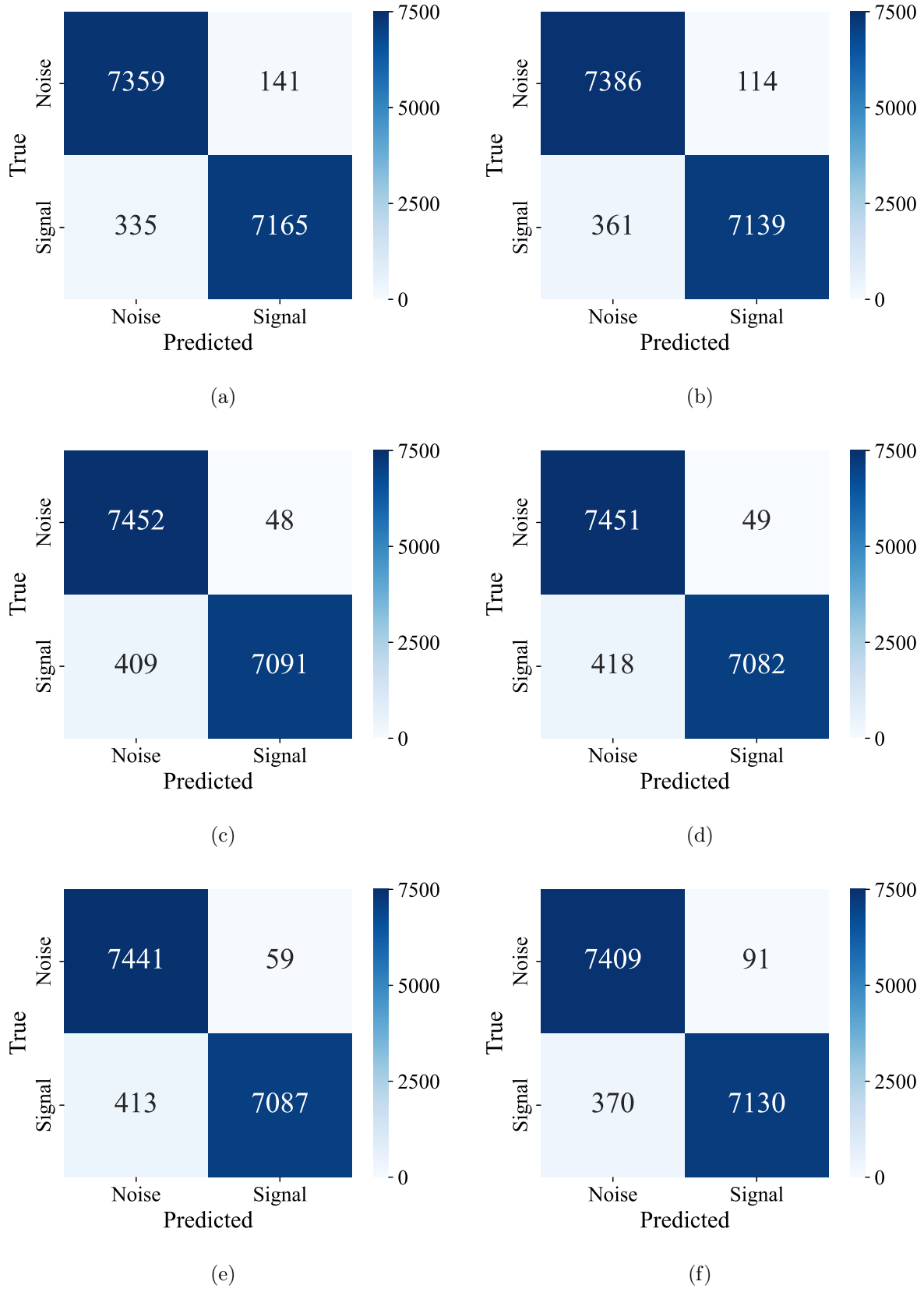
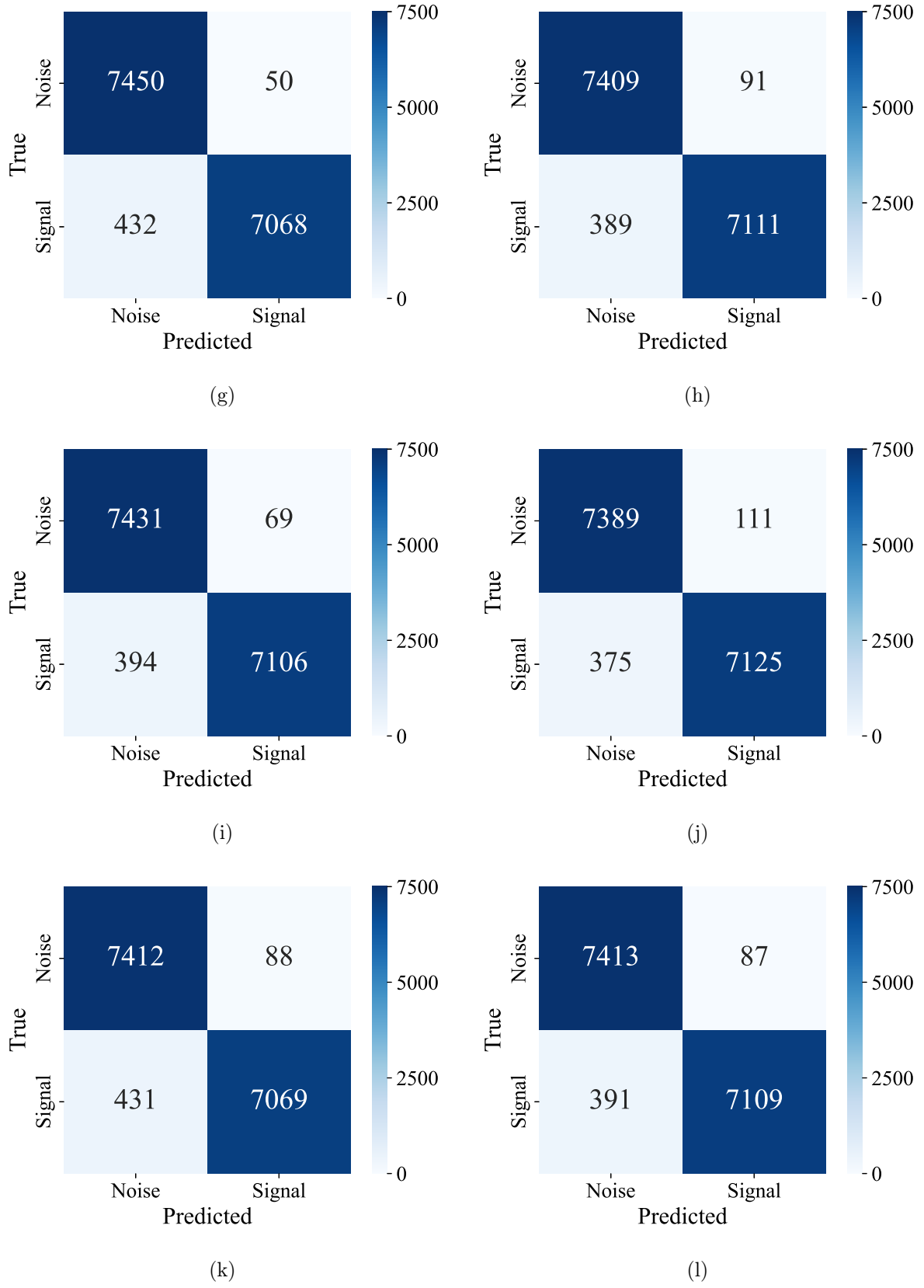


Figure A.1 Confusion matrices for evaluating the deep learning models combining the ResNet-101 CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.



(g) (h)

(i) (j)

(k) (l)

Figure A.1 (cont.) Confusion matrices for evaluating the deep learning models combining the ResNet-101 CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

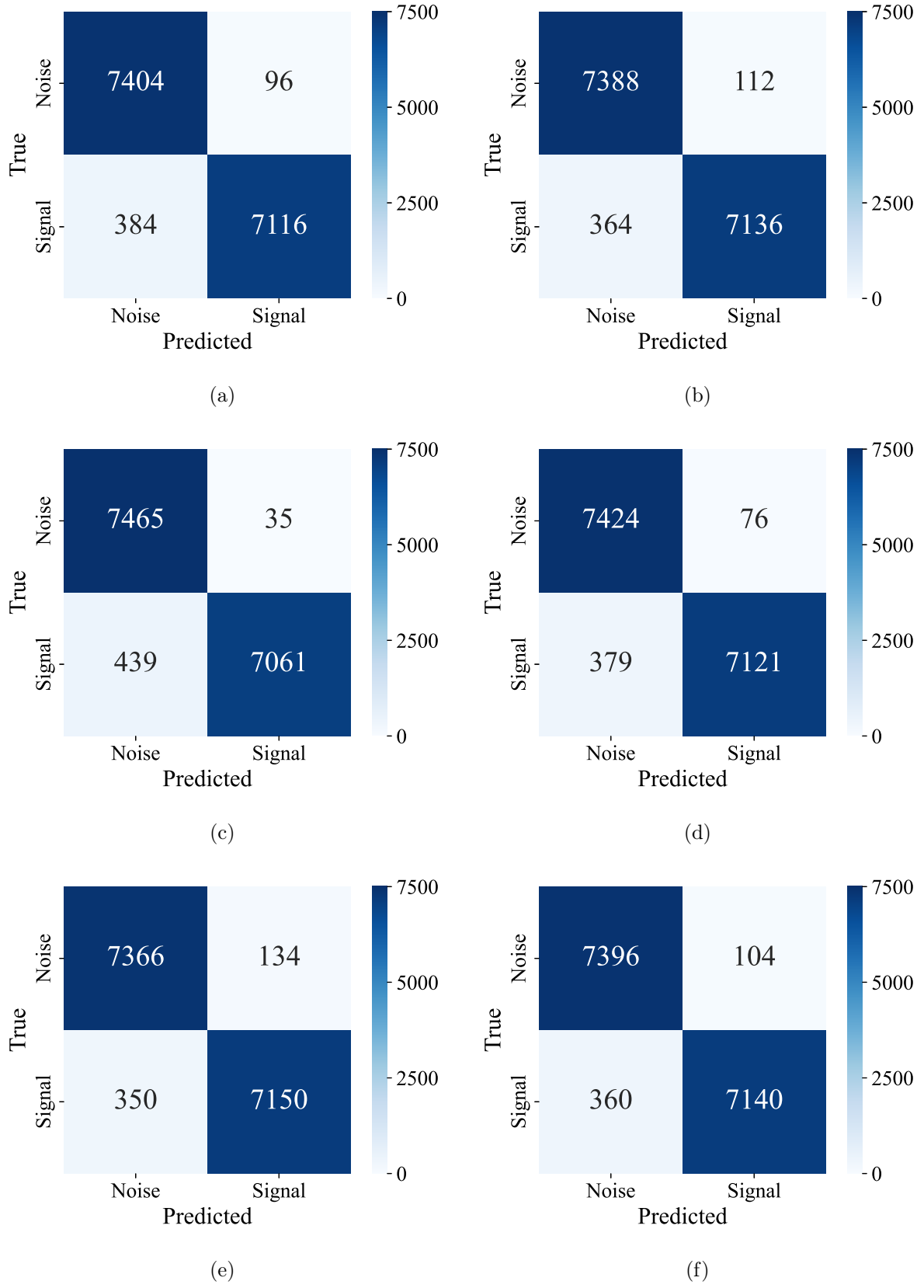
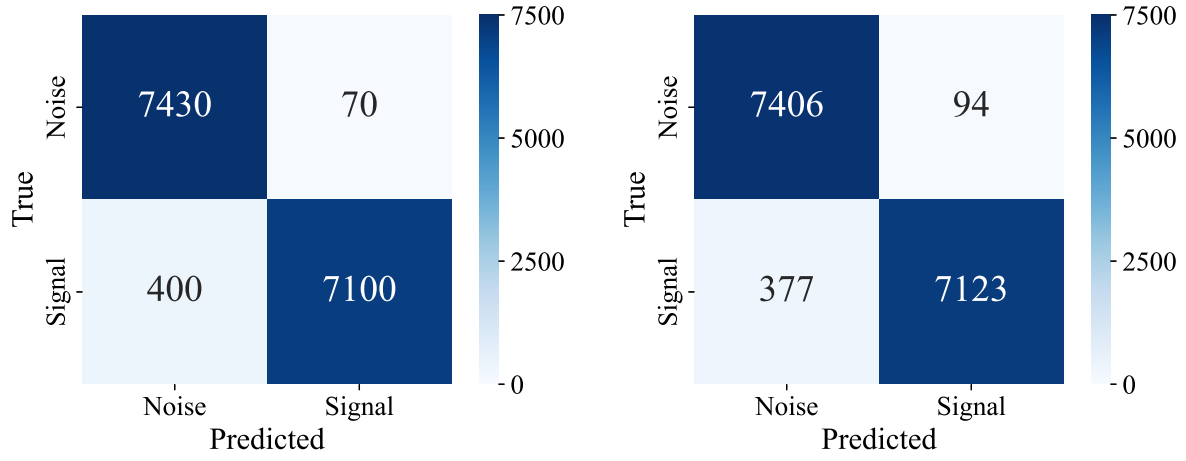
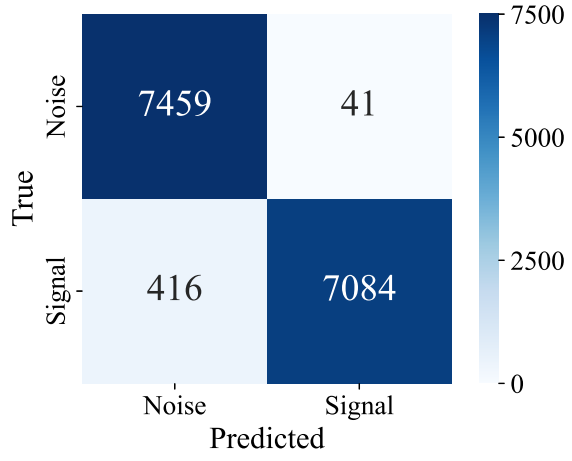


Figure A.2 Confusion matrices for evaluating the deep learning models combining the Xception CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

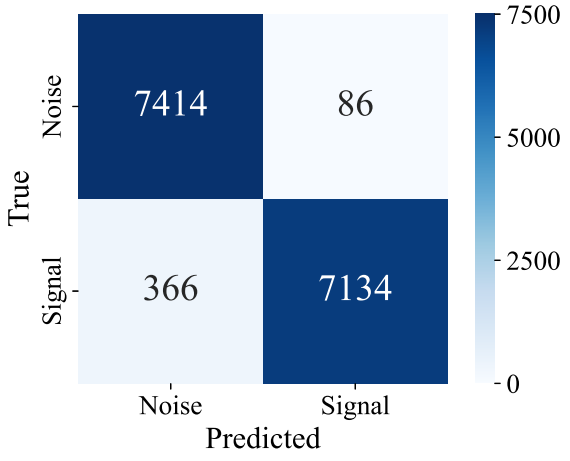


(g)

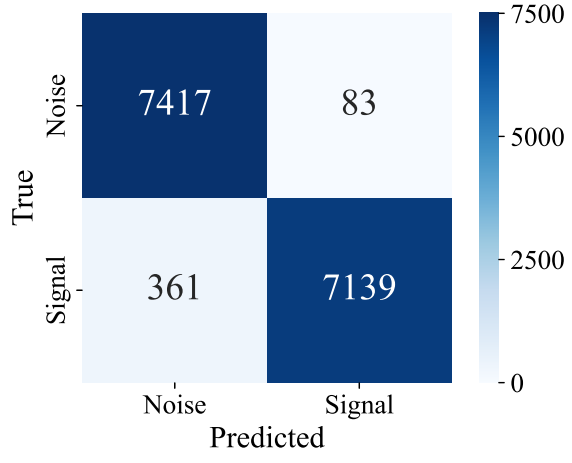
(h)



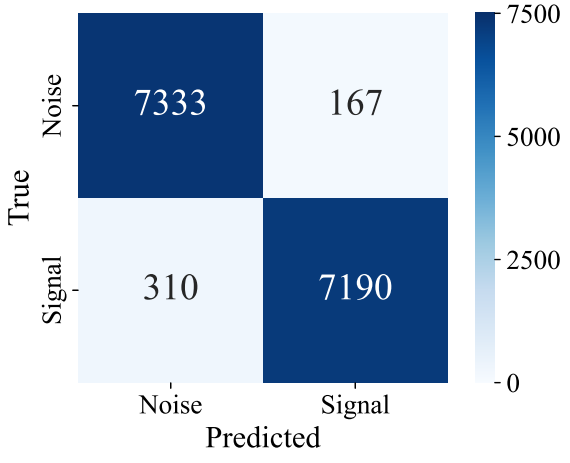
(i)



(j)



(k)



(l)

Figure A.2 (cont.) Confusion matrices for evaluating the deep learning models combining the Xception CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

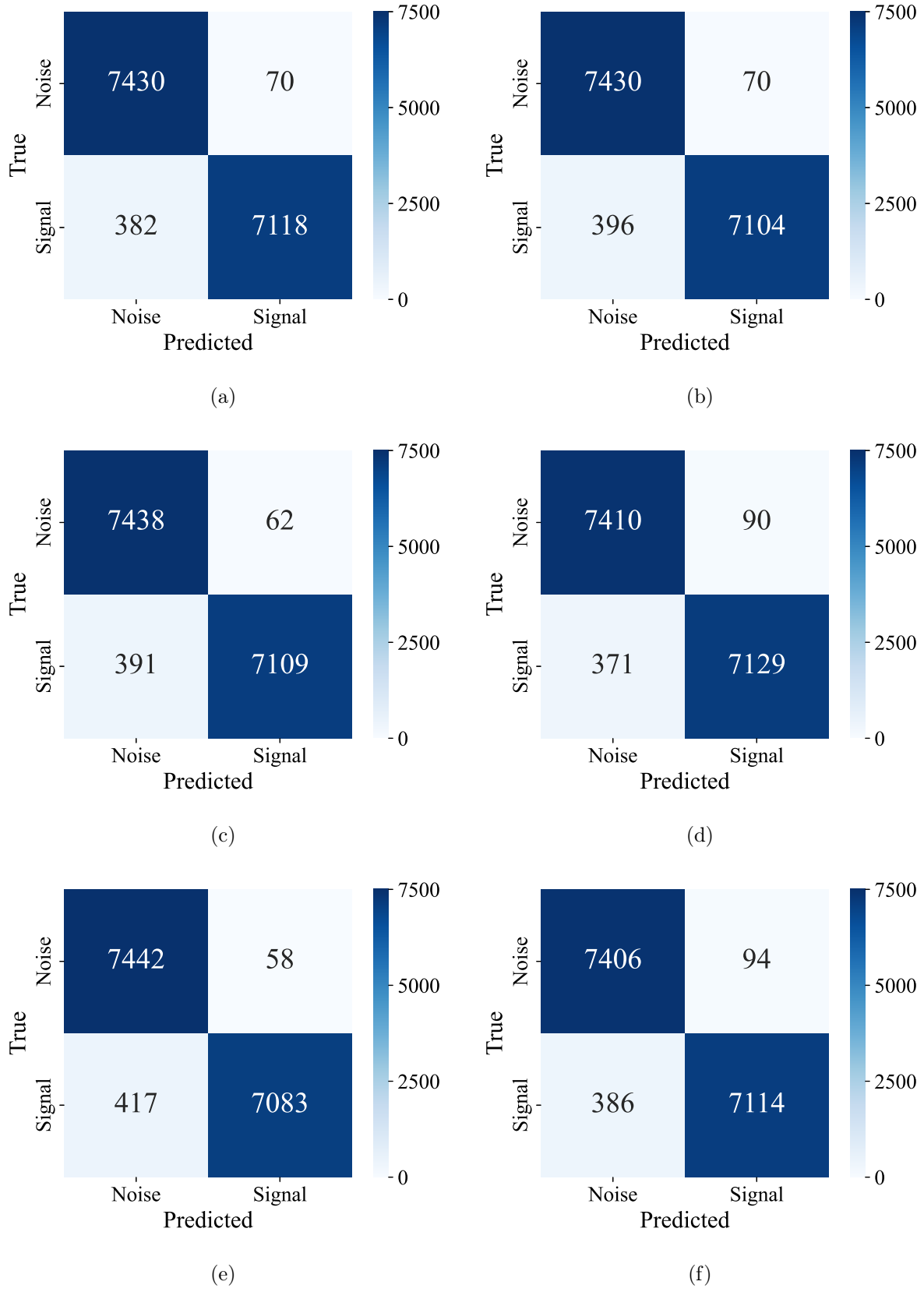


Figure A.3 Confusion matrices for evaluating the deep learning models combining the EfficientNet CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

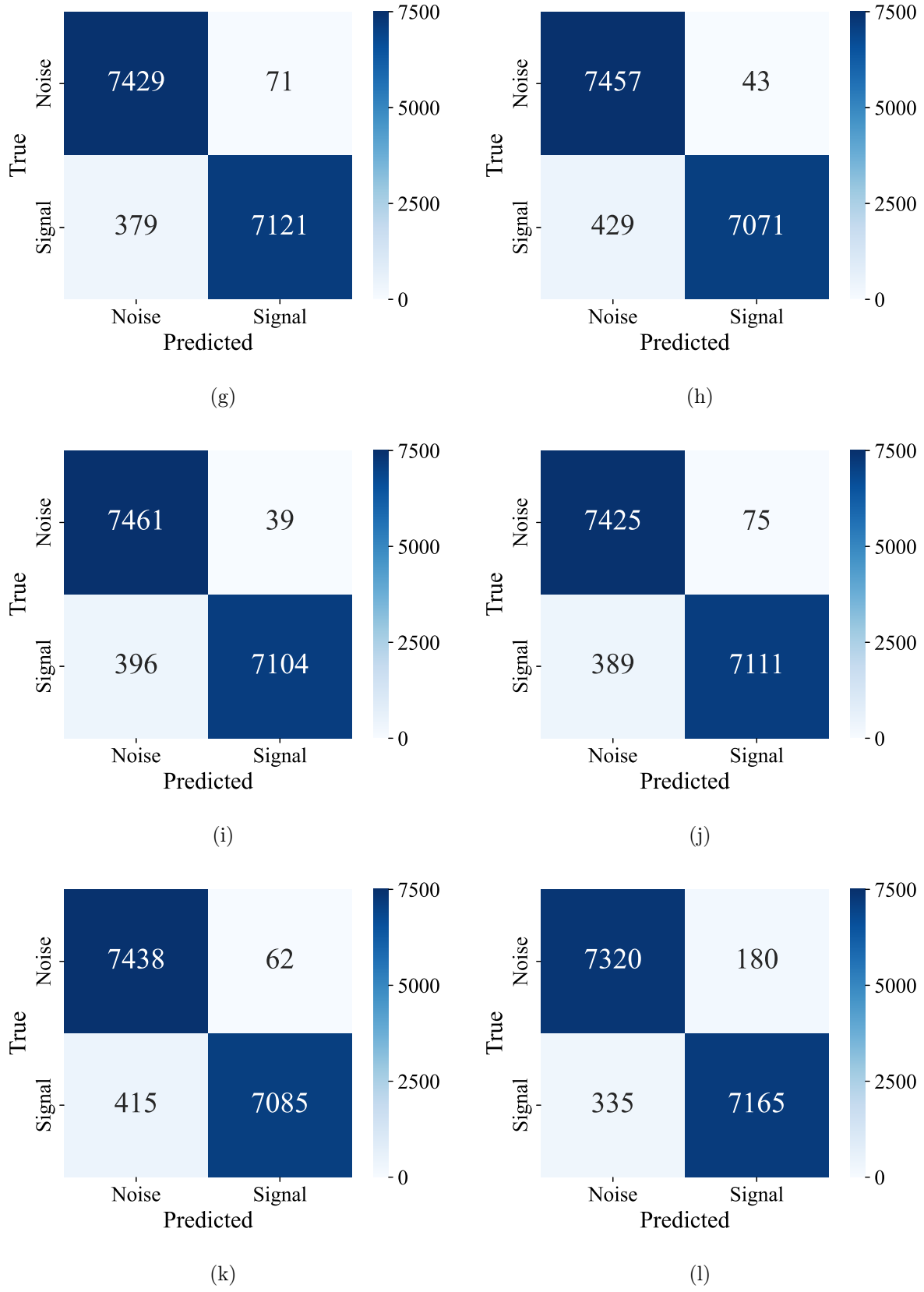


Figure A.3 (cont.) Confusion matrices for evaluating the deep learning models combining the EfficientNet CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

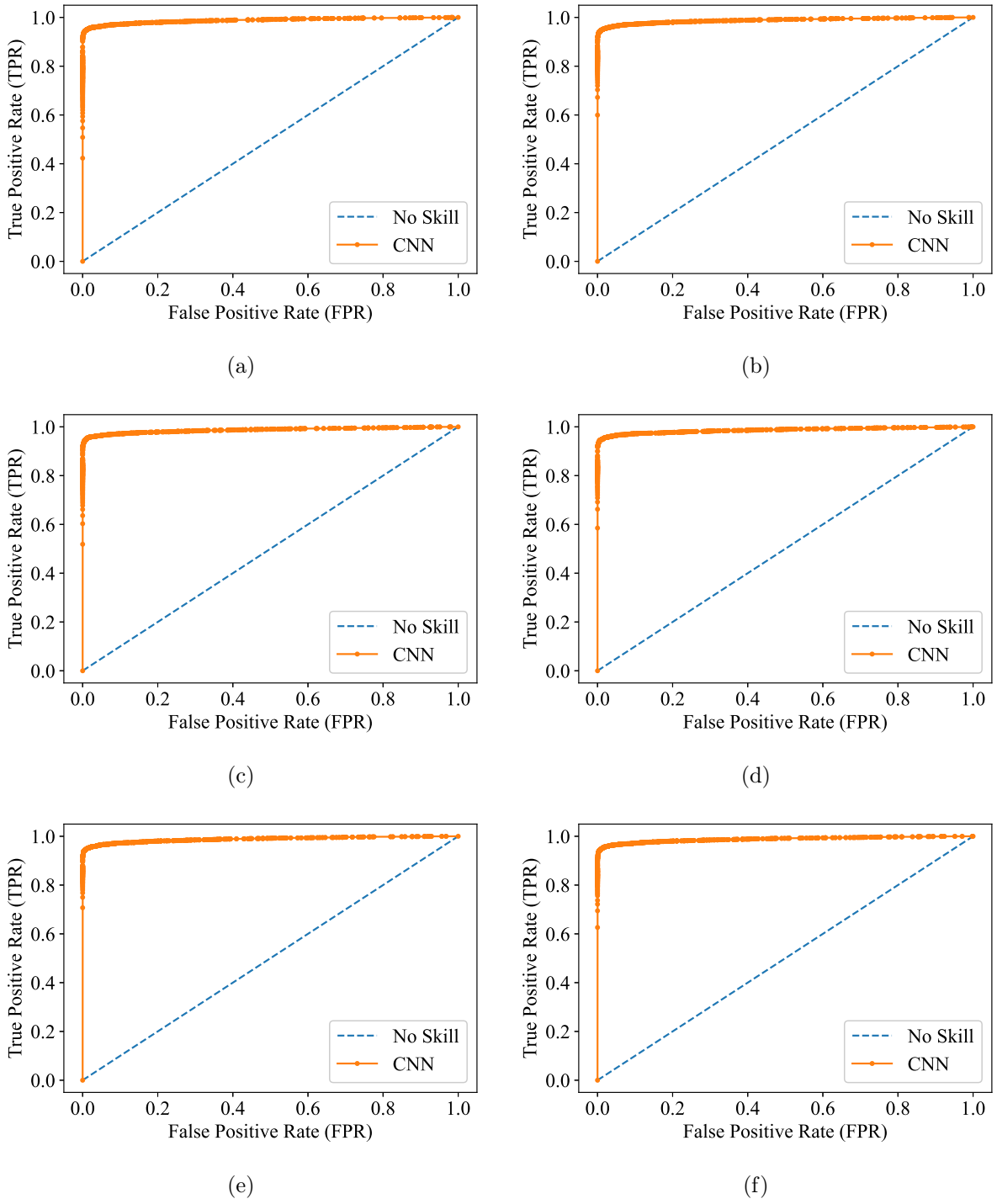


Figure A.4 ROC curves for evaluating the deep learning models combining the ResNet-101 CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

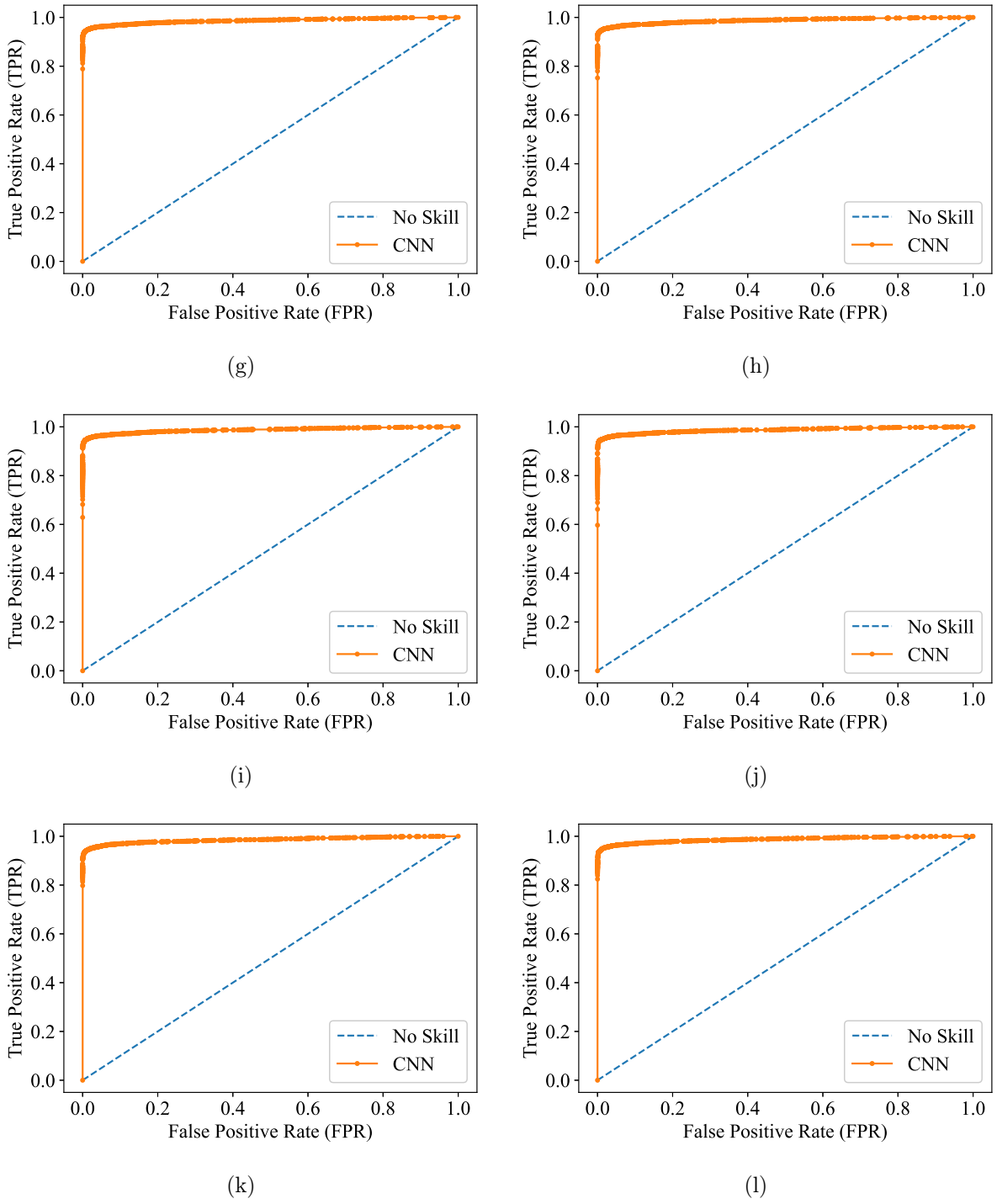


Figure A.4 (cont.) ROC curves for evaluating the deep learning models combining the ResNet-101 CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

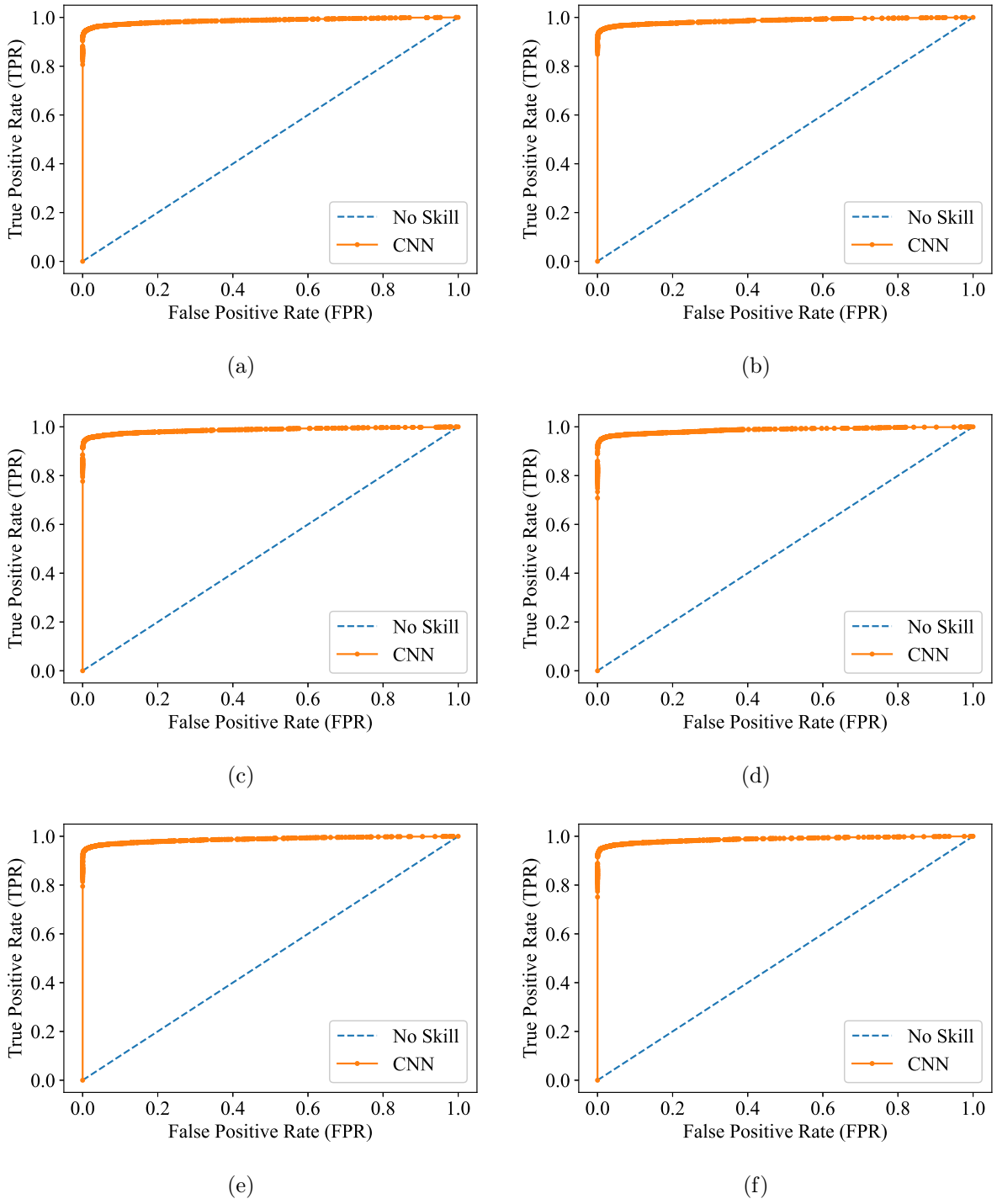


Figure A.5 ROC curves for evaluating the deep learning models combining the Xception CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

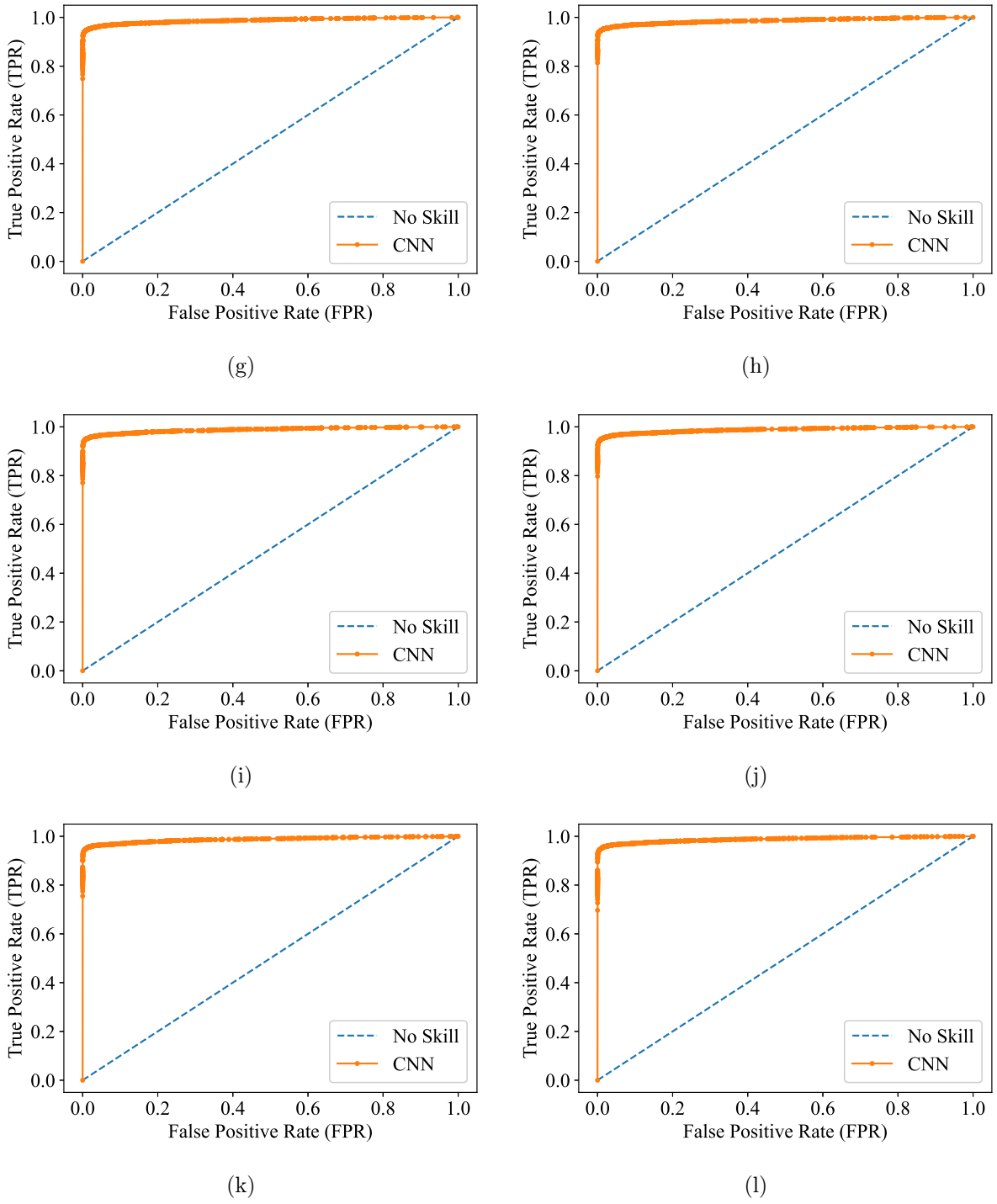


Figure A.5 (cont.) ROC curves for evaluating the deep learning models combining the Xception CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

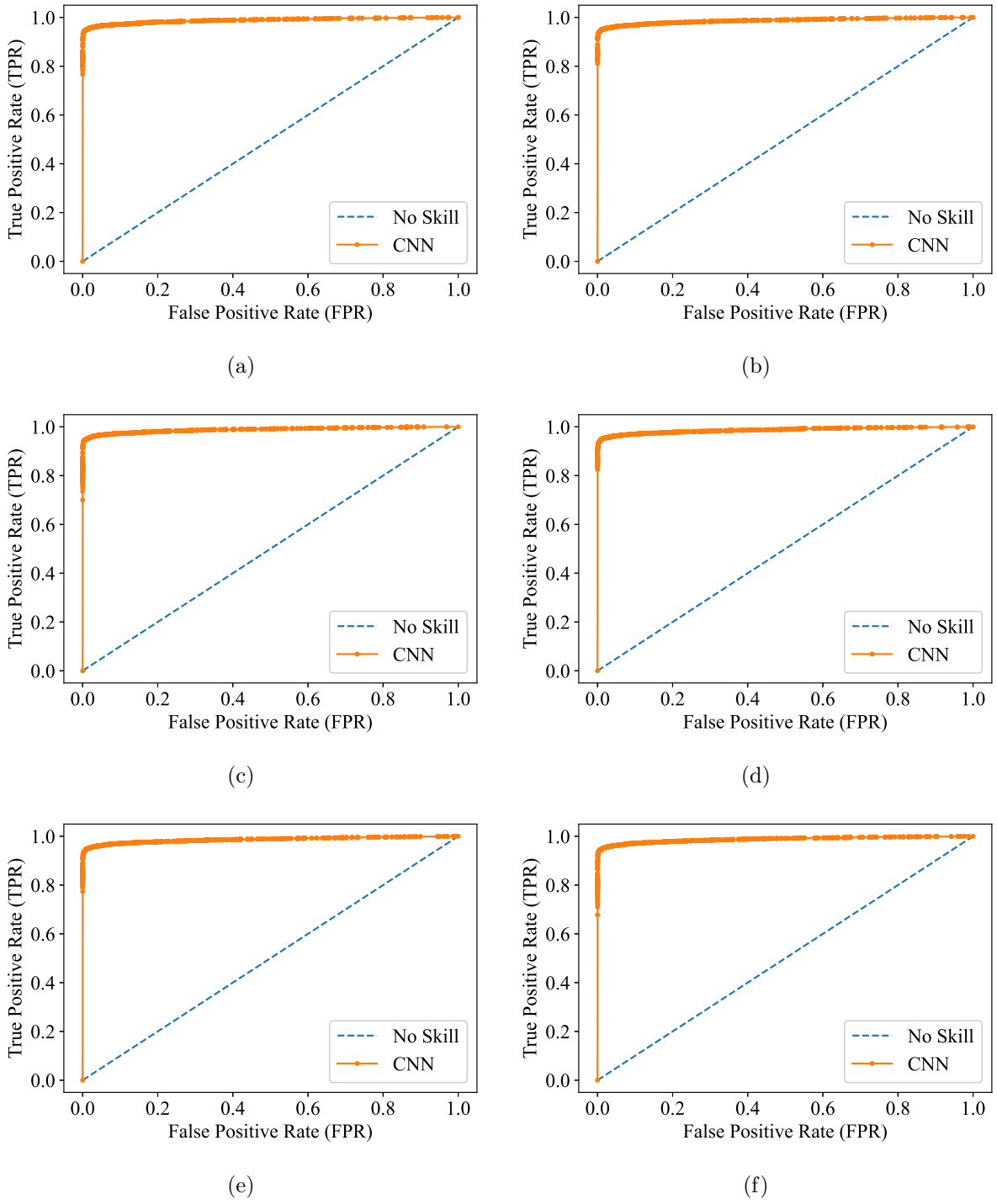
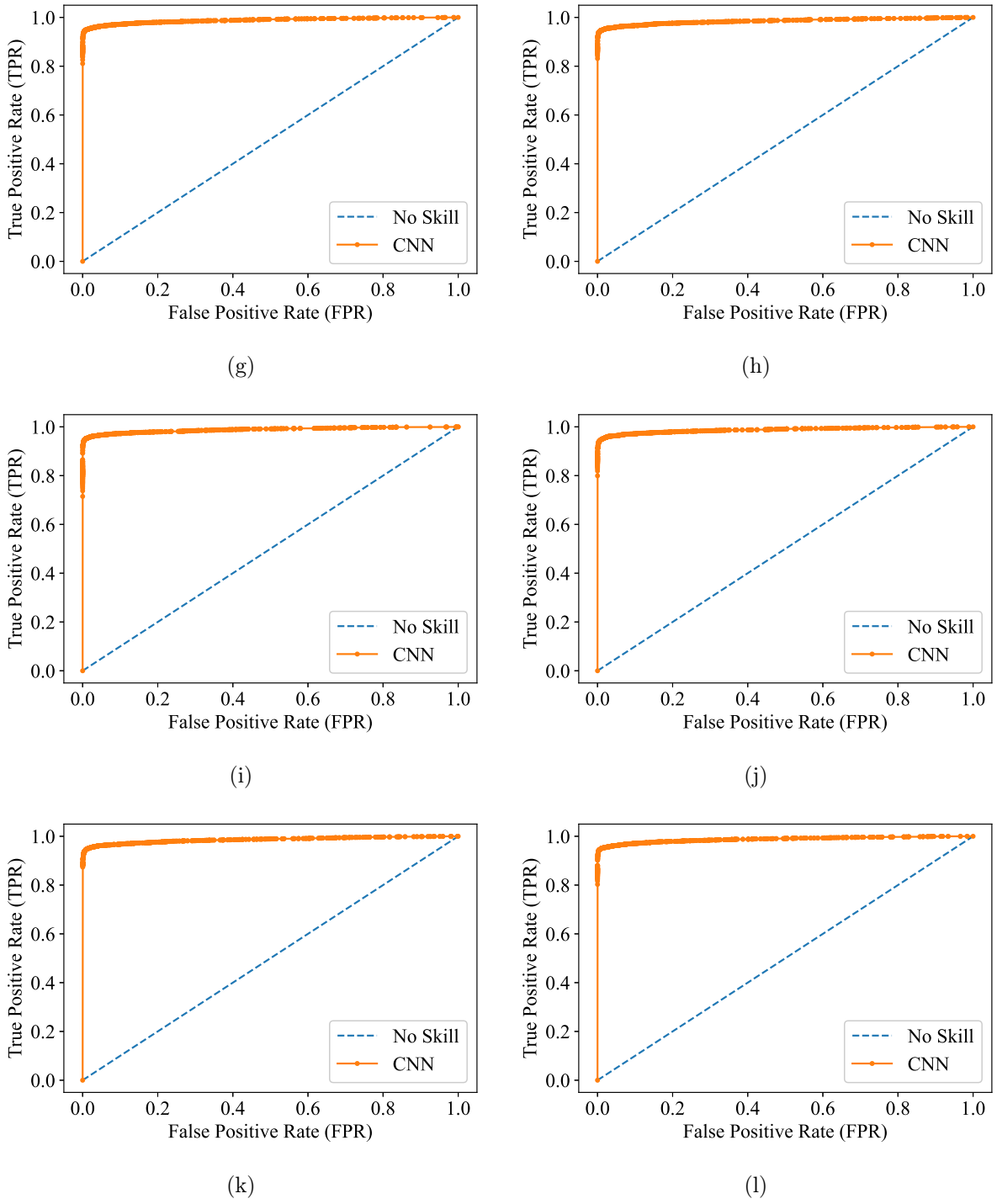


Figure A.6 ROC curves for evaluating the deep learning models combining the EfficientNet CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.



(g) (h) (i) (j) (k) (l)

Figure A.6 (cont.) ROC curves for evaluating the deep learning models combining the EfficientNet CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

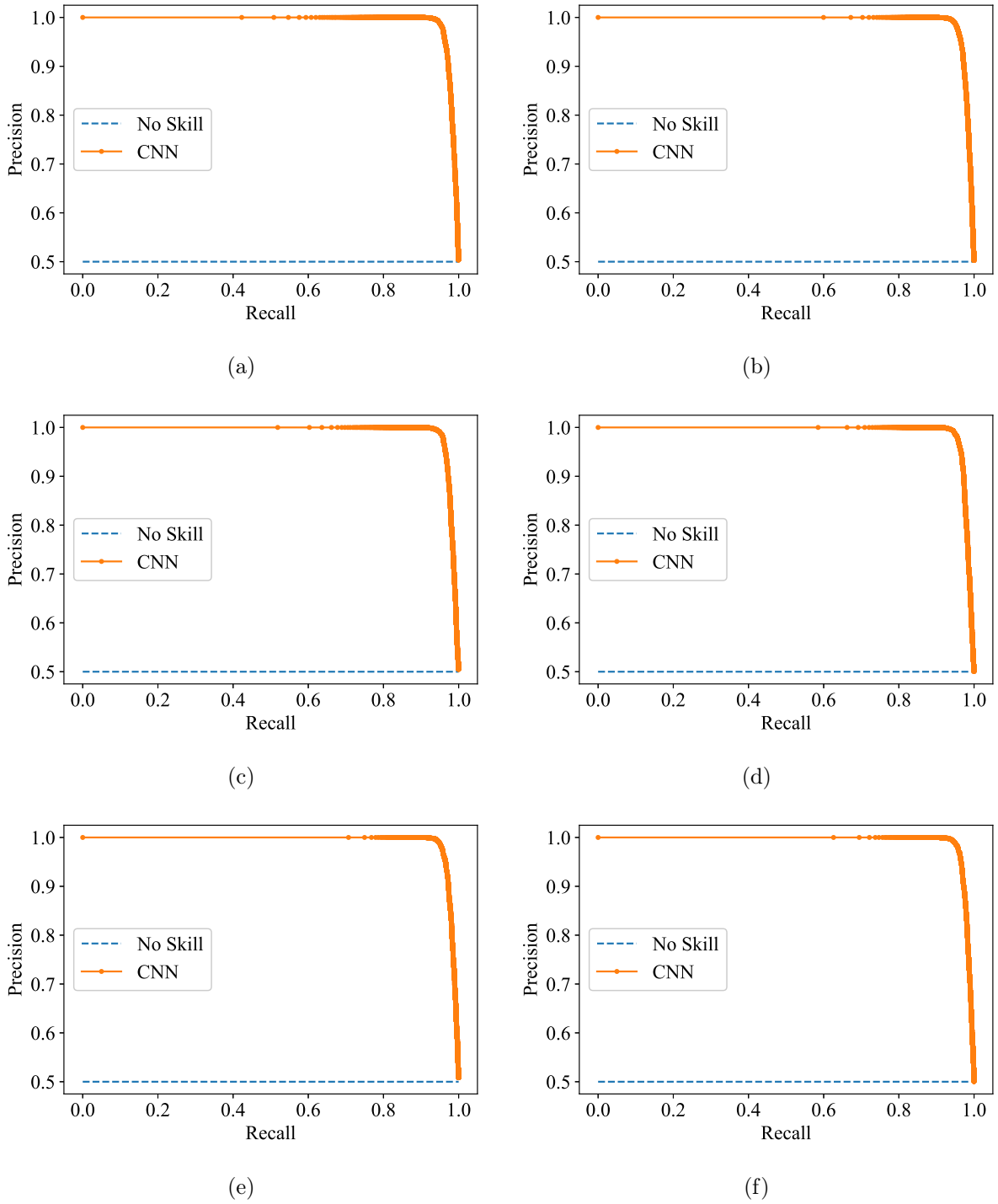


Figure A.7 Precision-recall curves for evaluating the deep learning models combining the ResNet-101 CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

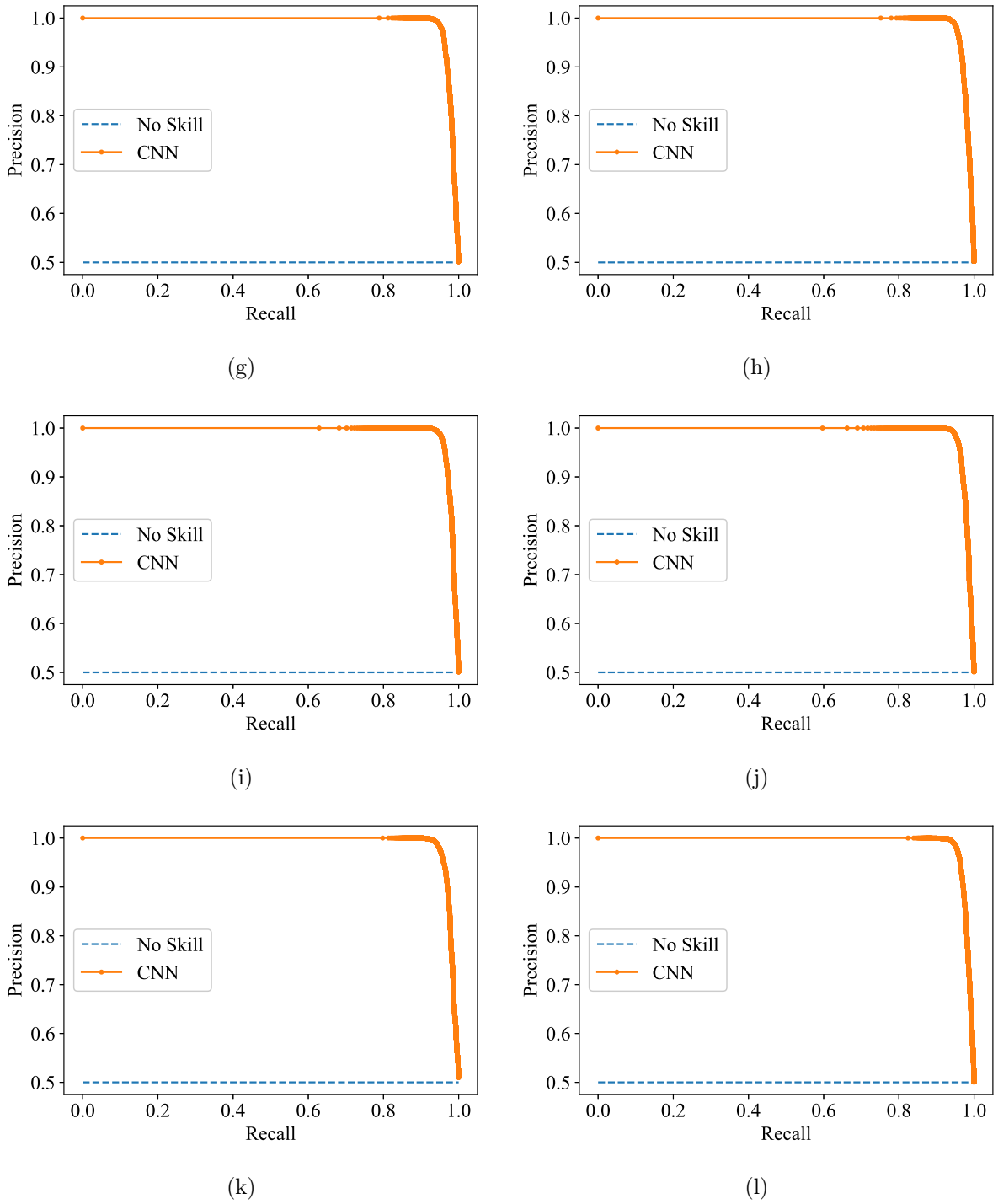


Figure A.7 (cont.) Precision-recall curves for evaluating the deep learning models combining the ResNet-101 CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

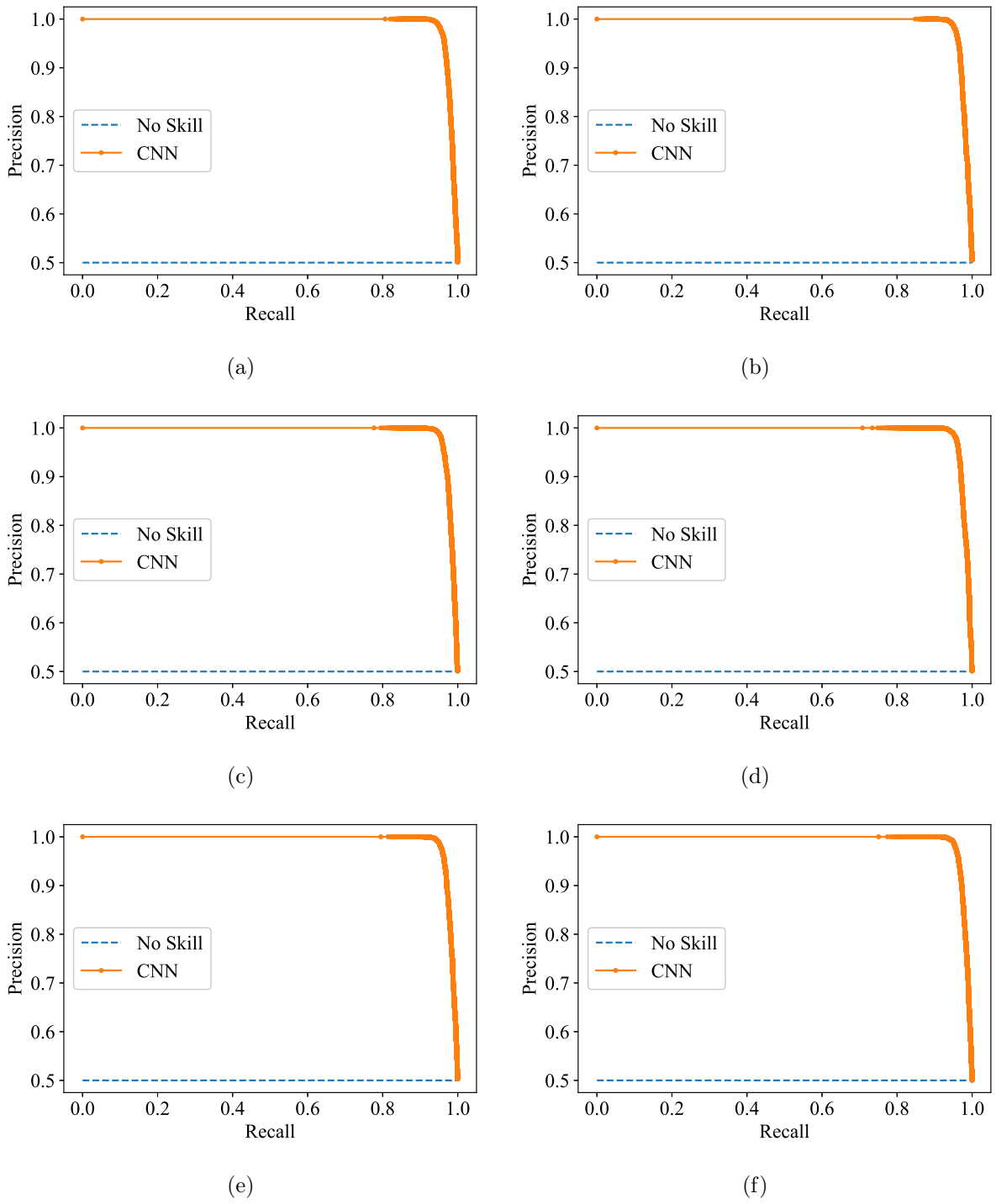


Figure A.8 Precision-recall curves for evaluating the deep learning models combining the Xception CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

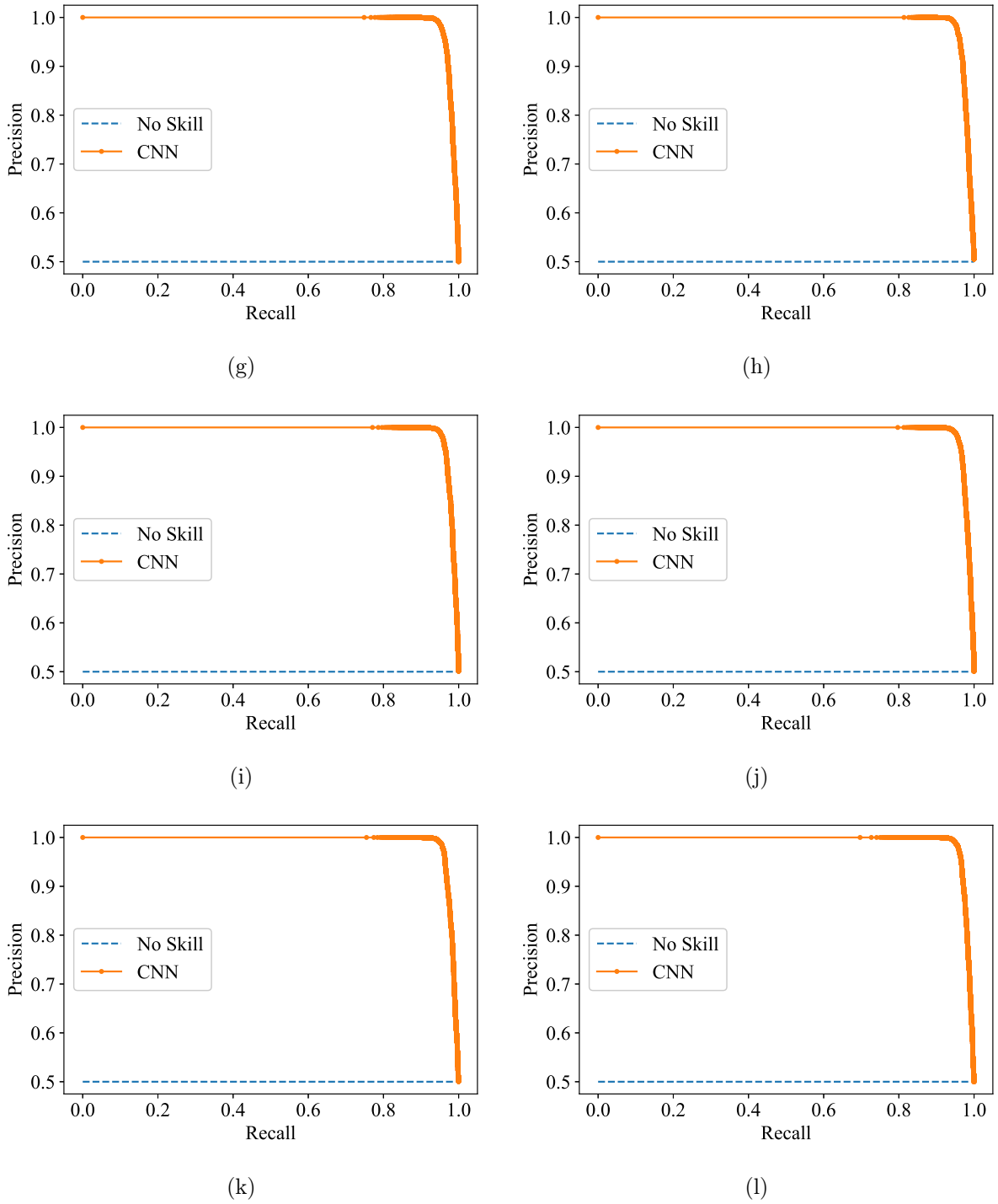


Figure A.8 (cont.) Precision-recall curves for evaluating the deep learning models combining the Xception CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

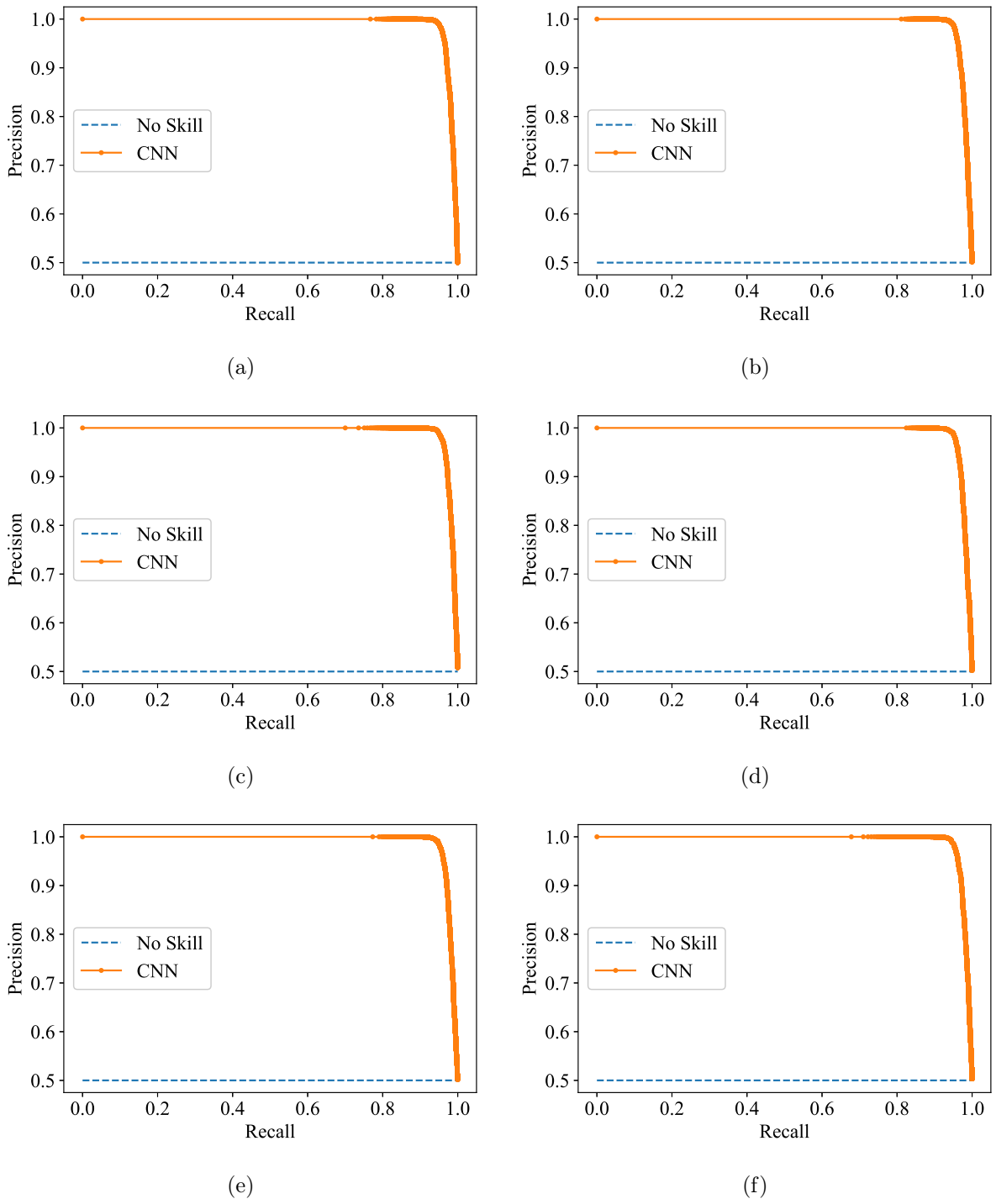


Figure A.9 Precision-recall curves for evaluating the deep learning models combining the EfficientNet CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

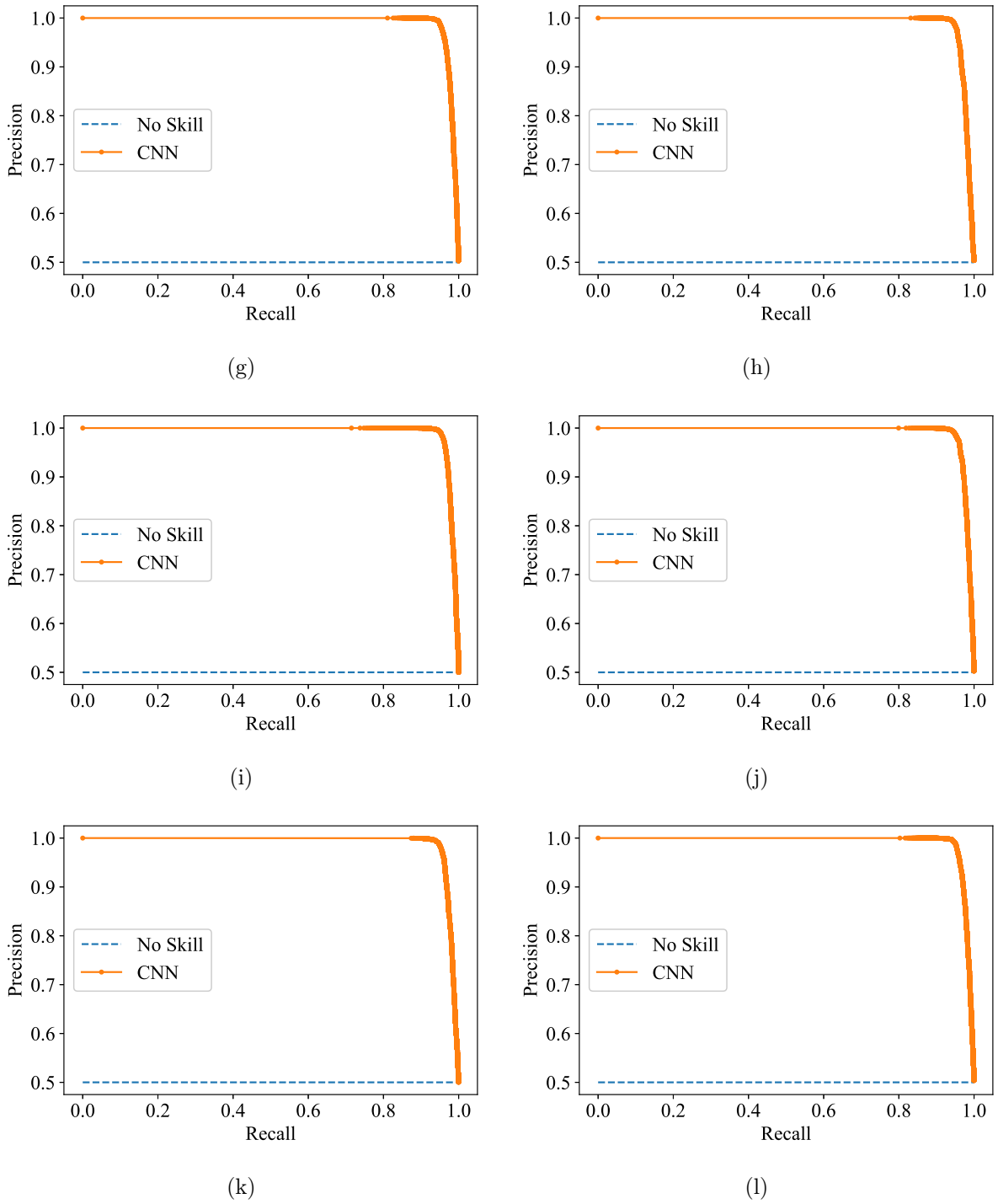


Figure A.9 (cont.) Precision-recall curves for evaluating the deep learning models combining the EfficientNet CNN architecture with the following TFDs of the input data: (a) BJD; (b) BUD; (c) CWD; (d) PWVD; (e) RIDB; (f) RIDBN; (g) RIDH; (h) RIDT; (i) SP; (j) SPWVD; (k) WVD; (l) ZAMD. The model evaluation is performed on the test dataset.

Appendix B

RESULTS OF THE STATISTICAL SIGNIFICANCE TESTS

This appendix presents more detailed results of the conducted statistical significance tests. The contingency tables used in McNemar's statistical tests to compare the baseline model with the CWD - ResNet-101, the WVD - Xception, and the SP - EfficientNet model are provided in Chapter 6. The contingency tables used to compare the baseline model with the rest of the TFD-CNN models are presented here.

Table B.1 Contingency table for the baseline model and the BJD - ResNet-101 model.

	BJD - ResNet-101 model Correct	**BJD - ResNet-101 model Incorrect**
Baseline model Correct	13830	142
Baseline model Incorrect	694	334

Table B.2 Contingency table for the baseline model and the BUD - ResNet-101 model.

	BUD - ResNet-101 model Correct	**BUD - ResNet-101 model Incorrect**
Baseline model Correct	13845	127
Baseline model Incorrect	680	348

Table B.3 Contingency table for the baseline model and the PWVD - ResNet-101 model.

	PWvD - ResNet-101 model Correct	**PWvD - ResNet-101 model Incorrect**
Baseline model Correct	13893	79
Baseline model Incorrect	640	388

Table B.4 Contingency table for the baseline model and the RIDB - ResNet-101 model.

	RIDB - ResNet-101 model Correct	**RIDB - ResNet-101 model Incorrect**
Baseline model Correct	13889	83
Baseline model Incorrect	639	389

Table B.5 Contingency table for the baseline model and the RIDBN - ResNet-101 model.

	RIDBN - ResNet-101 model Correct	**RIDBN - ResNet-101 model Incorrect**
Baseline model Correct	13869	103
Baseline model Incorrect	670	358

Table B.6 Contingency table for the baseline model and the RIDH - ResNet-101 model.

	RIDH - ResNet-101 model Correct	**RIDH - ResNet-101 model Incorrect**
Baseline model Correct	13882	90
Baseline model Incorrect	636	392

Table B.7 Contingency table for the baseline model and the RIDT - ResNet-101 model.

	RIDT - ResNet-101 model Correct	**RIDT - ResNet-101 model Incorrect**
Baseline model Correct	13865	107
Baseline model Incorrect	655	373

Table B.8 Contingency table for the baseline model and the SP - ResNet-101 model.

	SP - ResNet-101 model Correct	**SP - ResNet-101 model Incorrect**
Baseline model Correct	13877	95
Baseline model Incorrect	660	368

Table B.9 Contingency table for the baseline model and the SPWVD - ResNet-101 model.

	SPWVD - ResNet-101 model Correct	**SPWVD - ResNet-101 model Incorrect**
Baseline model Correct	13854	118
Baseline model Incorrect	660	368

Table B.10 Contingency table for the baseline model and the WVD - ResNet-101 model.

	WVD - ResNet-101 model Correct	**WVD - ResNet-101 model Incorrect**
Baseline model Correct	13862	110
Baseline model Incorrect	619	409

Table B.11 Contingency table for the baseline model and the ZAMD - ResNet-101 model.

	ZAMD - ResNet-101 model Correct	**ZAMD - ResNet-101 model Incorrect**
Baseline model Correct	13862	110
Baseline model Incorrect	660	368

Table B.12 Contingency table for the baseline model and the BJD - Xception model.

	BJD - Xception model Correct	**BJD - Xception model Incorrect**
Baseline model Correct	13856	116
Baseline model Incorrect	664	364

Table B.13 Contingency table for the baseline model and the BUD - Xception model.

	BUD - Xception model Correct	**BUD - Xception model Incorrect**
Baseline model Correct	13845	127
Baseline model Incorrect	679	349

Table B.14 Contingency table for the baseline model and the CWD - Xception model.

	CWD - Xception model Correct	**CWD - Xception model Incorrect**
Baseline model Correct	13905	67
Baseline model Incorrect	621	407

Table B.15 Contingency table for the baseline model and the PWVD - Xception model.

	PWVD - Xception model Correct	**PWVD - Xception model Incorrect**
Baseline model Correct	13883	89
Baseline model Incorrect	662	366

Table B.16 Contingency table for the baseline model and the RIDB - Xception model.

	RIDB - Xception model Correct	**RIDB - Xception model Incorrect**
Baseline model Correct	13829	143
Baseline model Incorrect	687	341

Table B.17 Contingency table for the baseline model and the RIDBN - Xception model.

	RIDBN - Xception model Correct	**RIDBN - Xception model Incorrect**
Baseline model Correct	13857	115
Baseline model Incorrect	679	349

Table B.18 Contingency table for the baseline model and the RIDH - Xception model.

	RIDH - Xception model Correct	**RIDH - Xception model Incorrect**
Baseline model Correct	13876	96
Baseline model Incorrect	654	374

Table B.19 Contingency table for the baseline model and the RIDT - Xception model.

	RIDT - Xception model Correct	**RIDT - Xception model Incorrect**
Baseline model Correct	13865	107
Baseline model Incorrect	664	364

Table B.20 Contingency table for the baseline model and the SP - Xception model.

	SP - Xception model Correct	**SP - Xception model Incorrect**
Baseline model Correct	13896	76
Baseline model Incorrect	647	381

Table B.21 Contingency table for the baseline model and the SPWVD - Xception model.

	SPWVD - Xception model Correct	**SPWVD - Xception model Incorrect**
Baseline model Correct	13866	106
Baseline model Incorrect	682	346

Table B.22 Contingency table for the baseline model and the ZAMD - Xception model.

	ZAMD - Xception model Correct	**ZAMD - Xception model Incorrect**
Baseline model Correct	13817	155
Baseline model Incorrect	706	322

Table B.23 Contingency table for the baseline model and the BJD - EfficientNet model.

	BJD - EfficientNet model Correct	**BJD - EfficientNet model Incorrect**
Baseline model Correct	13879	93
Baseline model Incorrect	669	359

Table B.24 Contingency table for the baseline model and the BUD - EfficientNet model.

	BUD - EfficientNet model Correct	**BUD - EfficientNet model Incorrect**
Baseline model Correct	13877	95
Baseline model Incorrect	657	371

Table B.25 Contingency table for the baseline model and the CWD - EfficientNet model.

	CWD - EfficientNet model Correct	**CWD - EfficientNet model Incorrect**
Baseline model Correct	13888	84
Baseline model Incorrect	659	369

Table B.26 Contingency table for the baseline model and the PWVD - EfficientNet model.

	PWVD - EfficientNet model Correct	**PWVD - EfficientNet model Incorrect**
Baseline model Correct	13867	105
Baseline model Incorrect	672	356

Table B.27 Contingency table for the baseline model and the RIDB - EfficientNet model.

	RIDB - EfficientNet model Correct	**RIDB - EfficientNet model Incorrect**
Baseline model Correct	13883	89
Baseline model Incorrect	642	386

Table B.28 Contingency table for the baseline model and the RIDBN - EfficientNet model.

	RIDBN - EfficientNet model Correct	**RIDBN - EfficientNet model Incorrect**
Baseline model Correct	13856	116
Baseline model Incorrect	664	364

Table B.29 Contingency table for the baseline model and the RIDH - EfficientNet model.

	RIDH - EfficientNet model Correct	**RIDH - EfficientNet model Incorrect**
Baseline model Correct	13875	97
Baseline model Incorrect	675	353

Table B.30 Contingency table for the baseline model and the RIDT - EfficientNet model.

	RIDT - EfficientNet model Correct	**RIDT - EfficientNet model Incorrect**
Baseline model Correct	13898	74
Baseline model Incorrect	630	398

Table B.31 Contingency table for the baseline model and the SPWVD - EfficientNet model.

	SPWVD - EfficientNet model Correct	**SPWVD - EfficientNet model Incorrect**
Baseline model Correct	13878	94
Baseline model Incorrect	658	370

Table B.32 Contingency table for the baseline model and the WVD - EfficientNet model.

	WVD - EfficientNet model Correct	**WVD - EfficientNet model Incorrect**
Baseline model Correct	13885	87
Baseline model Incorrect	638	390

Table B.33 Contingency table for the baseline model and the ZAMD - EfficientNet model.

	ZAMD - EfficientNet model Correct	**ZAMD - EfficientNet model Incorrect**
Baseline model Correct	13794	178
Baseline model Incorrect	691	337

CURRICULUM VITAE

Nikola Lopac was born on February 17, 1994, in Rijeka, Croatia. After completing his primary and secondary education as the best student of the generation in his hometown of Senj, he enrolled in the University of Rijeka, Faculty of Engineering, in 2012. He completed the undergraduate and the graduate university study of electrical engineering at the same institution in 2015 and 2017, respectively. He received both the bachelor's and the master's degree with *summa cum laude* honors, achieving a maximum grade point. During his studies, he received the following awards: University of Rijeka Rector's award for the best student of the generation in the Faculty of Engineering, the annual *Hrvoje Požar* award for outstanding performance during the studies from the Foundation of Croatian Energy Association, and five annual Faculty of Engineering Dean's awards for academic excellence. He also received several scholarships, including a scholarship from the City of Senj, a scholarship for excellence, and a scholarship for graduate students from the University of Rijeka. In 2017, he enrolled in the Faculty of Engineering for the Postgraduate university doctoral study in the area of engineering sciences in the field of electrical engineering.

During his undergraduate and graduate studies, he also worked as an undergraduate teaching assistant for several courses and as an automation engineer intern on several international projects in the industrial automation field. After graduation, from 2017 to 2020, he was a research and teaching assistant with the University of Rijeka, Faculty of Engineering, Department of Automation and Electronics. Since 2020, he has been a research and teaching assistant with the University of Rijeka, Faculty of Maritime Studies, Department of Electrical Engineering, Automation and Computing. Since 2020, he has also been a member of the research group within the Laboratory for Information Processing and Pattern Recognition at the University of Rijeka, Center for Artificial Intelligence and Cybersecurity.

He has participated in several scientific, industrial, and EU projects. He has been a teaching assistant in 12 different courses at the undergraduate university, graduate university, undergraduate vocational level of study, and in the lifelong learning programs. In addition, he has been a co-supervisor of three defended master's theses so far.

He has collaborated with several research groups at national and foreign academic institutions in his research work. He has co-authored five scientific articles in international

peer-reviewed journals indexed in the Current Contents (CC) database, one article in a peer-reviewed journal indexed in the Emerging Sources Citation Index (ESCI) in the Web of Science Core Collection (WoSCC) database, two scientific papers published in the proceedings of international conferences, two conference abstracts, and one professional paper. In addition, he has reviewed 31 scientific articles for 14 international CC-indexed journals and one scientific paper for an international conference in his research field. He was also extending his research contacts and skills by spending several months at foreign scientific institutions and participating in summer school in his research field. He received the research excellence award in the category of young researchers from the University of Rijeka, Faculty of Maritime Studies, for his research work in the academic year 2020/2021. His current research interests include statistical signal processing, time-frequency signal analysis, deep learning, and information processing and transmission.

LIST OF PUBLICATIONS

Scientific articles in peer-reviewed journals indexed in Current Contents (CC) database

1. N. Lopac, F. Hržić, I. Petrijevčanin Vuksanović, and J. Lerga, "Detection of non-stationary GW signals in high noise from Cohen's class of time-frequency representations using deep learning," *IEEE Access*, vol. 10, pp. 2408–2428, Jan. 2022, doi: 10.1109/ACCESS.2021.3139850.

2. 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, Dec. 2020, Art. no. 6920, doi: 10.3390/s20236920.

3. N. Lopac, I. Jurdana, J. Lerga, and N. Wakabayashi, "Particle-swarm-optimization-enhanced radial-basis-function-kernel-based adaptive filtering applied to maritime data," *Journal of Marine Science and Engineering*, vol. 9, no. 4, Apr. 2021, Art. no. 439, doi: 10.3390/jmse9040439.

4. I. Jurdana, N. Lopac, N. Wakabayashi, and H. Liu, "Shipboard data compression method for sustainable real-time maritime communication in remote voyage monitoring of autonomous ships," *Sustainability*, vol. 13, no. 15, Jul. 2021, Art. no. 8264, doi: 10.3390/su13158264.

5. N. Lopac, N. Bulić, and N. Vrkić, "Sliding mode observer-based load angle estimation for salient-pole wound rotor synchronous generators," *Energies*, vol. 12, no. 9, Apr. 2019, Art. no. 1609, doi: 10.3390/en12091609.

Scientific articles in peer-reviewed journals indexed in other Web of Science Core Collection (WoSCC) databases

1. N. Lopac, G. Šegon, and N. Bulić, "Application of model-based design tool X2C in induction machine vector control," *Engineering Review*, vol. 39, no. 1, pp. 90–104, Jan. 2019, doi: 10.30765/er.39.1.10.

Scientific papers in proceedings of international conferences

1. N. Lopac, J. Lerga, N. Saulig, L. Stanković, and M. Daković, "On optimal parameters for ICI-based adaptive filtering applied to the GWs in high noise," in *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech 2021)*, University of Split, Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture. Bol and Split, Croatia: IEEE, Sep. 2021, pp. 1–6, doi: 10.23919/SpliTech52315.2021.9566364.

2. N. Lopac and N. Bulić, "Estimation of the induction motor stator current frequency," in *International Conference on Innovative Technologies (IN-TECH 2018)*. Zagreb, Croatia: University of Rijeka, Faculty of Engineering, Sep. 2018, pp. 183–186.

Conference abstracts

1. N. Lopac, J. Lerga, and I. Jurdana, "On evolutionary metaheuristic optimization approaches in data-driven signal processing techniques," in *5th My First Conference (MFC 2021)*. Rijeka, Croatia: University of Rijeka, Faculty of Maritime Studies, Sep. 2021, p. 27.

2. N. Lopac and N. Bulić, "Time-frequency representations of induction machine stator current," in *2nd My First Conference (MFC 2018)*. Rijeka, Croatia: University of Rijeka, Faculty of Engineering, Sep. 2018, p. 17.

Professional papers

1. N. Lopac, "Implementation of induction machine vector control using X2C software tool," in *11th Days of Electrical Engineers*. Pula, Croatia: Croatian Chamber of Electrical Engineers, Sep. 2018, pp. 265–274.