

Alfaidi, Reem (2026) *Machine learning for early warning detection of gravitational wave signals.* PhD thesis.

https://theses.gla.ac.uk/86366/

Copyright and moral rights for this work are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This work cannot be reproduced or quoted extensively from without first obtaining permission from the author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

Enlighten: Theses
https://theses.gla.ac.uk/
research-enlighten@glasgow.ac.uk

Machine Learning for Early Warning Detection of Gravitational Wave Signals

Reem Alfaidi

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Physics and Astronomy
College of Science and Engineering
University Of Glasgow



May 2026

Declaration

All work presented in this thesis was carried out by the author under the supervision of Dr. Chris Messenger at the Institute for Gravitational Research, University of Glasgow. This thesis has not been submitted for any other degree at the University of Glasgow or any other institution. Chapters 1 and 2 provide background and introductory material on gravitational waves and machine learning.

Chapter 3 presents a literature review of classical and machine learning approaches for gravitational-wave data analysis, with a focus on early-warning detection of gravitational-wave signals.

Chapter 4 presents work that has been published in classical and quantum gravity [1]. The author led the development, implementation, and analysis of the machine learning model and produced the results presented in this chapter.

Chapters 5 and 6 present work including data analysis, model development, and evaluation on simulated and real gravitational-wave data.

Chapter 7 summarises the findings of this thesis and outlines directions for future work.

Grammarly, an AI-driven writing application, was used exclusively for proofreading, including grammar, spelling, and formatting.

إلى جدّتي،
التي علّمتني قبلَ العلم،
أنَّ الدهشةَ قد تولدُ في أبسطِ الأماكنِ وأصغرِها.

ورغمَ غيابِكِ،
ما يزالُ حبُّكِ يعبرُ إليَّ
كضوءِ نجمةٍ بعيدةٍ
يشقُّ طريقَهُ عبرَ الكون.

To my grandmother, who taught me, before science did,
that wonder can be found in the simplest and smallest of places.
Though you are gone, your love still reaches me
like the light of a distant star,
travelling across the universe.

Acknowledgements

Completing this PhD has been a long and meaningful journey, and I am deeply grateful to everyone who has supported me along the way.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr Chris Messenger, for his guidance, encouragement, patience, and continuous support throughout my PhD. I am particularly grateful for the trust he placed in me and for the many discussions that helped shape both this research and the way I think as a scientist. His advice, feedback, and willingness to explore new ideas with me have been invaluable throughout every stage of this work. It has been a privilege to carry out my PhD under his supervision.

I would also like to express my sincere thanks to my second supervisor, Professor Ik Siong Heng, for his guidance, support, and valuable advice throughout my PhD.

I am very grateful to my examiners, Professor Laura Nuttall and Dr Andrew Spencer, for such an engaging and insightful viva. I genuinely enjoyed the discussion and the opportunity to reflect on my research from different perspectives. Their thoughtful questions, constructive comments, and suggestions have undoubtedly helped to improve this thesis.

I would also like to thank everyone at the Institute for Gravitational Research at the University of Glasgow for their support throughout my PhD. Being part of the IGR has given me the opportunity to learn from researchers with a wide range of expertise, exchange ideas, and develop both academically and personally. I am grateful for all the discussions, advice, and shared experiences that have formed an important part of my time at Glasgow.

I would also like to express my sincere gratitude to Taibah University for its support throughout my PhD journey and for providing me with the opportunity to pursue my doctoral studies at the University of Glasgow. I am also deeply grateful to the Kingdom of Saudi Arabia for

the financial support and scholarship that made my doctoral studies and research possible.

I am especially grateful to my friends, both those who shared this PhD journey with me and those who supported me from outside academia. Thank you for listening, encouraging me, celebrating the good moments with me, and helping me through the more difficult ones. I am grateful for every conversation, coffee, laugh, and moment that gave me a chance to step away from the PhD and return with a clearer mind. Your friendship has meant more to me than I can express.

I would also like to thank my personal trainer, Shaun Dolan, for the support and perspective he has offered me throughout this journey. Many of our sessions came at times when I was feeling stressed or overwhelmed by work, and his advice, whether direct or indirect, often helped me to reset, reflect, and approach things with a better mindset. I am especially grateful for the way he has consistently encouraged me to keep improving and to become a better version of myself, both inside and outside the gym.

I am also grateful to my colleagues for their support, advice, and the many conversations we shared throughout these years. The opportunity to exchange ideas, discuss research problems, and learn from those around me has been an important part of this experience.

Above all, I am deeply grateful to my family for their unconditional love, patience, encouragement, and constant belief in me. Their support has been a source of strength throughout this journey, particularly during the most demanding stages of the PhD. I could not have reached this point without them.

Looking back, this PhD has been about much more than the research presented in this thesis. It has been a period of learning, growth, persistence, and discovery. I am incredibly grateful to everyone who has been part of this journey and helped me reach this point.

Abstract

The direct detection of gravitational waves has enabled detailed observations of compact binary coalescences, including binary black hole, binary neutron star, and neutron star-black hole systems. A central challenge in gravitational wave astronomy is the development of low-latency detection methods capable of identifying signals before merger, enabling rapid multi-messenger follow-up observations.

This thesis investigates the use of long short-term memory neural networks for early-warning detection of gravitational wave signals. The objective is to determine whether a data-driven model can reliably identify signals during the inspiral phase and provide early-warning of coalescence.

This work studies simulated datasets for binary black hole and binary neutron star systems and evaluates model performance using simulated signals embedded in both Gaussian and realistic detector noise. The long short-term memory model processes time-series strain data and produces per-timestep detection probabilities, enabling continuous monitoring of signal evolution.

These results demonstrate that long short-term memory-based models provide an effective and computationally efficient approach for early-warning detection of gravitational wave signals, establishing that reliable detection during the inspiral phase is achievable on physically meaningful timescales. The model can detect binary black hole signals several seconds before merger and binary neutron star signals up to approximately one minute before coalescence, depending on the signal strength. When applied to realistic detector noise following preprocessing and conditioning, the model continues to exhibit responses to injected signals. However, the results also highlight the challenges associated with detector non-stationarity and the mismatch between idealised Gaussian training conditions and real interferometer noise.

Thesis Overview

The direct detection of gravitational waves has opened a new observational window into the universe, enabling the study of compact binary coalescences such as binary black hole and binary neutron star systems. A key objective in gravitational wave astronomy is the development of early-warning detection methods capable of identifying signals before merger, thereby enabling low-latency multi-messenger observations.

This thesis investigates the use of long short-term memory neural networks for the early detection of gravitational wave signals and the estimation of the time remaining before merger.

The structure of the thesis is as follows.

Chapter 1 introduces the theoretical and observational foundations of gravitational wave astronomy. It presents the formulation of gravitational waves within general relativity, discusses their astrophysical sources, and reviews detection methods and data analysis techniques, including matched filtering and parameter estimation.

Chapter 2 provides an introduction to machine learning, with a focus on neural networks and sequence modelling. It develops the theoretical framework for recurrent neural networks and long short-term memory architectures, and discusses their suitability for analysing time-series data such as gravitational wave strain signals.

Chapter 3 reviews classical and machine-learning approaches to gravitational wave data analysis. It compares traditional matched-filtering pipelines with emerging machine-learning methods, with particular emphasis on early-warning detection strategies, and positions the contribution of this thesis within the broader context of existing work.

Chapter 4 presents the development of a long short-term memory-based model for early-warning detection of gravitational-wave signals. It describes the data generation process, model architecture, training procedure, and evaluation methodology, and analyses the performance of the model in comparison with matched-filtering approaches.

Chapter 5 extends the analysis to binary neutron star systems. It introduces a dedicated data generation pipeline and model adaptations required for long-duration signals, and evaluates detection sensitivity and early-warning performance.

Chapter 6 evaluates the trained model on realistic detector data from the O3Replay dataset. It describes the preprocessing pipeline applied to detector strain data and investigates model behaviour under realistic detector-noise conditions, including the effects of detector non-stationarity, transient artefacts, and the mismatch between idealised Gaussian training conditions and real interferometer noise.

Finally, Chapter 7 summarises the main findings of the thesis and outlines directions for future work, including potential extensions toward real-time deployment and integration with existing gravitational-wave detection pipelines.

This thesis demonstrates that machine-learning approaches, and long short-term memory-based models in particular provide a viable framework for early-warning detection of gravitational wave signals and represent a promising direction for future low-latency analysis in gravitational wave astronomy.

Contents

Declaration i

Acknowledgements iii

Abstract v

Thesis Overview vi

List of Figures xiii

List of Tables xv

List of Acronyms xvi

1 Introduction to Gravitational Waves 1

1.1 From General Relativity to Gravitational Waves 2

1.1.1 Weak-Field Approximation of Spacetime 3

1.1.2 Wave Equation 4

1.1.3 Effect of Gravitational Waves on Free Particles . . 5

1.1.4 The Quadrupolar Nature of Gravitational Waves . 7

1.1.5 Weakness of Gravitational-Wave Signals 8

1.2 Detection of Gravitational Waves 9

1.3 Sources of Gravitational Waves 12

1.3.1 Compact Binary Coalescence 12

1.3.2 Other Gravitational-Wave Sources 13

1.4 Gravitational-Wave Data Analysis 14

1.5 Matched Filtering 16

1.6 Practical Applications of Matched Filtering 17

1.6.1 Compact Binary Coalescence Searches 17

1.6.2 Continuous Wave Signals 18

1.6.3 Background Estimation and Time-Slides 19

1.7 Parameter Estimation 20

1.8 Bayesian Inference . . . 21
1.9 Multi-Messenger Astronomy . . . 21
1.10 Summary . . . 24

2 Introduction to Machine Learning 26
2.1 Machine Learning Key Concepts . . . 27
2.1.1 Types of Learning . . . 28
2.1.2 Learning Quality . . . 29
2.1.3 Prediction Challenges . . . 30
2.1.4 Enhancing Models . . . 31
2.2 Neural Network . . . 32
2.2.1 Neural Network Architectures . . . 32
2.2.2 Perceptron . . . 33
2.2.3 Activation Functions . . . 34
2.2.4 Multi-Layer Perceptron . . . 34
2.2.5 Logistic Regression . . . 36
2.2.6 Gradient Descent . . . 37
2.2.7 Gradient Descent Variants . . . 38
2.2.8 Backpropagation . . . 39
2.3 Recurrent Neural Networks . . . 39
2.3.1 Long Short-Term Memory . . . 43
2.3.2 LSTM Architectures . . . 45
2.4 Summary . . . 46

3 Gravitational-Wave Data Analysis: Classical and ML Methods 48
3.1 Signal Detection in GW Astronomy . . . 50
3.1.1 Classical Approaches . . . 50
3.1.2 Machine Learning Approaches . . . 52
3.2 Parameter Estimation . . . 56
3.2.1 Classical Bayesian Inference . . . 56
3.2.2 Machine Learning Approaches to Parameter Estimation . . . 57
3.3 Early-Warning Detection of Compact Binary Coalescences 59

3.3.1 Classical Early-Warning Pipelines 60
3.3.2 Limitations of Classical Early-Warning 63
3.3.3 Machine Learning Approaches to Early-Warning . . 64
3.3.4 Comparative Assessment of Classical and Machine Learning Approaches 67
3.3.5 Thesis Contribution and Position 70

4 LSTM for Early-Warning Detection of Gravitational Waves 72
4.1 Introduction 72
4.2 Recurrent Neural Networks 73
4.3 Data Generation Process 76
4.4 The LSTM Model 83
4.4.1 The LSTM Design 83
4.4.2 Training the LSTM Model 84
4.4.3 Testing the LSTM Model 86
4.4.4 Early-Warning Comparison with Matched-Filtering 87
4.4.5 Detection Sensitivity Comparison with Matched-Filtering 87
4.5 Results 88
4.5.1 LSTM Model Detection Sensitivity 88
4.5.2 Early-Warning Times from the LSTM Model 90
4.6 Conclusion and Discussion 100

5 Early-Warning Detection of Binary Neutron Star Systems Using Long Short-Term Memory Networks 104
5.1 Motivation and Overview 104
5.2 Generation of Binary Neutron Star Data 106
5.2.1 Waveform Synthesis and Priors 107
5.2.2 Sampling, Windowing, and Signal Alignment 110
5.2.3 Noise Generation and Whitening 111
5.2.4 Labelling via CSNR 113
5.2.5 Dataset Composition and Splits 116
5.3 LSTM Model Architecture and Training Adaptations . . . 116
5.3.1 Base Architecture and Temporal Output 116

5.3.2 Learning Rate Scheduling Strategies 117
5.3.3 Training Duration and Convergence 118
5.3.4 Summary of Model Adaptations 120
5.4 Training Procedure and Evaluation Methodology 120
5.4.1 Training Setup 120
5.4.2 Detection Statistic and Thresholding 121
5.4.3 False Alarm Rate Definition 121
5.4.4 Definition of Early-Warning Time 122
5.5 Detection Sensitivity and Early-Warning Performance . . . 123
5.5.1 Learning Rate Strategy Comparison 123
5.5.2 Distribution of Model Outputs 124
5.5.3 Representative Detection Behaviour 127
5.5.4 Early-Warning Performance 131
5.6 Detection Efficiency 134
5.7 Discussion and Conclusion 136

6 Evaluation of the Generalisation Performance of the LSTM Early-Warning Model on O3Replay Detector Data 141
6.1 Introduction and Motivation 141
6.2 O3Replay Data Description 143
6.2.1 Detector Strain Data 144
6.2.2 Injection Catalogue 148
6.3 Experimental Design and Evaluation Plan 151
6.4 Preprocessing Pipeline for O3Replay Strain 152
6.4.1 High-Pass Filtering 152
6.4.2 Whitening of the Strain Data 153
6.4.3 Low-Pass Filtering 154
6.4.4 Resampling and Downsampling 155
6.4.5 Construction of the Model Input Time Series 156
6.5 Verification of Preprocessing 156
6.5.1 Kolmogorov–Smirnov Test 156
6.6 Model Performance on Preprocessed O3Replay Data . . . 160

6.6.1 Statistical Diagnostics of the Preprocessed Detector Noise . . . 160
6.6.2 Model Behaviour on Noise-Only Data . . . 164
6.6.3 Model Response to Signal Injections in Noise . . . 170
6.7 Discussion and Conclusion . . . 177

7 Conclusions and Future Work **181**
7.1 Conclusions . . . 181
7.2 Future Work . . . 184

References **I**

List of Figures

1.1 GW polarisation modes (+/×) . . . 7
1.2 Michelson interferometer schematic . . . 10

2.1 Bias–variance trade-off . . . 31
2.2 Sigmoid vs Tanh . . . 35
2.3 ReLU vs Leaky ReLU . . . 35
2.4 Three-layer perceptron . . . 36
2.5 Neural network architecture and its recurrent formulation . 42
2.6 LSTM cell schematic . . . 45

4.1 Structure of an LSTM cell . . . 74
4.2 Advanced LIGO design-sensitivity PSD . . . 80
4.3 Simulated training time series examples . . . 81
4.4 Noise-only training example . . . 82
4.5 Training and validation loss evolution . . . 85
4.6 ROC comparison: LSTM vs matched filtering . . . 89
4.7 Early-warning detection comparison . . . 91
4.8 Detection time vs SNR . . . 93
4.9 Detection time vs chirp mass . . . 95
4.10 Early-warning detection example . . . 97
4.11 Signal detection example . . . 98
4.12 Noise-only detection example . . . 99
4.13 Signal vs noise output distribution . . . 100

5.1 Average H1 noise power spectral density . . . 112
5.2 BNS training examples with CSNR labelling . . . 115
5.3 Training and validation loss for BNS LSTM model . . . 119
5.4 ROC curves for BNS LSTM learning rate strategies . . . 124
5.5 Distribution of model output statistic Z (signal vs noise) . 125
5.6 Distribution of model output statistic Z for different CSNR ranges . . . 126

5.7 LSTM detection: signal+noise vs noise-only 130
5.8 Accumulated SNR at detection vs time before merger . . . 132
5.9 Estimated time to merger vs total SNR grouped by chirp mass 134
5.10 Detection Efficiency vs Inspiral SNR 135

6.1 O3Replay strain with glitches 146
6.2 PSD of O3Replay strain 147
6.3 Glitch in O3Replay strain 147
6.4 Injected CBC signal examples 149
6.5 Normalised Strain Distribution (KS Test) 159
6.6 Band-power evolution in preprocessed detector noise 161
6.7 Temporal evolution of rolling variance 162
6.8 Temporal evolution of rolling variance after 100 s local normalisation 163
6.9 Model Response to Gaussian Noise and Glitches 165
6.10 Glitch response in noise-only segments 166
6.11 Zoomed noise behaviour 167
6.12 LSTM Output Comparison for O3Replay and Gaussian Noise 168
6.13 SNR distributions for BNS injections 171
6.14 Long-duration LSTM output on O3Replay data 173
6.15 Example LSTM response to a BNS injection 175
6.16 Second example of LSTM response to a BNS injection . . . 176

List of Tables

4.1 BBH parameter prior distributions 77
4.2 LSTM vs CSNR detection comparison 92

5.1 Prior distributions for BNS parameters 108

List of Acronyms

GW	Gravitational Wave
LIGO	Laser Interferometer Gravitational-Wave Observatory
MCMC	Markov Chain Monte Carlo
LSC	LIGO Scientific Collaboration
BCE	Binary Cross-Entropy
PE	Parameter Estimation
BBH	Binary Black Hole
BNS	Binary Neutron Star
CBC	Compact Binary Coalescence
CNN	Convolutional Neural Network
GRB	Gamma-Ray Burst
GWTC-3	Third Gravitational-Wave Transient Catalog
SGWB	Stochastic Gravitational-Wave Background
LSTM	Long Short-Term Memory
PSD	Power Spectral Density
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SNR	Signal-to-Noise Ratio
PCA	Principal Component Analysis
MDP	Markov Decision Process

ML	Machine Learning
AUC	Area Under the Curve
MAE	Mean Absolute Error
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
DBI	Davies-Bouldin Index
CHI	Calinski-Harabasz Index
t-SNE	t-Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
EDA	Exploratory Data Analysis
GRU	Gated Recurrent Unit
MMA	Multi-Messenger Astronomy
FNN	Feedforward Neural Network
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation
ReLU	Rectified Linear Unit
MLP	Multi-Layer Perceptron
GPU	Graphics Processing Unit
RMSProp	Root Mean Squared Propagation
BPTT	Backpropagation Through Time
ANN	Artificial Neural Network
VGP	Vanishing Gradient Problem

BiLSTM	Bidirectional LSTM
ConvLSTM	Convolutional LSTM
LVK	LIGO-Virgo-KAGRA
MBTA	Multi-Band Template Analysis
FAR	False Alarm Rate
GF	Gravitational Field
TT	Transverse-Traceless
CSNR	Cumulative Signal-to-Noise Ratio
EM	Electromagnetic
CW	Continuous Wave
NSBH	Neutron Star-Black Hole
Tanh	Hyperbolic Tangent
NPE	Neural Posterior Estimation
IMR	Inspiral-Merger-Ringdown
O3	Third Observing Run
O4	Fourth Observing Run
IIR	Infinite Impulse Response
SPIIR	Summed Parallel Infinite Impulse Response
PR-AUC	Precision-Recall Area Under the Curve
FPGA	Field-Programmable Gate Array
CLDNN	Convolutional, Long Short-Term Memory, and Fully Connected Deep Neural Network

MDC	Mock Data Challenge
KS	Kolmogorov-Smirnov
GOTO	Gravitational-wave Optical Transient Observer

CHAPTER 1

Introduction to Gravitational Waves

The study of Gravitational Waves (GWs) lies at the intersection of theory, observation, and data analysis, linking Einstein's general relativity to state-of-the-art detection techniques and computational methods in modern physics. This chapter establishes the connection between Einstein's equations and the physical interpretation of GWs as propagating perturbations of spacetime. It further outlines the experimental principles underlying their detection, surveys the principal astrophysical sources responsible for their generation, and introduces the methodologies used to extract physical information from weak and noisy observational data. The chapter is structured to provide a coherent progression from theoretical foundations to practical detection and statistical interpretation.

We begin by examining how general relativity gives rise to GWs. In the weak-field regime, spacetime can be approximated as nearly flat, with small perturbations that satisfy a wave equation. This approximation provides a clear and intuitive framework for understanding GWs as carriers of energy and information emitted by dynamical astrophysical systems. The discussion then turns to the experimental challenge of detecting these weakly interacting signals. This is achieved using kilometre-scale laser interferometers, which employ suspended mirrors and coherent laser beams to measure spacetime distortions far smaller than the size of a proton, enabling direct comparison between theoretical predictions and observational data.

With detection established, the focus turns to the astrophysical sources of GWs. These include Compact Binary Coalescences (CBCs), rotating neutron stars, stochastic backgrounds, and transient burst events. Each source class is characterised by distinct physical mechanisms and signal morphologies, which, in turn, inform the design of targeted search strategies. This naturally leads to the problem of data analysis for GW

detection, where the objective is to identify weak signals embedded within large volumes of noisy, omnidirectional detector data. Addressing this challenge requires the development of sophisticated algorithms, cross-detector consistency checks, and computationally efficient search techniques. A central component of GW data analysis is matched filtering, which provides an optimal framework for detecting signals of known form in stationary noise. By correlating detector data with template waveforms, matched filtering enables the identification of CBC signals and other well-modelled sources, albeit at significant computational cost due to the high dimensionality of the parameter space.

Interpreting detected signals requires robust statistical methods. In particular, Parameter Estimation (PE) relies on maximum likelihood techniques and Bayesian inference to determine source properties such as masses, spins, distances, and orientations, while also quantifying associated uncertainties. These approaches form the basis for modern GW inference and are essential for extracting astrophysical insight from observed events. This framework naturally extends to Multi-Messenger Astronomy (MMA), where GW observations are combined with Electromagnetic (EM), neutrino, and cosmic-ray data to provide a more comprehensive understanding of astrophysical phenomena.

Overall, this chapter provides a structured foundation for the study of GW physics, progressing from the theoretical description of spacetime perturbations to their detection, classification, and interpretation. These elements collectively establish the conceptual and methodological basis required for the development of early-warning detection techniques explored in this thesis.

1.1 From General Relativity to Gravitational Waves

The study of GWs begins with a simplified treatment of spacetime under weak Gravitational Fields (GFs). In general relativity, the geometry of spacetime is curved by matter and energy [2]. However, in regions where the GF is weak, spacetime is only slightly perturbed from its flat, Minkowski form. This approximation not only makes the mathematics

more tractable but also allows us to identify and interpret GWs as small ripples propagating through spacetime. To build towards this understanding, we proceed step by step from the approximation of weakly curved spacetime, to the identification of wave-like solutions, to their observable physical effects and astrophysical origins, and finally to the immense experimental challenge of detection. This progression not only frames the theory of GWs but also motivates its significance in modern physics and astrophysics.

1.1.1 Weak-Field Approximation of Spacetime

In the absence of a GF, spacetime is described by the flat Minkowski metric. In the presence of a weak GF, spacetime can be approximated as nearly flat, with small deviations from Minkowski geometry. Under this weak-field approximation following Ref. [2], the metric tensor can be expressed as

$$g_{\alpha\beta} = \eta_{\alpha\beta} + h_{\alpha\beta}, \tag{1.1}$$

where $\eta_{\alpha\beta} = \mathrm{diag}(-1, 1, 1, 1)$ is the Minkowski metric, and $h_{\alpha\beta}$ represents a small perturbation satisfying $|h_{\alpha\beta}| \ll 1$. A coordinate system in which Eq. 1.1 holds is referred to as an approximately Lorentzian coordinate system. This formulation allows the Einstein field equations to be linearised, significantly simplifying their structure while preserving the essential physics of weak GFs. In this limit, the metric perturbations behave as tensor fields propagating on a flat spacetime background.

To first order in $h_{\alpha\beta}$, the Riemann tensor simplifies to

$$R_{\alpha\beta\gamma\delta} = \frac{1}{2}\left(h_{\alpha\delta,\beta\gamma} + h_{\beta\gamma,\alpha\delta} - h_{\alpha\gamma,\beta\delta} - h_{\beta\delta,\alpha\gamma}\right), \tag{1.2}$$

where higher-order terms have been neglected. This expression is invariant under infinitesimal gauge transformations at linear order. By contraction, the Ricci tensor is obtained as

$$R_{\mu\nu} = \frac{1}{2}\left(h^{\alpha}{}_{\mu,\nu\alpha} + h^{\alpha}{}_{\nu,\mu\alpha} - h_{\mu\nu,\alpha}{}^{,\alpha} - h_{,\mu\nu}\right), \tag{1.3}$$

with $h \equiv \eta^{\alpha\beta} h_{\alpha\beta}$. The corresponding scalar curvature is given by

$$R = \eta^{\alpha\beta} R_{\alpha\beta}. \tag{1.4}$$

The Einstein tensor then takes the form

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}R. \tag{1.5}$$

For convenience, it is standard to introduce the trace-reversed metric perturbation

$$\bar{h}_{\mu\nu} \equiv h_{\mu\nu} - \frac{1}{2}\eta_{\mu\nu}h, \tag{1.6}$$

which allows the Einstein tensor to be written in a more compact form

$$G_{\mu\nu} = -\frac{1}{2}\left(\bar{h}_{\mu\nu,\alpha}{}^{,\alpha} + \eta_{\mu\nu}\bar{h}_{\alpha\beta}{}^{,\alpha\beta} - \bar{h}_{\mu\alpha,\nu}{}^{,\alpha} - \bar{h}_{\nu\alpha,\mu}{}^{,\alpha}\right). \tag{1.7}$$

This linearised formulation provides the foundation for describing GWs as propagating perturbations of spacetime in the weak-field regime.

1.1.2 Wave Equation

Starting from the Einstein field equations

$$G_{\mu\nu} = 8\pi T_{\mu\nu}, \tag{1.8}$$

and using the linearised expression for the Einstein tensor (Eq. 1.7), we obtain [2]

$$-\bar{h}_{\mu\nu,\alpha}{}^{,\alpha} - \eta_{\mu\nu}\bar{h}_{\alpha\beta}{}^{,\alpha\beta} + \bar{h}_{\mu\alpha,\nu}{}^{,\alpha} + \bar{h}_{\nu\alpha,\mu}{}^{,\alpha} = 16\pi T_{\mu\nu}. \tag{1.9}$$

A convenient simplification is achieved by adopting the Lorenz gauge condition

$$\partial^{\alpha}\bar{h}_{\mu\alpha} = 0, \tag{1.10}$$

which reduces Eq. 1.9 to

$$-\bar{h}_{\mu\nu,\alpha}{}^{,\alpha} = 16\pi T_{\mu\nu}. \tag{1.11}$$

In free space, where $T_{\mu\nu} = 0$, this equation takes the form

$$\Box\,\bar{h}_{\mu\nu} = \left(-\frac{\partial^2}{\partial t^2} + \nabla^2\right)\bar{h}_{\mu\nu} = 0, \tag{1.12}$$

where $\Box$ denotes the d'Alembert operator and ∇^2 is the Laplacian. Restoring the speed of light c, this can be written as

$$\left(-\frac{\partial^2}{\partial t^2} + c^2\nabla^2\right)\bar{h}_{\mu\nu} = 0. \tag{1.13}$$

Eq. 1.13 demonstrates that, in the weak-field limit, the Einstein field equations reduce to a wave equation for the metric perturbations, indicating that GWs propagate at the speed of light.

The general solution to Eq. 1.12 can be expressed as a plane wave

$$\bar{h}_{\mu\nu} = \mathrm{Re}\left[A_{\mu\nu}\exp\left(ik_\alpha x^\alpha\right)\right], \tag{1.14}$$

where $A_{\mu\nu}$ is the wave amplitude and k_α is the wave vector. The Lorenz gauge condition implies

$$A_{\mu\alpha}k^\alpha = 0, \tag{1.15}$$

and the null condition on the wave vector

$$k_\alpha k^\alpha = 0, \tag{1.16}$$

indicates propagation at the speed of light. Further gauge freedom can be used to reduce the number of independent components of $A_{\mu\nu}$ to two, corresponding to the physical degrees of freedom of GWs. In the Transverse-Traceless (TT) gauge, the metric perturbation takes the simplified form

$$h^{\mathrm{TT}}_{\mu\nu} = A^{\mathrm{TT}}_{\mu\nu}\cos\left[\omega(t-z)\right], \tag{1.17}$$

where $A^{\mathrm{TT}}_{\mu\nu}$ is the wave amplitude tensor in the TT gauge, ω is the angular frequency, t is time, and z denotes the direction of wave propagation. Only two independent components remain, corresponding to the two polarisation modes of GWs.

1.1.3 Effect of Gravitational Waves on Free Particles

The physical significance of GWs becomes evident when examining their effect on freely falling particles. Consider a set of freely falling particles initially at rest in a local Lorentz frame. In this coordinate system,

the particles remain at fixed coordinate positions as a GW passes. However, this does not imply the absence of physical effects, since the proper distances between particles evolve with time. To quantify this effect, we consider the proper distance between two nearby particles separated along the x-axis. To first order in the perturbation, the proper distance can be written as

$$\Delta\ell \simeq \left[1 + \frac{1}{2}h_{xx}^{\mathrm{TT}}(t)\right]\epsilon, \tag{1.18}$$

where ϵ is the initial separation. As $h_{xx}^{\mathrm{TT}}(t)$ varies with time, the distance between the particles oscillates, indicating the presence of a GW.

A more precise description is obtained using the geodesic deviation equation

$$\frac{\partial^2\xi^\alpha}{\partial t^2} = -R^\alpha{}_{\mu\nu\beta}\, u^\mu u^\nu\, \xi^\beta, \tag{1.19}$$

where ξ^α represents the separation vector between two nearby particles. For particles initially at rest, this reduces to

$$\frac{\partial^2\xi^i}{\partial t^2} = -R^i{}_{tjt}\, \xi^j. \tag{1.20}$$

In the TT gauge, the relevant components of the Riemann tensor are directly related to the metric perturbations, leading to

$$\frac{\partial^2\xi^x}{\partial t^2} = \frac{1}{2}\frac{\partial^2}{\partial t^2}h_{xx}^{\mathrm{TT}}\,\xi^x, \quad \frac{\partial^2\xi^y}{\partial t^2} = \frac{1}{2}\frac{\partial^2}{\partial t^2}h_{xy}^{\mathrm{TT}}\,\xi^x. \tag{1.21}$$

These equations show that GW induces oscillatory motion in the separation between particles.

For a plane GW propagating along the z-axis, the metric perturbation in the TT gauge can be written as

$$h_{\mu\nu}^{\mathrm{TT}} = B_{\mu\nu}^{\mathrm{TT}} \cos\left[\omega(t-z)\right]. \tag{1.22}$$

The resulting particle motion reveals two distinct polarisation modes. The h_{xx}^{TT} component produces the plus (+) polarisation, while the h_{xy}^{TT} component produces the cross ($\times$) polarisation. These modes correspond to orthogonal patterns of stretching and compression in the plane transverse to the direction of propagation. The effect of these two polarisation

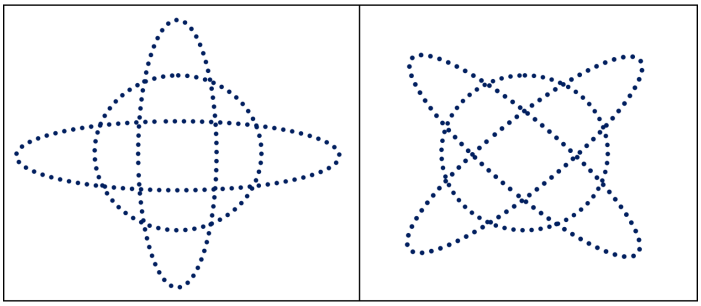


Figure 1.1: Illustration of the effect of a GW on a ring of free test particles in the transverse plane. The left panel shows the plus (+) polarisation mode, characterised by alternating stretching and compression along orthogonal axes. The right panel shows the cross (×) polarisation mode, rotated by 45° with respect to the plus mode. In each case, the deformation of the ring is shown at three distinct values of ωt (i.e. different phases of the wave), as described by Eq. 1.22.

modes on a ring of free particles is illustrated in Fig. 1.1. This behaviour forms the basis for gravitational-wave detection experiments such as laser interferometers.

1.1.4 The Quadrupolar Nature of Gravitational Waves

GWs are generated by the accelerated motion of mass distributions. However, not all forms of motion produce detectable radiation. In the weak-field limit, the leading contribution to GW emission arises from the time variation of the mass quadrupole moment. Throughout this section, we adopt geometrised units in which $G = c = 1$. The amplitude of the emitted GWs can be expressed, to lowest order, as [2]

$$h_{\mu\nu} = \frac{2}{r}\frac{d^2}{dt^2}Q_{\mu\nu}, \tag{1.23}$$

where $Q_{\mu\nu}$ is the mass quadrupole moment tensor, defined as

$$Q_{\mu\nu} = \int \rho\, x_\mu x_\nu \, \mathrm{d}^3x, \tag{1.24}$$

with ρ denoting the mass density.

The absence of lower-order radiation terms can be understood from fundamental conservation laws. A monopole term would correspond to

the total mass of the system, which is conserved and therefore does not vary with time. As a result, monopole radiation cannot occur. Similarly, a dipole term is associated with the motion of the centre of mass. Conservation of linear momentum ensures that the centre of mass of an isolated system moves with constant velocity, implying no net acceleration and hence no dipole radiation. Consequently, the leading-order contribution to GW emission arises from the quadrupole moment, which captures time-dependent distortions in the mass distribution. Systems with a varying quadrupole moment, such as binary systems in orbital motion, are therefore efficient sources of GW. This result also illustrates why GW emission is intrinsically weak. The absence of monopole and dipole radiation implies that the leading contribution arises only at the quadrupole order, involving higher-order time variations of the mass distribution. In contrast, EM radiation can be generated at the dipole level and is further enhanced by the much stronger EM coupling compared to gravity.

1.1.5 Weakness of Gravitational-Wave Signals

GWs are inherently weak, making their detection extremely challenging. This weakness arises from the quadrupolar nature of gravitational radiation and the small coupling strength of gravity. The quadrupole formula (Eq. 1.23) shows that the amplitude of the strain depends on the second time derivative of the mass quadrupole moment and decreases with distance from the source [2, 3]. As illustrated below, an order-of-magnitude estimate using typical astrophysical parameters shows that the resulting strain is extremely small, thereby explaining the difficulty of direct detection. The dimensionless strain h represents the fractional change in the separation between two test masses, defined as

$$h \sim \frac{\Delta L}{L}, \tag{1.25}$$

where L is the proper separation and ΔL is the change induced by the passing GW [2].

In geometric units, where $G = c = 1$, mass and distance are expressed in the same units. Under these conventions, the amplitude of the

GW strain can be estimated as:

$$h \sim \frac{Q^{(2)}}{r}, \tag{1.26}$$

where $Q^{(2)}$ denotes the second time derivative of the quadrupole moment and r is the distance to the source [3]. Near the source, the perturbation to the spacetime metric can be significant. However, GWs propagate outward, and their amplitude decreases inversely with distance. For astrophysical sources located at cosmological distances, this leads to extremely small strain amplitudes at Earth [2]. For example, a compact binary system located at a distance of tens to hundreds of megaparsecs can produce a strain at Earth of order $h \sim 10^{-21}$ or smaller [3]. Such values correspond to changes in length that are many orders of magnitude smaller than the size of an atomic nucleus. This extreme weakness presents a major experimental challenge, requiring highly sensitive detectors capable of measuring minute variations in distance.

1.2 Detection of Gravitational Waves

Ground-based GWs are detected using highly sensitive laser interferometers based on the Michelson interferometer configuration [4], as shown in Fig. 1.2. These detectors are designed to measure extremely small changes in the relative lengths of two orthogonal arms induced by a passing GW. In a basic Michelson interferometer, laser light is split into two perpendicular arms by a beam splitter, as illustrated in Fig. 1.2. In GW detectors such as LIGO and Virgo, Fabry–Perot cavities are incorporated into the arms, allowing the light to undergo many round trips within the cavities and thereby increasing the effective optical path length and enhancing the detector sensitivity [4, 5]. For example, in Advanced Virgo, the light undergoes approximately 280 round trips within the arm cavities [5]. The beams are then recombined at the beam splitter, producing an interference pattern at the photodetector. In addition to the intrinsic weakness of GW signals, detector noise poses a major challenge to their detection. Ground-based interferometers such as Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo are affected

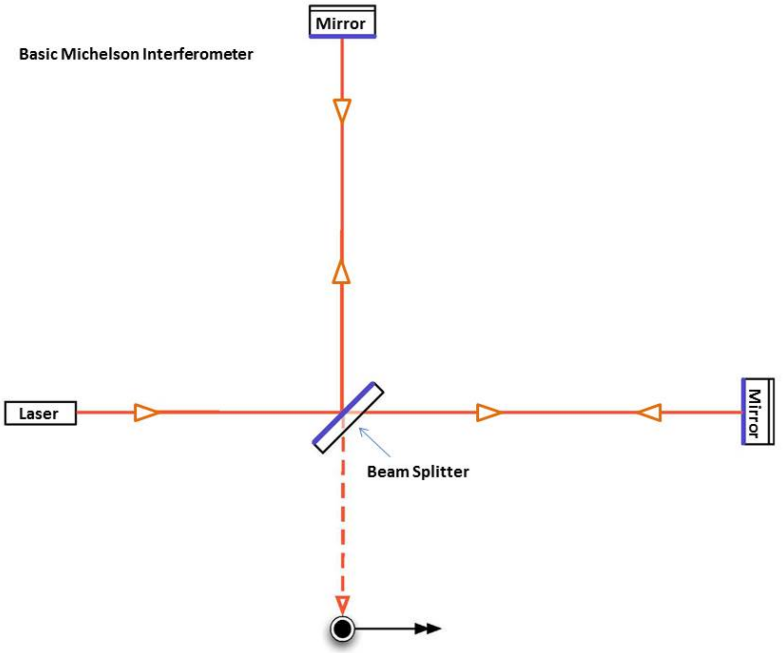


Figure 1.2: Schematic of a Michelson interferometer used in gravitational-wave detection. Source: [4].

by a variety of noise sources across different frequency ranges. The frequency content of a GW signal evolves with time. In particular, during the inspiral of a compact binary, the signal sweeps from lower to higher frequencies as the binary approaches merger. The frequency-dependent sensitivity of the detector therefore determines which parts of the signal can be observed most effectively. At low frequencies, below approximately 20-30 Hz, seismic and environmental disturbances dominate, while thermal noise from the mirrors and suspensions becomes significant at intermediate frequencies. At higher frequencies, above a few hundred Hz, quantum noise, primarily photon shot noise, limits the sensitivity of the detectors [6, 7]. Furthermore, the noise in GW detectors is neither perfectly stationary nor perfectly Gaussian. Stationarity refers to whether the statistical properties of the noise remain constant with time, whereas Gaussianity refers to whether the distribution of noise amplitudes follows a Gaussian probability distribution. Real detector noise can therefore exhibit non-stationary behaviour, in which its statistical properties vary

with time, as well as non-Gaussian features. In addition, detector data contain short-duration transient noise artefacts, commonly referred to as glitches, which can produce excess power over short time intervals and may resemble or obscure astrophysical GW signals [8, 9]. As a result, distinguishing genuine signals from noise fluctuations requires sophisticated data analysis techniques. These challenges highlight the necessity of advanced signal processing methods, such as matched filtering and statistical background estimation, which are essential for extracting weak GW signals from noisy detector data [10, 11].

The sensitivity of the interferometer depends on its ability to measure fractional changes in arm length, characterised by the strain $h \sim \Delta L/L$. Increasing the arm length enhances the detector response, as the induced displacement scales with L. A global network of ground-based interferometers has been developed to detect GWs. The main detectors include:

- Advanced LIGO: two 4 km interferometers located in Hanford and Livingston in the United States [6].
- Advanced Virgo: a 3 km interferometer located in Cascina, Italy [5].
- KAGRA: a 3 km underground interferometer located in Kamioka, Japan [12].
- GEO600: a 600 m interferometer located in Germany, primarily used for technology development [13].

These detectors operate collaboratively as part of an international network, including the LIGO Scientific Collaboration (LSC), the Virgo Collaboration, and the KAGRA Collaboration. Joint observations from multiple detectors improve source localisation through differences in the arrival time and detector response to the same signal, while also increasing confidence that a candidate is astrophysical rather than instrumental noise. The first direct detection of a GW, GW150914, was observed on

14 September 2015 and originated from the merger of a Binary Black Hole (BBH) system [10]. Since then, the number of detected events has increased substantially. The GWTC-4.0 catalogue, which includes observations from the first part of the Fourth Observing Run (O4)a, added 128 new significant candidates, bringing the total number of events in the catalogue to 218 [14].

1.3 Sources of Gravitational Waves

GWs are generated by systems with a time-varying mass quadrupole moment. As discussed in Subsec. 1.1.4, lower-order contributions such as monopole and dipole radiation are forbidden due to conservation laws, making the quadrupole term the leading source of GW emission. Astrophysical sources capable of producing detectable GW are therefore characterised by large masses undergoing highly accelerated, non-spherically symmetric motion. Compact binary systems are the most prominent examples, including Binary Neutron Star (BNS), BBH, and Neutron Star-Black Hole (NSBH) systems. As these systems inspiral and merge, they produce strong, time-dependent quadrupole radiation.

Other important sources include core-collapse supernovae, which can generate burst-like GW signals due to asymmetric mass dynamics [15], and rapidly rotating neutron stars, which may emit continuous GWs if they possess small deviations from axial symmetry [16]. Such continuous signals are expected to be intrinsically weak and have not yet been directly detected [17]. GW signals are commonly classified into four main categories: CBC, Continuous Waves (CWs), burst signals, and stochastic backgrounds. Each category is associated with distinct signal morphologies and astrophysical origins, enabling targeted search strategies in detector data [15].

1.3.1 Compact Binary Coalescence

CBCs are the primary class of GW sources detected by ground-based interferometers. These systems consist of two compact objects in a decaying orbit, including BBH, BNS, and NSBH systems. GW emission from

these systems arises due to the time-varying mass quadrupole moment of the orbiting bodies. As the system loses energy and angular momentum through gravitational radiation, the orbital separation decreases, leading to an inspiral phase in which the orbital frequency gradually increases. The waveform emitted during this process depends on several physical parameters, including the component masses, spins, orbital orientation, and distance to the source. In systems involving neutron stars, additional matter effects, such as tidal interactions, can introduce measurable deviations from the waveform predicted for BBH systems [18].

As the inspiral progresses, both the frequency and amplitude of the GW signal increase, producing a characteristic "chirp" signal. Ground-based detectors are sensitive to a finite frequency band, and therefore observe only the final stages of the inspiral, merger, and ringdown phases. The duration of the observed signal depends strongly on the masses of the binary components. Systems with lower masses, such as BNSs, spend more time within the sensitive frequency band of the detectors, producing signals that can remain within the detector band for several minutes. In contrast, higher-mass systems, such as BBHs, evolve more rapidly through the detector band and produce shorter-duration signals.

1.3.2 Other Gravitational-Wave Sources

In addition to CBCs, several other classes of GW sources are predicted, each characterised by distinct emission mechanisms and signal morphologies. Continuous GWs are expected from rapidly rotating neutron stars that possess a non-zero mass quadrupole moment. Such a quadrupole can arise from small deviations from perfect axial symmetry, for example, due to crustal deformations or internal magnetic stresses [16]. If the rotation of the neutron star is stable, the emitted GW signal is approximately monochromatic, with slowly evolving frequency and amplitude over long timescales [16, 19].

Burst GWs are transient signals associated with short-duration, poorly modelled astrophysical events, such as core-collapse supernovae or other highly dynamical processes involving asymmetric mass motion. These

signals are typically of short duration and lack precise waveform models [20].

Stochastic GW backgrounds arise from the superposition of many unresolved sources or from cosmological processes in the early universe [15]. These signals are characterised statistically rather than by individual waveforms, and may originate from a population of astrophysical sources or from primordial processes [15]. The detectability and potential overlap of these signals depend not only on the properties of the sources but also on the detectors' sensitivity and bandwidth. In this thesis, the focus is placed on CBCs, as these sources provide well-modelled signals suitable for the development and testing of early-warning detection methods.

1.4 Gravitational-Wave Data Analysis

The detection of CBC signals in ground-based detector data is challenging because the signals are often comparable to, or weaker than, the detector noise and must be identified within long stretches of continuously recorded strain data. Searches therefore rely on signal-processing and statistical methods designed to distinguish the characteristic inspiral, merger, and ringdown waveforms from instrumental and environmental noise. The following section introduces the main challenges involved in detecting and characterising CBC signals. Modern ground-based detectors produce continuous streams of data in which GW signals must be identified and extracted. Unlike many astronomical observations, GW detectors have a broad antenna response, requiring searches to be conducted over the entire sky. In addition, CBC signals evolve across a broad frequency range as the binary inspirals, requiring searches to track the signal evolution over time and frequency. A key feature of GW detection is the use of coherent analysis techniques that combine information from multiple detectors while accounting for the expected phase and time delays between them. Matched filtering further enhances detectability by accumulating Signal-to-Noise Ratio (SNR) over many waveform cycles while tracking the expected signal phase evolution. However, such approaches require accurate waveform models and extensive searches over parameter

spaces that can contain up to 15 dimensions for a generic quasi-circular CBC, including component masses, spins, sky location, distance, orientation, phase, and coalescence time [21].

The use of multiple detectors is essential for improving detection confidence and enabling PE. Joint observations allow for coincidence tests, improved sky localisation, and the measurement of physical properties such as source inclination [22]. Multi-detector networks also play a critical role in distinguishing true astrophysical signals from instrumental noise. Although hundreds of GW events have now been detected, individual signals can have amplitudes comparable to the detector noise, requiring sophisticated statistical techniques and robust noise modelling for reliable identification.

GW data analysis is also computationally demanding. In matched-filter searches for CBCs, the detector data are compared against a template bank, which is a collection of modelled waveforms covering a specified range of binary parameters. Candidate signals are identified from significant matches, ranked for statistical significance, and checked for consistency across detectors before PE is performed to infer the source properties [23, 24]. PE, in particular, involves exploring high-dimensional likelihood surfaces, often using stochastic sampling methods, making it one of the most computationally intensive stages of the analysis. Recent advances in computational hardware and software, including the use of Graphics Processing Units (GPUs) and Machine Learning (ML) techniques, are playing an increasingly important role in accelerating GW data analysis. In particular, ML approaches offer promising alternatives for low-latency detection [25]. This is especially relevant for early-warning detection of CBCs, where identifying a signal before merger can provide additional time for electromagnetic observatories to respond and begin follow-up observations. Developing methods capable of recognising low-SNR inspiral signals at low latency therefore forms a central motivation for the work presented in this thesis.

1.5 Matched Filtering

Matched filtering is an optimal linear filtering technique for detecting a signal of known form in stationary Gaussian noise, as it maximises the SNR under these assumptions [26]. The method consists of correlating the detector output with a set of template waveforms that model expected GW signals. The detector output $x(t)$ is assumed to consist of a signal $h(t)$ and noise $n(t)$

$$x(t) = n(t) + h(t - t_a), \tag{1.27}$$

where t_a denotes the signal arrival time. The Fourier transform of a time-domain quantity $x(t)$ is defined as

$$\tilde{x}(f) = \int_{-\infty}^{\infty} x(t)\, e^{-2\pi i f t}\, dt, \tag{1.28}$$

with the inverse Fourier transform given by

$$x(t) = \int_{-\infty}^{\infty} \tilde{x}(f)\, e^{2\pi i f t}\, df. \tag{1.29}$$

The correlation between the data and a template $q(t)$ can be expressed in the Fourier domain as

$$c(\tau) = \int_{-\infty}^{\infty} \tilde{q}^*(f)\, \tilde{x}(f)\, e^{2\pi i f \tau}\, df, \tag{1.30}$$

where τ represents the time lag. The optimal matched filter is obtained by maximising the SNR. In GW data analysis, the SNR is defined using a noise-weighted inner product

$$\langle x, q \rangle = 4 \int_0^{\infty} \frac{\tilde{x}^*(f)\tilde{q}(f)}{S_n(f)}\, df, \tag{1.31}$$

where $S_n(f)$ is the one-sided noise Power Spectral Density (PSD), which characterises the frequency-dependent noise in the detector. For stationary noise, the one-sided PSD is defined through

$$\langle \tilde{n}(f)\tilde{n}^*(f') \rangle = \frac{1}{2} S_n(f)\, \delta(f - f'), \tag{1.32}$$

where $\delta(f - f')$ is the Dirac delta function. Using this definition, the optimal SNR can be written as

$$\rho^2 = \langle h, h \rangle = 4 \int_0^\infty \frac{|\tilde{h}(f)|^2}{S_n(f)}\, df, \tag{1.33}$$

where ρ denotes the SNR. This expression shows that matched filtering naturally weights the signal by the inverse of the detector noise, giving greater importance to frequency regions where the detector is most sensitive. In practice, matched filtering requires searching over large template banks spanning the parameter space of possible signals, including masses and spins. For example, the GstLAL template bank used for CBC searches in O4 contains approximately 1.8×10^6 templates [27].

1.6 Practical Applications of Matched Filtering

Matched filtering is widely used in GW data analysis to detect signals with known waveform models. Its primary application is in searches for CBCs, where accurate theoretical waveforms are available.

1.6.1 Compact Binary Coalescence Searches

In matched-filter searches for CBC signals, a set of model waveforms, known as a template bank, is constructed to span the parameter space of possible signals. Each template corresponds to a specific combination of source parameters, such as component masses and spins [28]. The detector data are then filtered against this bank to identify signals with high SNR. Template banks used in standard CBC searches primarily span intrinsic parameters, such as the component masses and spins, which determine the waveform shape. Extrinsic parameters, such as sky location, distance, and orientation, affect how the signal is observed but are not generally dimensions of the template bank. For systems involving neutron stars, additional parameters, such as tidal deformability, may also play a role.

A template bank is designed to ensure that any true signal within the parameter space is sufficiently close to at least one template, such

that the loss in SNR remains below a chosen mismatch threshold, typically corresponding to a minimal match of about 0.97, or an SNR loss of approximately 3% [28]. This requires careful placement of templates across the parameter space, typically using a metric-based approach. The computational cost of matched filtering scales with the number of templates in the bank and the duration of the signals. As a result, searches for low-mass systems, which produce long-duration signals, are generally more computationally expensive than those for high-mass systems. Modern GW searches employ large template banks and advanced data analysis pipelines, as demonstrated in recent observing runs that have resulted in the detection of hundreds of compact binary mergers [7]. These analyses rely on high-performance computing infrastructure and continue to evolve with improvements in waveform modelling and computational techniques.

1.6.2 Continuous Wave Signals

CW searches target long-lived, nearly monochromatic GW signals, most commonly expected from rapidly rotating neutron stars with a non-axisymmetric mass distribution. In practice, the signal is not perfectly monochromatic, as its observed phase evolution is modulated by the source's intrinsic spin-down and by the detector's motion relative to the Solar System barycentre. Matched-filter searches for CW signals are computationally demanding because the signal may persist for months or years, while the search must span frequency, spin-down, and often sky position. For this reason, fully coherent all-sky searches over the full observing time are generally impractical, and modern analyses often adopt semi-coherent or hierarchical strategies that balance sensitivity against computational cost [17, 29].

Recent LIGO-Virgo-KAGRA (LVK) searches for CWs have applied multiple pipelines to Advanced LIGO and Virgo data from Third Observing Run (O3), including all-sky and directed searches for signals from isolated neutron stars and neutron stars in binary systems [17, 29, 30].

Einstein@Home, a volunteer distributed-computing project that uses donated computing resources to analyse astrophysical data, continues to play an important role in some of the deepest CW searches [31, 32]. Although no continuous GW signal has yet been confirmed, these searches place increasingly stringent upper limits on neutron-star ellipticity and related source properties, and they remain one of the most computationally intensive applications of matched filtering in GW astronomy [17, 29].

1.6.3 Background Estimation and Time-Slides

In GW searches, assessing the statistical significance of candidate events is a critical component of the analysis. Even for stationary Gaussian noise, the detection statistic has a non-zero background distribution, which must be characterised in order to assess the statistical significance of candidate events. Therefore, it is essential to estimate the background distribution of the statistic in order to determine the threshold corresponding to a given False Alarm Rate (FAR). In practice, detector noise is also non-Gaussian and non-stationary, introducing additional transient features that can further mimic true astrophysical signals, reinforcing the need for robust background estimation. A widely used technique for background estimation is the time-slides method [33, 34]. This approach involves introducing artificial time shifts between the data streams of different detectors. Since GW signals propagate at the speed of light, the maximum time delay between geographically separated detectors is on the order of tens of milliseconds. By applying time shifts much larger than the inter-site light-travel time, typically of order seconds or more, any true astrophysical signal is effectively removed from coincident analysis, while the statistical properties of the detector noise are preserved [23].

By repeating the analysis over many independent time shifts, a large number of noise-only coincidences can be generated. These are used to construct a background distribution of the detection statistic, which, in turn, allows the estimation of the FAR associated with candidate events

in the unshifted (zero-lag) data. This procedure provides a robust, data-driven method for quantifying the statistical significance of GW detections. The time-slides method is a standard component of some modern GW search pipelines, including PyCBC [23], and plays a central role in establishing detection confidence in CBC searches. Accurate background estimation is particularly important for low-latency and early-warning detection, where rapid identification of candidate events must be balanced against the risk of false alarms. While matched filtering enables the detection of candidate signals, further analysis is required to infer the source's physical properties.

1.7 Parameter Estimation

PE in GW astronomy aims to infer the physical properties of astrophysical sources from detector data. These parameters include intrinsic quantities such as component masses and spins, as well as extrinsic parameters such as sky location, distance, and orbital orientation. Modern GW PE is performed within a Bayesian inference framework. In this approach, the posterior probability distribution of the source parameters is computed by combining a likelihood function, which quantifies the agreement between the data and a waveform model, with prior information about the parameters.

Bayesian methods have been the standard approach for GW PE over a decade and are routinely applied to all CBC detections reported by the LVK collaboration [35]. These analyses provide full posterior distributions rather than single-point estimates, allowing for robust uncertainty quantification and the study of parameter correlations. The PE problem is computationally challenging due to the high dimensionality of the parameter space and the cost of generating accurate waveform models. Sampling techniques such as Markov Chain Monte Carlo (MCMC) and nested sampling are typically employed to explore the posterior distribution. Despite these challenges, PE plays a central role in GW science, enabling precise measurements of source properties and tests of general relativity in the strong-field regime.

1.8 Bayesian Inference

Bayesian inference provides a principled framework for estimating the parameters of GW sources by combining prior knowledge with observational data. The foundation of this approach is Bayes' theorem, which relates the posterior probability distribution of a set of parameters θ given data d to the likelihood and prior

$$P(\theta|d) = \frac{P(d|\theta), P(\theta)}{P(d)}. \tag{1.34}$$

Here, $P(\theta|d)$ is the posterior distribution, $P(d|\theta)$ is the likelihood, $P(\theta)$ is the prior, and $P(d)$ is the evidence, which acts as a normalisation constant. In GW data analysis, the likelihood function quantifies how well a waveform model, specified by parameters θ, matches the detector data. The prior encodes our assumptions about the possible values of these parameters, such as physically allowed mass ranges or spin distributions.

Bayesian inference has been the standard approach for PE in CBC analyses for over a decade. It enables the full posterior distribution of source parameters to be computed, allowing for robust uncertainty quantification and the study of parameter correlations. This approach is routinely applied in LVK analyses of detected signals [35]. Evaluating the posterior distribution is computationally demanding due to the high dimensionality of the parameter space and the cost of waveform generation. To address this, stochastic sampling techniques such as MCMC and nested sampling are widely used. These methods efficiently explore the parameter space and generate samples from the posterior distribution. More recently, ML approaches have been investigated as a means of accelerating PE, particularly in low-latency or real-time applications. These methods aim to approximate the mapping between detector data and source parameters, reducing the computational cost associated with traditional sampling techniques.

1.9 Multi-Messenger Astronomy

MMA represents a transformative approach to studying the universe by combining information from different cosmic messengers GWs, EM

radiation, neutrinos, and cosmic rays [36]. Unlike traditional astronomy, which relies primarily on light, MMA provides a more comprehensive understanding of astrophysical phenomena by capturing signals arising from diverse physical processes [37]. The coordinated detection of these messengers enables scientists to probe extreme environments, test fundamental physics, and uncover the origins of the most energetic events in the cosmos [38]. GWs play a central role in MMA, particularly in the study of compact-object mergers involving neutron stars, such as BNS and NSBH systems [36]. These sources can emit both GWs and EM radiation across multiple wavelengths, making them ideal targets for multi-messenger observations [36]. Observing these signals together provides deeper insight into the merger physics, the remnant properties, and the environment in which the event occurs [39].

Detecting and localising GW events in low-latency is essential for enabling timely multi-messenger follow-up [40]. Many EM counterparts, such as kilonovae and short Gamma-Ray Bursts (GRBs), evolve on short timescales and can fade within hours to days [41]. Rapid alerts from GW detectors allow telescopes and observatories to be directed quickly to the right region of the sky, increasing the chances of observing transient emissions [42]. The importance of coordinated multi-messenger observations was clearly demonstrated by GW170817. A short GRB was detected approximately 1.7 seconds after the merger [43]. The joint detection of the GW and gamma-ray signals motivated extensive follow-up observations across the electromagnetic spectrum, leading to the identification of the optical counterpart and subsequent multi-wavelength observations [44].

Following a low-latency GW candidate, an alert is distributed to the astronomical community through public alert systems such as the Gamma-ray Coordinates Network (GCN). These alerts provide information required for follow-up observations, including the event time, source classification, estimated distance, and a sky-localisation probability map, which allows electromagnetic observatories to prioritise the most probable regions of the sky for rapid observations [45]. The response time depends on factors such as the observatory field of view, source visibility,

localisation uncertainty, and observing strategy. Follow-up observations may therefore begin within seconds or minutes of an alert and continue over subsequent hours, days, or longer.

The subsequent multi-wavelength observations of GW170817 further demonstrate the value of rapid EM follow-up. The optical counterpart AT 2017gfo was identified approximately 11 hours after the merger, with optical and near-infrared emission evolving over the following days. These observations were consistent with kilonova emission powered by radioactive decay in neutron-rich merger ejecta [46, 47]. X-ray and radio counterparts were detected on later timescales, approximately 9 and 16 days after the merger, respectively, and remained observable for substantially longer periods [46]. The different wavelength regimes provide complementary information: optical and near-infrared observations constrain the properties and composition of the ejecta and the production of heavy elements, while X-ray and radio observations probe the relativistic outflow, its interaction with the surrounding medium, and the geometry of the post-merger jet [47, 48].

To support real-time follow-up, GW observatories such as LIGO and Virgo operate low-latency analysis pipelines, including GstLAL [49], PyCBC Live [50], and Summed Parallel Infinite Impulse Response (SPIIR) [51, 52]. These systems continuously analyse incoming data and can issue alerts within seconds to minutes of a signal entering the detectors' sensitive band [49]. In specific cases, such as long inspiral signals from BNS mergers, these pipelines have demonstrated limited early-warning capability, issuing alerts just a few seconds before the final merger [53]. Such pre-merger alerts, although rare and modest in lead time, represent an important step toward enabling proactive EM follow-up [54]. Nevertheless, significant challenges remain in achieving reliable early-warning detection [55]. Signals in the early inspiral phase are typically weak and easily masked by detector noise, making confident detection difficult without increasing false alarms [49]. Moreover, early alerts often lack precise sky localisation and source classification, which are crucial for meaningful follow-up [56]. Advancing early-warning capabilities will require more

sensitive and predictive methods that can identify faint pre-merger signals in real time [57]. Improving these systems is critical for expanding the reach and effectiveness of MMA in future observing runs [7]. The scientific value of early-warning detection is strongly linked to the response time of electromagnetic observatories. For binary neutron star mergers, warnings of tens of seconds can already be astrophysically useful, as sufficiently nearby systems may be detected approximately 10–60 s before merger, potentially allowing wide-field telescopes to begin observations before or immediately after coalescence [53]. This timescale is also comparable to the response capability of rapid robotic facilities. For example, GOTO has demonstrated response times below one minute for GW alerts under favourable observing conditions, with a fastest recorded response of 28 s [58].

1.10 Summary

This chapter provides the theoretical and methodological foundation for the development of GW detection techniques. It begins by introducing the theory of GWs within the framework of general relativity, using the weak-field approximation to show that spacetime perturbations propagate as waves travelling at the speed of light. This establishes the physical origin of GW signals and their key properties, including their polarisation states and quadrupolar emission, which are essential for understanding the signals targeted in this work. The chapter then outlines the principles of GW detection using ground-based interferometers, highlighting the challenges associated with measuring extremely small strain amplitudes. This motivates the need for advanced data analysis techniques capable of extracting weak signals from noisy data.

Matched filtering is introduced as the primary method for detecting CBC signals in current GW searches. The concepts of template banks and the computational challenges of searching large parameter spaces are discussed, as they form the basis for modern detection pipelines. In addition, the time-slides method is presented as a key technique for estimating background and assessing the statistical significance of candidate events,

both of which are crucial for reliable detection. While PE is briefly discussed to provide context, the emphasis of this work is on detection rather than inference. Modern GW analyses rely on Bayesian methods to characterise detected signals; however, these techniques are computationally intensive and are typically applied only after a confident detection. Finally, the chapter situates GW detection within the broader context of MMA, highlighting the importance of low-latency detection for enabling EM follow-up observations. This directly motivates the need for rapid and efficient detection algorithms, which is the primary focus of this thesis.

CHAPTER 2

Introduction to Machine Learning

This chapter serves as a comprehensive introduction to the foundations and practical aspects of ML, beginning with a clear definition of the field and an exploration of its main learning paradigms: supervised, unsupervised, semi-supervised, and reinforcement learning. We then introduce essential concepts that underpin ML systems. The chapter proceeds to explain the need for robust model evaluation by discussing the fundamental bias-variance trade-off, alongside the important distinction between underfitting and overfitting. These concepts help diagnose common modelling issues and guide improvements. When models fall short of expectations, enhancements can often be achieved through Exploratory Data Analysis (EDA), feature engineering, or hyperparameter tuning—key strategies that contribute significantly to model accuracy and robustness.

Building on these foundational ideas, the chapter transitions to deep learning, a subset of ML that has revolutionised fields such as computer vision and natural language processing. We explain the key components of deep learning architectures, including activation functions, loss functions, and optimizers, which together enable neural networks to learn complex patterns. Sec. 2.3 delves into Recurrent Neural Networks (RNNs), a specialized class of deep learning models designed for sequential data. Here, we explore the structure and purpose of advanced variants, specifically Long Short-Term Memory (LSTM) networks, and discuss their strengths in capturing temporal dependencies. We also explore the challenges inherent in training RNNs and highlight state-of-the-art advancements that have pushed the boundaries of what these models can achieve. By the end of this chapter, readers will have a well-rounded understanding of ML and deep learning fundamentals, and will be equipped with the conceptual and practical tools needed to begin building and evaluating intelligent

systems.

2.1 Machine Learning Key Concepts

The roots of ML trace back to classical mathematics, statistics, and computer science. While the field has rapidly evolved in recent decades, its foundational ideas are deeply tied to concepts in linear algebra, calculus, probability theory, and particularly inferential statistics—the study of making predictions and generalisations from data [59]. At its core, ML is about building models that can learn from data and generalise to unseen situations. This mirrors the objectives of traditional statistical inference, in which models are designed to estimate underlying patterns or distributions and to draw conclusions about a population based on sample data. For example, techniques such as linear regression, which originated in statistics to model relationships between variables, are now fundamental to supervised learning [59].

Furthermore, the Bayesian framework, a statistical approach that updates the probability of a hypothesis as more evidence becomes available, is a cornerstone of many ML methods. It provides a principled way to incorporate prior knowledge and handle uncertainty, which is especially crucial in fields like physics where data can be noisy or incomplete. Advancements in computational power, data availability, and algorithmic development have transformed these mathematical and statistical tools into scalable ML systems. Modern ML techniques such as support vector machines, decision trees, and neural networks build on these early foundations, extending them into complex, high-dimensional problem spaces. Thus, while ML is often seen as a modern technological breakthrough, its essence lies in the long-standing mathematical pursuit of understanding data, modelling uncertainty, and making predictions—an endeavour shared deeply with the scientific method itself [59].

In this context, the application of ML in GW and astrophysical research provides a clear illustration of this continuity between mathematics, data, and discovery. At the most basic level, supervised learning methods have been employed for tasks such as classifying noise transients

in detector data and distinguishing true GW signals from spurious instrumental artifacts [9, 60]. Beyond noise characterisation, machine-learning approaches have been explored for estimating key parameters of compact binary mergers more efficiently than traditional Bayesian inference methods [61]. More sophisticated deep learning architectures, such as Convolutional Neural Networks (CNNs), have been proposed as alternative search methods for CBC signals directly from raw time series data, potentially enabling rapid identification without the need for handcrafted templates, although published GW detections to date still rely on established matched-filter search pipelines [62, 63]. At the frontier, advanced generative models and physics-informed neural networks are now being developed to accelerate waveform modelling and simulate astrophysical populations [64, 65].

RNNs and LSTMs have been widely applied to sequential-data problems outside GW astronomy, including speech and language processing, time-series forecasting, and biological sequence modelling [66, 67]. Within GW astronomy, deep-learning approaches have been explored for tasks such as rapid signal detection and classification [62]; however, prior early-warning studies have primarily focused on other deep-learning architectures. The use of an LSTM network specifically for early-warning detection of GW signals forms a novel contribution of this thesis.

2.1.1 Types of Learning

ML builds systems that learn from data rather than relying solely on explicit programming. Over time, several fundamental learning paradigms have emerged, each suited to different types of problems and data scenarios. The main types of learning are supervised, unsupervised, semi-supervised, and reinforcement learning.

Supervised learning involves training a model on labelled data, where each input is associated with a target output. The model learns to predict outputs from inputs, either for discrete labels (classification) or continuous values (regression), by minimising the prediction error [68].

Unsupervised learning, in contrast, operates on unlabelled data. The model identifies underlying patterns or structures without explicit targets [68]. Common approaches include clustering algorithms and dimensionality reduction techniques such as Principal Component Analysis (PCA) [68].

Semi-supervised learning combines both approaches by using a small amount of labelled data alongside a larger set of unlabelled data. This is particularly useful when labelling data is costly, as in image classification tasks.

Reinforcement learning focuses on decision-making: an agent interacts with an environment to maximise cumulative reward. Unlike supervised learning, the correct output is not provided; instead, the agent learns through trial and error based on feedback from its actions. This process is often modelled using a Markov Decision Process (MDP), where future states depend only on the current state and action [68].

2.1.2 Learning Quality

Evaluating model performance is essential to determine how well a ML model generalises beyond its training data. The choice of evaluation metric depends on the nature of the task (e.g., classification or regression) and the practical cost of different types of errors [69]. In imbalanced datasets, metrics such as precision, recall, and F1 score are often more informative than accuracy. Precision measures the proportion of predicted positive cases that are correct, recall measures the proportion of actual positive cases that are correctly identified, and the F1 score provides a balance between precision and recall [70]. For supervised learning, classification performance is commonly assessed using accuracy, precision, recall, F1 score, and the Receiver Operating Characteristic (ROC)-Area Under the Curve (AUC), which captures performance across different decision thresholds [70, 71]. For regression tasks, standard metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and the coefficient of determination R^2, which quantify prediction error and goodness of fit [72].

In unsupervised learning, evaluation is more challenging due to the absence of ground-truth labels. Instead, internal validation metrics such as the Silhouette score, Davies-Bouldin Index (DBI), and Calinski-Harabasz Index (CHI) are used to assess clustering quality by measuring intra-cluster cohesion and inter-cluster separation [73–75]. In addition, dimensionality reduction techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP) provide qualitative insight into the structure of high-dimensional data [76]. Overall, appropriate evaluation strategies are crucial to ensure that models capture meaningful patterns and generalise reliably to unseen data.

2.1.3 Prediction Challenges

Evaluation metrics not only assess model performance but also reveal potential issues in model behaviour. High performance on training data may indicate overfitting, where the model captures noise rather than meaningful patterns, while consistently low performance may suggest underfitting due to insufficient model complexity. Overfitting occurs when a model is overly complex, leading to excellent performance on training data but poor generalisation to unseen data. In practice, overfitting can be identified by comparing training and validation performance: if the training error continues to decrease while the validation error begins to increase, the model is likely fitting noise or dataset-specific features rather than learning patterns that generalise. This is typically associated with low bias and high variance. In contrast, underfitting arises when the model is too simple to capture the underlying structure of the data, resulting in poor performance on both training and test sets, and is associated with high bias and low variance [77].

Bias refers to the systematic error introduced by overly simple model assumptions, whereas variance describes the sensitivity of a model to variations in the training data. A high-bias model tends to underfit because it cannot capture the underlying structure of the data, while a high-

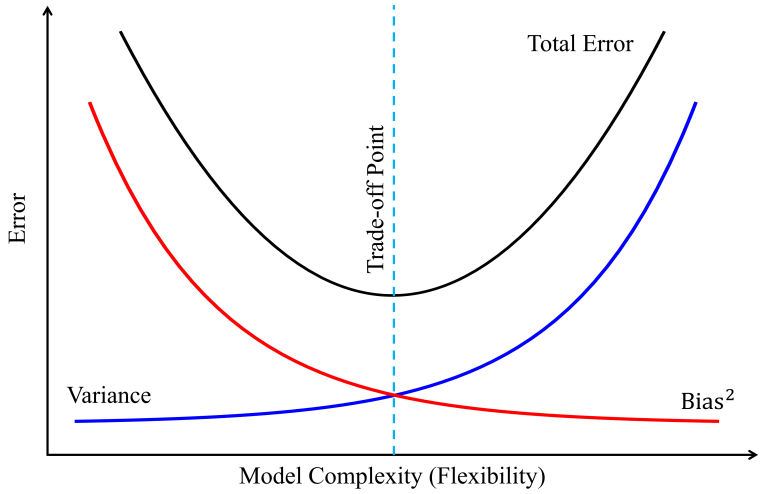


Figure 2.1: Illustration of how model complexity affects bias, variance, and total error in supervised learning models. Bias starts high on the left (e.g., simple models like linear regression) and decreases as complexity increases. The trade-off point is the best generalisation of unseen data. Adapted from [79]

variance model may fit the training data closely but perform poorly on unseen data.

These challenges are commonly understood through the bias–variance trade-off, which describes the balance between model complexity and generalisation performance. As model complexity increases, bias decreases while variance increases, and an optimal balance must be achieved to minimise overall error [78–80] (Fig. 2.1).

2.1.4 Enhancing Models

Model improvement typically begins with diagnosing performance issues identified during evaluation. Overfitting may indicate the need for regularisation, additional data, or reduced model complexity, while underfitting may require more expressive models or improved feature representations [81]. A key aspect of model improvement lies in understanding and refining the input features. This is achieved through EDA, which helps identify patterns, anomalies, and relationships within the data, and

through feature engineering, which transforms raw inputs into more informative representations. These steps are essential for improving model accuracy and generalisation.

Feature selection further enhances performance by retaining only the most relevant input variables, reducing dimensionality and mitigating overfitting [82]. This leads to more efficient models with improved interpretability. In addition, hyperparameter tuning plays a crucial role in optimising model performance. Techniques such as grid search, random search, and gradient-based optimisation are commonly used to identify suitable parameter configurations [83, 84]. While these strategies can significantly improve performance, they often increase computational cost. In practice, more expressive models, such as deep neural networks, can sometimes reduce the need for extensive manual feature design by learning hierarchical representations directly from data.

2.2 Neural Network

Among ML methods, neural networks have emerged as powerful models capable of capturing complex relationships in data through layered computational structures inspired by biological neurons. A neural network is composed of neurons (nodes) organised in layers (input, hidden, and output). It can be shallow (e.g., with only one hidden layer) or deep (with many hidden layers). Deep learning focuses on large-scale neural networks that automatically learn features from large datasets, often requiring significant computational resources, unlike traditional ML methods that rely on manual feature engineering. This section presents neural network architectures for various tasks, along with key concepts underlying the learning process, including activation functions, gradient descent, and backpropagation.

2.2.1 Neural Network Architectures

Neural network architectures differ in how layers are structured and how information flows through them, leading to distinct capabilities across

tasks. Feedforward Neural Networks (FNNs) form the foundational architecture, while more specialised models such as CNNs and RNNs are designed for structured data, including images and sequences [85]. In FNNs, information flows in one direction from the input to the output, without feedback connections. These models are widely used for classification and regression tasks, such as image classification and predicting continuous numerical values, and are computationally efficient, but they are not well-suited for sequential or spatial data [86].

CNNs are designed to capture spatial hierarchies through convolutional operations, making them highly effective for image and video analysis. Their strength lies in parameter efficiency and their ability to learn local patterns directly from data [87]. RNNs, in contrast, incorporate recurrent connections that allow information to persist across time steps, enabling the modelling of sequential data such as time series and language. However, they can suffer from training instabilities, including vanishing gradients, which motivated the development of advanced variants such as LSTM and Gated Recurrent Unit (GRU) [88]. Overall, selecting an appropriate architecture is crucial for achieving strong performance and generalisation, particularly when dealing with structured or sequential data.

2.2.2 Perceptron

The perceptron is a fundamental building block of Artificial Neural Networks (ANNs), representing a single-layer feedforward model capable of binary classification [89]. It models an artificial neuron that computes a weighted sum of its inputs, followed by applying a non-linear activation function [90]. During training, the weights are adjusted to minimise a loss function, which quantifies the difference between predicted and true outputs, typically using optimisation methods such as gradient descent [91]. Mathematically, the perceptron is defined as:

$$y = \sigma(w \cdot x + b), \tag{2.1}$$

where x represents the input vector, w the corresponding weights, b the bias term, and σ the activation function.

2.2.3 Activation Functions

Activation functions introduce non-linearity into neural networks, enabling them to learn complex relationships beyond linear mappings. They are essential for effective training, as gradient-based optimisation methods such as backpropagation rely on differentiable functions. Early activation functions, such as the step function, are limited by their non-differentiability and are therefore unsuitable for modern deep learning. Similarly, linear activation functions do not increase model expressiveness when stacked, as they cannot capture non-linear patterns.

Nonlinear activation functions such as the sigmoid and Hyperbolic Tangent (Tanh) were widely used in early neural networks because they provide smooth, differentiable nonlinear transformations that enable gradient-based optimisation (Fig. 2.2). While these functions enable probabilistic interpretation and smooth gradients, they suffer from the Vanishing Gradient Problem (VGP), which limits their effectiveness in deep architectures. In contrast, the Rectified Linear Unit (ReLU) and its variants, such as Leaky ReLU, are now widely adopted in deep learning because they are computationally simple and maintain stronger gradients for positive inputs, helping to mitigate the VGP and enable faster convergence during training (Fig. 2.3). Overall, the choice of activation function significantly impacts training stability and model performance, particularly in deep neural networks.

2.2.4 Multi-Layer Perceptron

A Multi-Layer Perceptron (MLP) is an FNN composed of multiple layers of interconnected neurons, extending the single-layer perceptron to model complex, non-linear relationships [92]. It consists of an input layer, one or more hidden layers, and an output layer, where each layer applies a linear transformation followed by a non-linear activation function (Fig. 2.4).

The connections between layers can be efficiently represented using matrix operations. For a given layer, the network input is computed as:

$$\text{net} = W \cdot x + b, \tag{2.2}$$

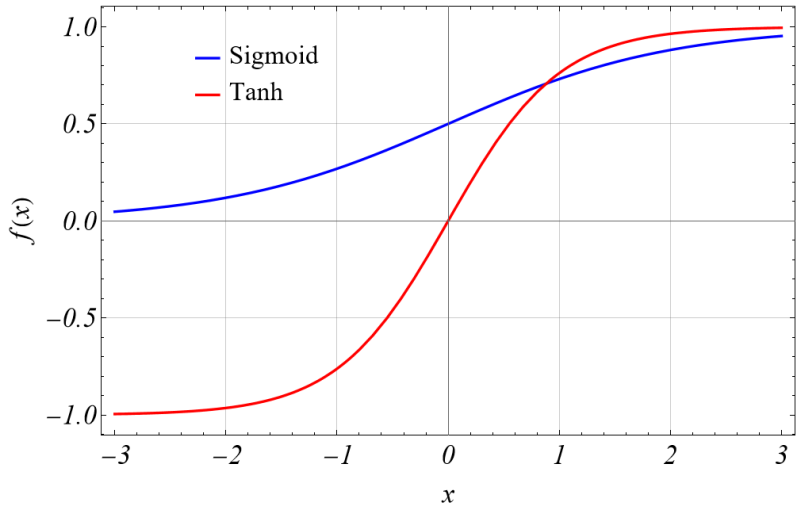


Figure 2.2: Sigmoid and Hyperbolic Tangent (Tanh) functions comparison.

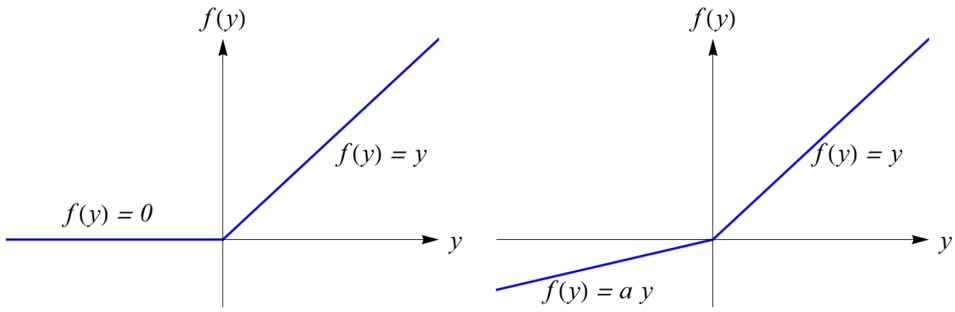


Figure 2.3: Rectified Linear Unit (ReLU) and Leaky ReLU functions comparison.

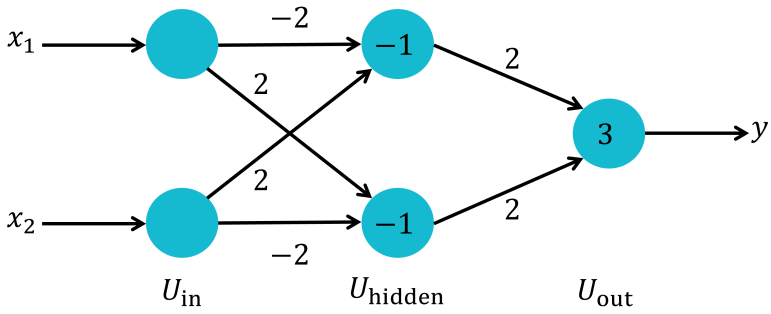


Figure 2.4: Three-layer perceptron architecture. The numerical values shown on the connections represent the weights w_{uv}, which scale the input signals and quantify the strength and sign of the influence between neurons in adjacent layers.

where x represents the input vector, W the weight matrix, and b the bias vector. This formulation enables efficient forward propagation through the network and serves as the basis for backpropagation-based training. MLPs serve as the foundation of many modern neural network architectures and remain widely used for a variety of supervised learning tasks.

2.2.5 Logistic Regression

Logistic regression is a fundamental supervised learning algorithm used for binary classification tasks [92]. Unlike linear regression, which predicts continuous values, logistic regression models the probability that a given input belongs to a particular class. The model applies a linear transformation to the input features, followed by a sigmoid activation function:

$$y = \frac{1}{1 + e^{-(w \cdot x + b)}}, \tag{2.3}$$

where w represents the weight vector, x the input features, and b the bias term. The output y lies in the range $(0, 1)$ and can be interpreted as a probability.

Logistic regression can be viewed as a simple neural network with no hidden layers, making it closely related to the perceptron model. During training, the model parameters are optimised by minimising a suitable loss

function, typically the cross-entropy loss, using gradient-based optimisation methods. Although logistic regression is limited to linearly separable problems, it provides a foundational framework for understanding more complex neural networks and serves as a stepping stone towards training deeper architectures using gradient descent.

2.2.6 Gradient Descent

Gradient descent is a fundamental optimisation method used to train neural networks by iteratively updating model parameters, such as weights and biases, in the direction that minimises a chosen loss function. In the context of MLPs, this requires differentiable activation functions so that gradients can be computed efficiently during training [93]. The gradient of the objective function indicates the direction of steepest increase, and gradient descent updates the parameters in the opposite direction:

$$u^{(i+1)} = u^{(i)} - \eta \nabla_u f(u)\Big|_{u^{(i)}}, \tag{2.4}$$

where η is the learning rate and $\nabla_u f(u)$ is the gradient of the objective function with respect to the parameter vector u.

The choice of learning rate is critical. If it is too small, convergence may be slow; if it is too large, the optimisation process may become unstable or fail to converge. In practice, gradient descent may converge to local minima or saddle points, depending on the structure of the loss landscape [94]. To improve computational efficiency, neural networks are usually trained using stochastic or mini-batch variants of gradient descent rather than full-batch updates. In Stochastic Gradient Descent (SGD), parameters are updated using individual training examples, while mini-batch training uses small subsets of the data, providing a compromise between computational efficiency and training stability [93]. Training proceeds over multiple epochs, where one epoch corresponds to a complete pass through the training dataset. Monitoring performance across epochs helps assess convergence and diagnose issues such as underfitting and overfitting [95, 96].

2.2.7 Gradient Descent Variants

Standard gradient descent can suffer from slow convergence, sensitivity to the learning rate, and difficulty navigating complex loss landscapes. To address these limitations, several variants have been developed. A commonly used extension is momentum, which accelerates optimisation by incorporating a fraction of the previous update into the current step:

$$\Delta w_t = -\eta \nabla_w e\big|_{w_t} + \alpha \Delta w_{t-1}, \tag{2.5}$$

where α is the momentum coefficient. This helps smooth noisy updates and reduces oscillatory behaviour during training [97, 98].

Adaptive optimisation methods further improve convergence by adjusting the learning rate for each parameter. Root Mean Squared Propagation (RMSProp) achieves this by maintaining an exponentially decaying average of squared gradients:

$$\Delta w_t = -\eta \frac{\nabla_w e\big|_{w_t}}{\sqrt{v_t} + \epsilon}, \tag{2.6}$$

where v_t is a moving average of past squared gradients [99]. One of the most widely used optimisation algorithms is Adaptive Moment Estimation (Adam), which combines the ideas of momentum and adaptive learning rates:

$$\Delta w_t = -\eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \tag{2.7}$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments of the gradients [93, 100].

In addition, weight decay is often used as a regularisation technique to discourage excessively large weights and improve generalisation:

$$\Delta w(t) = -\eta \nabla_w e\big|_{w_t} - \xi w(t), \tag{2.8}$$

where ξ controls the strength of the penalty [101]. These variants play a central role in modern neural network training by improving optimisation stability, convergence speed, and generalisation performance.

2.2.8 Backpropagation

The backpropagation algorithm, introduced by [98], is a fundamental method for training ANNs. It efficiently computes the gradients of the loss function with respect to the network parameters using the chain rule of calculus. These gradients are then used to update the weights through gradient descent. Given an input, the network performs a forward pass to compute the predicted output $\hat{y}$ and the corresponding loss $\mathcal{L}(\hat{y}, y)$. Backpropagation then computes error signals starting from the output layer and propagates them backward through the network. For the output layer, the error term depends on the derivative of the loss function and the activation function. For hidden layers, the error is computed recursively based on the weighted sum of errors from subsequent layers.

Once these error terms are obtained, the gradients of the loss with respect to each weight can be computed, allowing parameter updates using optimisation methods such as SGD. This process enables neural networks to learn complex mappings between inputs and outputs by iteratively minimising the loss function. While highly effective, backpropagation relies on differentiable activation functions and may encounter challenges such as vanishing gradients in deep architectures. The concepts of backpropagation and gradient descent together form the foundation of neural network training. However, standard feedforward architectures are limited in their ability to capture temporal dependencies, which motivates the introduction of RNNs in the following section.

2.3 Recurrent Neural Networks

As discussed in the previous section, when using gradient descent to train parameters in a sequence of functions, the algorithm depends on the chain rule. This often results in parameters that are farther from the output being updated more slowly. This slowdown occurs because activation functions tend to shrink gradients as they propagate backward, leading to vanishing gradients across many layers. In RNNs, which process sequential data by propagating information across time steps, this issue becomes even more pronounced. Training involves Backpropagation

Through Time (BPTT), which can create very long computational chains, making it difficult for RNNs to learn long-term dependencies effectively. When an event from the distant past is expected to affect the current output, its impact may not be adequately captured. This occurs because, by the time the gradient propagates back to the point where the event occurred, it may have weakened so much that it nearly vanishes, making it difficult to update the corresponding parameters.

Unlike feedforward networks, RNNs incorporate feedback connections, allowing them to maintain a memory of previous inputs. This internal state enables them to capture information from sequences with potentially unlimited context. While historically challenging due to their complexity and large number of parameters, recent advances in architecture design, optimisation methods, and parallel computing have made large-scale training of RNNs more feasible and effective. Early approaches to sequence modelling attempted to incorporate temporal context using fixed-size sliding windows [102]. While effective for capturing local dependencies, such methods are limited in modelling long-range relationships across sequences. Architectures such as LSTMs and Bidirectional RNNs have demonstrated strong performance in a variety of sequence modelling tasks, including image captioning, language translation, and handwriting recognition [103, 104].

In contrast, traditional neural networks face significant limitations in similar tasks. One key drawback is their assumption that training and test examples are independent. As a result, once an example is processed, the network's internal state is reset. While this is acceptable for independently generated data, it fails for temporally or spatially correlated inputs—such as video frames, audio segments, or words in a sentence, where context is essential. Furthermore, standard networks typically require inputs to be fixed-length vectors, making them ill-suited for modeling data with sequential or variable-length structures. To address these challenges, it is critical to extend neural networks to effectively handle temporally dependent data, especially in domains where they already represent the state-of-the-art [105].

RNNs are neural networks designed to process sequential data by maintaining information across time steps. This enables them to effectively model sequences where elements are not independent, capturing both input and output dependencies. Moreover, RNNs are well-suited to handling temporal and sequential relationships across multiple time scales. Any RNN architecture defined by its nodes, connections, and activation functions is fully differentiable from end to end. This allows for the computation of gradients of the loss function with respect to all model parameters, making RNNs compatible with gradient-based training methods. Additionally, RNNs can be regularized using well-established techniques such as weight decay, dropout, and constraints on model complexity, which help mitigate the risk of overfitting [105].

RNNs are designed to handle sequences, where the input, output, or both may be sequential. These sequences, which can be finite or countably infinite, are typically represented as ordered vectors indexed by time, such as $(x^{(1)}, x^{(2)}, ..., x^{(T)})$ for inputs and $(y^{(1)}, y^{(2)}, ..., y^{(T)})$ for targets. A training set usually consists of pairs of input and target sequences. While RNNs are often applied to temporal data like audio or video, they are equally effective for non-temporal sequences such as genetic data or natural language [67], where the order of elements matters even without a clear notion of time. In temporal terms, input data points $x^{(t)}$ arrive at discrete time steps t, which may correspond to evenly spaced samples from a continuous process or reflect an ordinal order. RNNs are thus versatile, capable of modeling both time-based and order-based sequences.

RNNs extend FNNs by introducing recurrent connections that link nodes across adjacent time steps, enabling the modelling of temporal sequences. Unlike standard feedforward connections, recurrent connections can form cycles, such as self-loops, spanning time. At each time step t, nodes receive input from both the current input $x^{(t)}$ and the previous hidden state $h^{(t-1)}$, allowing past information to influence current and future outputs $\hat{y}^{(t)}$:

$$h^{(t)} = \sigma\left(W^{hx}x^{(t)} + W^{hh}h^{(t-1)} + b_h\right), \tag{2.9}$$

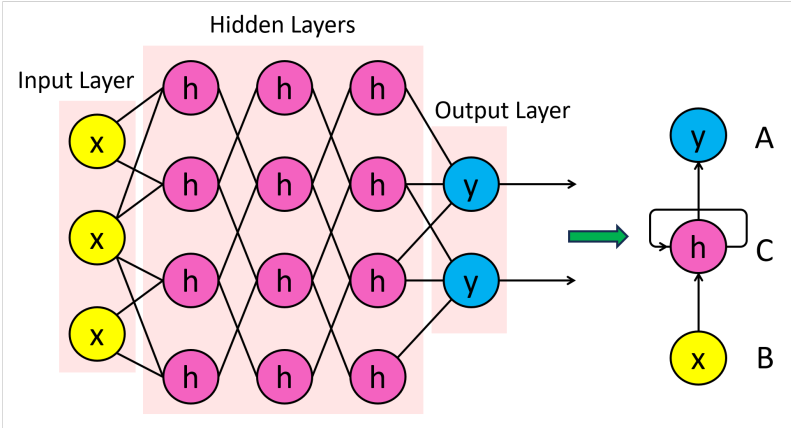


Figure 2.5: Illustration of a neural network architecture and its recurrent formulation. The left panel depicts a multi-layer network with several hidden units, while the right panel presents a simplified RNN with a single input (x), a hidden unit (h), and an output (y). The recurrent structure allows the hidden state to carry information forward through time.

and

$$\hat{y}^{(t)} = \text{softmax}(W^{yh} h^{(t)} + b_y). \tag{2.10}$$

The network's behaviour at each step is governed by two core equations involving weights from the input to the hidden layer (W^{hx}), the hidden-to-hidden recurrent weights (W^{hh}), and bias terms (b_h, b_y). This structure enables training via BPTT, a method introduced by [106], and now standard in training RNNs (Fig. 2.5).

Training RNNs has historically been challenging, largely because of the difficulty of learning long-term dependencies. Even in feedforward networks, optimisation is computationally very difficult [107], and RNNs exacerbate this issue due to vanishing and exploding gradients, as explained by [108] and later extended by [109]. These issues arise when gradients propagated backward through time either vanish or explode exponentially as the temporal distance between relevant inputs and outputs increases. For example, in a minimal RNN with a single recurrent hidden node, an input at time τ and an error computed at time t lead to exponential sensitivity based on the recurrent weight w. If $|w| < 1$, gradients

vanish; if $|w| > 1$, they explode. This behaviour depends not only on the magnitude of the weight but also on the activation function. Sigmoid functions typically aggravate vanishing gradients, while ReLU can lead to exploding ones. Pascanu et al. [110] provided a rigorous mathematical treatment, identified specific conditions under which these problems occur, and proposed regularisation to constrain weights to a stable regime.

A workaround proposed by [111], known as Truncated BPTT, restricts the number of time steps over which errors are backpropagated. This approach helps alleviate exploding gradients but compromises the network's ability to capture long-term dependencies. The LSTM architecture later emerged as a direct response to these challenges, employing specially designed recurrent units with fixed unit weights to preserve gradient flow over time. The most influential RNNs architectures for sequence learning originate from a landmark paper published in 1997. LSTM by [103] introduced the memory cell—a novel computational unit that replaces standard hidden nodes in RNNs. These memory cells address the VGP, enabling more stable training and improved handling of long-term dependencies. A key innovation is the self-connected recurrent edge with a fixed weight of one within each memory cell, which allows gradients to flow across many time steps without vanishing or exploding [105].

2.3.1 Long Short-Term Memory

The term LSTM highlights the model's ability to manage information across different time scales. Long-term memory is encoded in the network's weights and represented by the cell state, which is gradually updated during training and serves as a mechanism for retaining relevant information throughout the sequence. In contrast, short-term memory is stored in the hidden state, which reflects the network's current activation and captures information about what has been processed up to a given time step t. The LSTM structure introduced by [103] enables the error signal to flow consistently across many time steps without diminishing.

The key concept is to treat the cell state as a quantity that can be carried forward through time with minimal modification. In the absence of new input, the cell state is propagated unchanged, effectively applying an identity mapping with a derivative of one. This mechanism, known as the Constant Error Carousel, ensures that gradients can propagate backward through time without vanishing. Instead of recomputing the activation solely from new inputs, the model learns how to selectively update the existing cell state, allowing information to persist over long sequences [103].

Each memory cell comprises multiple components, including multiplicative gates that regulate the flow of information over time. This internal structure enables the LSTM to selectively update, forget, and output parts of its state, making it particularly effective for modelling long-range dependencies in sequential data. The structure of an LSTM cell is illustrated in Fig. 2.6. Together, these components enable LSTM cells to regulate what information to add, retain, or output at each time step, effectively managing long-range dependencies in sequential data.

- Input Node $g^{(t)}$: This unit receives input from the current data point $x^{(t)}$ and the previous hidden state $h^{(t-1)}$, and applies a Tanh activation function to produce a candidate update to the cell state.
- Input Gate $i^{(t)}$: A sigmoid gate that determines how much of the candidate information should be incorporated into the cell state.
- Forget Gate $f^{(t)}$: Introduced by [112], this gate controls how much of the previous cell state should be retained or discarded.
- Output Gate $o^{(t)}$: This gate controls the cell's output by regulating how much of the cell state is exposed through the hidden state.

The following equations describe the operations of a standard LSTM cell [105]:

$$g^{(t)} = \phi(W^{(gx)}x^{(t)} + W^{(gh)}h^{(t-1)} + b_g), \tag{2.11}$$

$$i^{(t)} = \sigma(W^{(ix)}x^{(t)} + W^{(ih)}h^{(t-1)} + b_i), \tag{2.12}$$

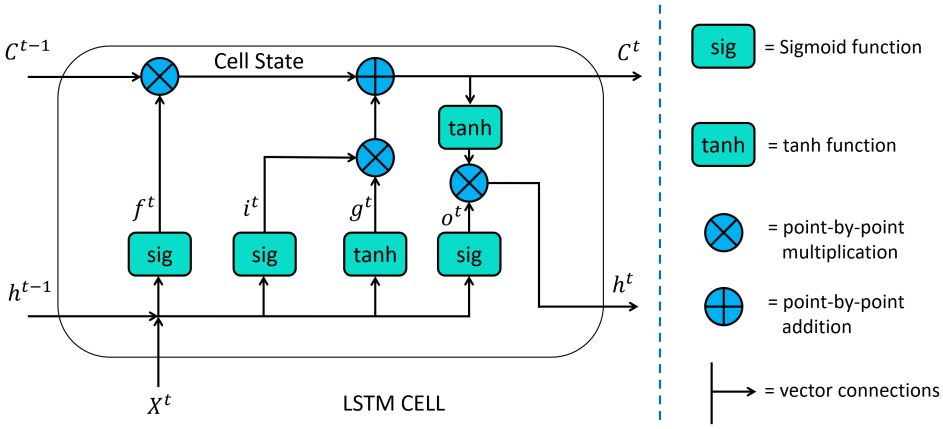


Figure 2.6: LSTM cell architecture showing the input, forget, and output gates that regulate information flow. Here, $x^{(t)}$ denotes the input at the current time step, $h^{(t-1)}$ and $h^{(t)}$ are the previous and current hidden states, respectively, and $c^{(t-1)}$ and $c^{(t)}$ are the previous and current cell states. Adapted from [103].

$$f^{(t)} = \sigma(W^{(fx)}x^{(t)} + W^{(fh)}h^{(t-1)} + b_f), \tag{2.13}$$

$$o^{(t)} = \sigma(W^{(ox)}x^{(t)} + W^{(oh)}h^{(t-1)} + b_o), \tag{2.14}$$

$$c^{(t)} = g^{(t)} \odot i^{(t)} + c^{(t-1)} \odot f^{(t)}, \tag{2.15}$$

and

$$h^{(t)} = \phi(c^{(t)}) \odot o^{(t)}. \tag{2.16}$$

In these equations, $i^{(t)}$, $f^{(t)}$, and $o^{(t)}$ denote the input, forget, and output gates, respectively. The sigmoid function is represented by σ, while ϕ denotes the Tanh activation function. The term $h^{(t-1)}$ represents the hidden state from the previous time step. The weights W and biases b define the transformations applied to the input and hidden state. The cell state $c^{(t)}$ represents the long-term memory, while $g^{(t)}$ defines the candidate update, enabling the LSTM to capture long-term dependencies effectively [86].

2.3.2 LSTM Architectures

LSTM architectures are specialised forms of RNNs designed to capture long-range dependencies in sequential data through memory cells

and gating mechanisms, including input, forget, and output gates. Several architectural variants have been proposed to enhance performance in different contexts. The standard or Vanilla LSTM, introduced by [103], consists of a single layer of memory cells and serves as the foundational architecture for modelling temporal dependencies. While effective, its representational capacity can be extended through deeper structures.

Stacked LSTM architectures increase model depth by layering multiple LSTM units, allowing the network to learn hierarchical temporal features. This enables more complex representations of sequential data [66]. Bidirectional LSTM (BiLSTM) further improves performance by processing sequences in both forward and backward directions. This provides a richer contextual representation, which is particularly beneficial in tasks where both past and future information are important, such as speech recognition and natural language processing [113]. Other extensions include Convolutional LSTM (ConvLSTM), which integrates convolutional operations to capture spatiotemporal dependencies, and attention-based LSTM models, which allow the network to focus on relevant parts of the input sequence. However, in this work, the focus remains on standard and stacked LSTM architectures due to their suitability for time series signal analysis.

2.4 Summary

This chapter provides an overview of ML, a branch of artificial intelligence focused on developing systems that learn from data to make predictions or informed decisions. It introduces key learning paradigms, including supervised, unsupervised, and reinforcement learning, and discusses factors that influence learning quality, such as data availability, feature representation, and model selection. Common challenges, including overfitting, noise, and limited generalisation, are also addressed, alongside techniques for improving model performance, such as feature engineering, regularisation, and hyperparameter tuning.

The chapter further explores neural networks as a powerful class of ML models inspired by the structure of the human brain. It presents

the perceptron as a fundamental building block and extends this concept to multi-layer architectures such as MLPs, which enable the modelling of complex non-linear relationships. Training these models is achieved through optimisation techniques such as gradient descent and its variants, supported by the backpropagation algorithm for efficient gradient computation. To address sequential and time-dependent data, the chapter introduces RNNs, which incorporate memory of previous inputs. It highlights the limitations of standard RNNs, particularly the VGP, and presents LSTM networks as an effective solution. By employing gating mechanisms to regulate information flow, LSTMs enable the modelling of long-range dependencies and form a key component of modern deep learning approaches for sequence modelling.

CHAPTER 3

Gravitational-Wave Data Analysis: Classical and ML Methods

The development of GW data-analysis methods began well before the first direct detection of GW150914 and has continued to evolve rapidly since then, leading to a sophisticated ecosystem of techniques for signal detection, parameter estimation, detector characterisation, and low-latency analysis [26, 65].

In practice, GW data analysis encompasses several tightly coupled tasks. Signal detection involves deciding whether a segment of noisy strain data contains a GW signal and, if so, identifying the astrophysical source that produced it. For transient signals, these sources may include CBC systems such as BBH, BNS, or NSBH mergers, as well as unmodelled burst events. In addition to these transient sources, GW detectors also search for persistent signals, including CW from rotating neutron stars and the Stochastic Gravitational-Wave Background (SGWB). Once a confident detection is made, PE is used to infer the physical properties of the source, such as component masses, spins, sky location, inclination angle, and luminosity distance. For CBC sources involving neutron stars, a further emerging goal is early-warning detection, in which the inspiral is recognised before merger so that EM and neutrino observatories can be alerted in advance. Such alerts enable searches for coincident neutrino emission associated with the same astrophysical event, providing an additional messenger that can help constrain the source physics and strengthen multi-messenger interpretations.

Historically, these tasks have been addressed by classical methods grounded in statistical signal processing and Bayesian inference. Template-based matched-filter searches, combined with coincidence tests across multiple detectors and hierarchical veto strategies, have delivered all confident CBC detections reported by the LVK collaboration to date

[14, 35, 114]. Similarly, Bayesian stochastic sampling algorithms remain the standard for PE, providing well-calibrated posterior distributions at the cost of substantial computational resources [115, 116].

Over the past decade, however, there has been an accelerating shift toward data-driven, ML approaches. The core idea is to replace or augment hand-crafted, physics-driven pipelines with models that learn discriminative or generative structures directly from data. CNNs have been proposed as an efficient approach for rapid detection of GW signals [62, 117]. RNNs and LSTM networks have also been explored for time series analysis and signal detection [1]. More recently, deep learning approaches have been investigated for denoising and rapid inference [118], while normalising flows and other deep generative models have been proposed for fast PE [118]. These models often achieve dramatic gains in throughput and latency, albeit at the cost of reduced interpretability and the need for large, carefully curated training sets.

For the purposes of this thesis, two aspects of this landscape are particularly important. First, classical pipelines remain the reference standard against which any new ML-based method must be compared. Their strengths and limitations therefore set the performance targets and define the relevant regimes of interest (e.g., low SNR, long-duration BNS inspirals, non-stationary noise). Second, ML approaches open a qualitatively new possibility: instead of passively accumulating SNR until a fixed threshold is crossed, one can train time series models, such as RNNs and LSTM networks, to recognise the evolving chirp pattern of an inspiral and trigger on informative partial waveforms, potentially well before merger.

This chapter reviews the landscape of GW data analysis techniques from two complementary perspectives. Sec. 3.1 discusses signal detection, contrasting matched-filter and burst pipelines with modern supervised ML classifiers and highlighting trade-offs between sensitivity, computational cost, robustness to noise artefacts, and interpretability. Sec. 3.2 turns to PE, comparing classical Bayesian samplers with emerging neural posterior-inference methods. The final part of the chapter, Sec. 3.3,

focuses extensively on early-warning detection, the primary topic of this thesis. We review classical early-warning modes implemented in GstLAL, SPIIR, and Multi-Band Template Analysis (MBTA), followed by ML-based early-warning studies that explore the use of CNN and RNNs, including LSTM, for pre-merger detection. Throughout, particular attention is paid to comparative assessment: the differences between classical and ML approaches in their statistical guarantees, computational requirements, and suitability for real-time early-warning. This comparative perspective motivates the LSTM-based early-warning framework developed in Chapters 4 and 5

3.1 Signal Detection in GW Astronomy

3.1.1 Classical Approaches

Matched filtering has been the cornerstone of CBC searches since the initial LIGO observing era [26]. As discussed in Sec. 1.5, the method correlates detector data with a bank of waveform templates predicted by general relativity, maximising the SNR over unknown parameters such as coalescence time and phase [26, 28]. For stationary, Gaussian noise and perfectly known signal models, matched filtering is provably optimal according to Neyman and Pearson, maximising detection probability at a fixed false-alarm probability. This optimality result underpins the design philosophy of current CBC pipelines. Modern pipelines such as PyCBC [34], GstLAL [119], MBTA [120], and SPIIR [121] implement matched filtering in different but conceptually related architectures. PyCBC uses frequency-domain filtering with discrete Fourier transforms and constructs a ranking statistic that combines SNR with χ^2-based signal-consistency tests to suppress glitches [122]. GstLAL performs time-domain and frequency-domain matched filtering in a streaming architecture optimised for low-latency, leveraging singular-value decomposition to compress template banks and reduce computational cost [119]. The MBTA divides the signal into multiple frequency bands, enabling a multi-band matched-filter implementation that accelerates computation while preserving sensitivity to CBC signals [120]. SPIIR approximates

the matched filter by a bank of Infinite Impulse Response (IIR) filters, enabling fully streaming, low-latency operation [121].

A key practical limitation of matched-filter searches is their scaling with the template bank size. To cover the parameter space of BNS and BBH systems with minimal loss in SNR, one constructs a discrete bank of templates such that the mismatch between any true signal and the nearest template is below a prescribed tolerance, typically corresponding to a maximum mismatch of approximately 3% [28]. The number of required templates grows rapidly as one extends to longer signals (lower starting frequencies), higher-dimensional parameter spaces (including spins and eccentricity), or more extreme mass ratios. For BNS systems, which can remain in the sensitive band of ground-based detectors for tens to hundreds of seconds at design sensitivity, the template banks become large and the associated computational cost substantial.

Additional complications arise from the non-Gaussian, non-stationary character of real detector noise. Transient instrumental artefacts, commonly referred to as glitches, can mimic or obscure CBC signals, leading to spurious triggers or reduced sensitivity [10]. Classical pipelines therefore rely on a combination of data-quality vetoes, signal-consistency tests (such as the reduced χ^2 statistic), and coincidence and coherence checks across detectors to maintain acceptable FARs. These techniques are effective but introduce further complexity and latency, especially when operating in real time. Despite these challenges, classical matched-filter pipelines offer several decisive advantages. Their statistical properties are well understood; they produce interpretable ranking statistics; and they can be systematically validated through injection campaigns and Mock Data Challenges (MDCs). For this reason, they remain the authoritative source of detections in LVK observing runs and provide the baseline against which alternative approaches, including ML methods, must be assessed.

3.1.2 Machine Learning Approaches

ML approaches to GW signal detection aim to either (i) replace matched filtering entirely with learned discriminators, or (ii) complement classical searches by providing auxiliary triggers, denoised data, or fast pre-classification of candidate events. The majority of existing work focuses on supervised deep learning, in which neural networks are trained to distinguish between segments of time series data that contain CBC signals and those that contain only noise (or noise plus glitches).

CNNs were among the earliest architectures explored for CBC detection. Initial work showed that deep CNNs trained on whitened time series inputs can recover BBH signals in simulated Gaussian noise with sensitivity comparable to matched-filter searches. Gabbard et al. [62] subsequently demonstrated that deep neural networks can achieve similar performance for CBC detection. Subsequent studies extended this approach to non-Gaussian, real detector noise and to multi-detector inputs [123, 124]. These works showed that, given sufficient training data spanning the relevant parameter space and noise conditions, CNNs can approximate the matched-filter decision boundary while evaluating candidate segments orders of magnitude faster than template-bank correlation. As discussed in Chapter 2, RNNs, including LSTM and GRU architectures, are well suited to sequential data such as GW time series. These models can learn temporal dependencies in the data and track the evolving chirp pattern of an inspiralling compact binary system.

Several studies have investigated CNN-based detectors that process either raw or pre-processed strain data and output a time-dependent detection probability [123, 125]. Hybrid models combining CNN feature extractors with LSTM architectures, sometimes referred to as Convolutional, Long Short-Term Memory, and Fully Connected Deep Neural Networks (CLDNNs), have also been proposed, aiming to exploit both local time-frequency features and long-term temporal structure. Beyond deterministic deep networks, Bayesian neural networks have been explored as approaches for quantifying epistemic uncertainty, that is, uncertainty arising from incomplete knowledge of the model parameters [126]. These

models attach probabilistic interpretations to network outputs, which is important for connecting ML-based detection statistics to astrophysically meaningful FARs and for integrating ML triggers into existing ranking and vetting frameworks.

In addition to direct detection, ML techniques have been applied to tasks that indirectly support signal searches. For example, deep CNNs have been used to classify and cluster glitches in LIGO data [9], improving data-quality vetoes and mitigating transient noise artefacts. Autoencoders and generative models have been used to denoise strain data, potentially enhancing the performance of both classical and ML detectors at low SNR [124].

Although these lines of work are not the main focus of this thesis, they illustrate the breadth of ML applications in GW data analysis. In recent years, several studies have moved beyond proof-of-concept models towards the development of end-to-end ML pipelines for CBC detection. A major benchmark in this area is the first ML GW Search Mock Data Challenge (MLGWSC-1), which was designed to provide a common framework for comparing different ML-based search algorithms under increasingly realistic conditions. Participants were asked to identify simulated BBH signals embedded in four datasets of increasing complexity. The final dataset contained real detector noise from the first half of O3, with signals lasting up to approximately 20 s and including physical effects such as spin precession and higher-order modes. Six search algorithms were evaluated using one month of previously unseen test data, including four ML-based methods and two traditional matched-filter-based production analyses. Performance was compared primarily in terms of sensitive distance as a function of FAR, together with computational runtime. In simulated Gaussian noise, the best ML searches achieved up to approximately 95% of the sensitive distance of the matched-filter searches at a false-alarm rate of one per month, while in real O3a noise the leading ML method achieved approximately 70% of the matched-filter sensitivity. These results showed that ML-based searches can approach classical matched-filter performance in some regimes, while also

highlighting the challenge of maintaining low false-alarm rates in realistic detector noise [127]. Building on these developments, several dedicated ML search pipelines have been proposed, including AresGW [128], which employs deep residual networks for BBH detection, Aframe [25], an end-to-end neural-network pipeline designed for low-latency CBC searches, and SAGE [129], a ML-based BBH search pipeline developed to address learning biases and improve detection performance in realistic detector noise.

Classical vs ML Approaches to Detection

Several key differences exist between classical matched-filter pipelines and ML-based detectors, which are directly relevant to this work.

- **Statistical optimality and calibration**: Matched filtering provides detection statistics whose false-alarm and false-dismissal rates can be estimated analytically and through injection campaigns. ML models, by contrast, typically do not enjoy such closed-form optimality results. Their outputs must be empirically calibrated to FARs using large validation sets, and their behaviour outside the training distribution can be difficult to quantify.

- **Computational cost and latency**: Classical pipelines require evaluating correlations against large template banks, with cost scaling linearly with the number of templates (modulo algorithmic optimisations). This becomes particularly burdensome for long-duration BNS signals and in low-latency operation. Once trained, ML models such as CNN and LSTM can evaluate candidate segments extremely quickly on modern hardware (especially GPUs), enabling near-real-time scanning of continuous data streams.

- **Dependence on waveform models**: Matched-filter searches rely explicitly on accurate waveform models and can suffer from reduced sensitivity if physical effects (e.g., precession, eccentricity, tides) are not included in template banks. ML detectors are also dependent on the waveform models used to generate their training data. Training on a diverse population of simulated signals can expose the model

to a broader range of source parameters and waveform morphologies, but this does not remove modelling uncertainties. If relevant physical effects are absent or inadequately represented in the training set, ML performance may also degrade when applied to signals outside the training distribution.

- **Robustness to non-stationary noise**: Classical pipelines incorporate specific, hand-designed signal-consistency tests and veto strategies to mitigate the impact of glitches and non-stationary noise. ML models can learn to distinguish glitches from CBC signals if such artefacts are adequately represented in the training data, but their generalisation to new or evolving noise conditions is not guaranteed. A change in the noise distribution can therefore lead to misclassification, reduced detection sensitivity, or an increased false-alarm rate if the model encounters noise features that were not adequately represented during training. This places a premium on continuous retraining or domain-adaptation strategies if ML detectors are to be deployed in production.

- **Interpretability**: The matched filter, χ^2 tests, and ranking statistics in classical pipelines have transparent physical and statistical meanings. By contrast, deep neural networks are often treated as black boxes, with internal feature representations that are difficult to interpret in terms of waveform morphology or detector behaviour. This can complicate validation and acceptance of ML triggers in a conservative, safety-critical context such as LVK detection pipelines.

In summary, classical matched-filter pipelines currently provide the statistically rigorous backbone of GW detection, while ML methods offer compelling advantages in speed and flexibility. For early-warning detection, where latency and the ability to exploit partial inspiral information are crucial, ML sequence models such as LSTMs may provide unique benefits that complement, rather than replace, classical pipelines. This motivates a careful comparative exploration of their capabilities and limitations in the context of CBC inspirals.

3.2 Parameter Estimation

3.2.1 Classical Bayesian Inference

As discussed in Chapter 1, classical PE for GW signals is formulated within a Bayesian framework. Rather than repeating the Bayesian formalism here, this section focuses on the computational methods used to evaluate the posterior distribution in practical GW analyses and on their limitations for low-latency applications. Classical PE frameworks such as LALInference [115] and Bilby [116] approximate the posterior using stochastic sampling algorithms. MCMC and nested sampling are the most widely used approaches. These methods explore the high-dimensional parameter space by evaluating the likelihood at many points, guided by the posterior structure. While highly flexible and asymptotically exact under broad conditions, they are computationally expensive: a single event can require millions of likelihood evaluations, each involving generating a model waveform and computing its overlap with the data. For BBH events with relatively short signals, PE can be performed in hours on modest computing resources. For BNS events, however, the situation is more challenging. The longer inspiral phase requires longer waveform durations and more costly likelihood evaluations, and the higher dimensionality of the parameter space (e.g., due to tidal parameters) further increases sampling complexity. This leads to PE runs that can take many hours to days, even with reduced-order quadratures or surrogate waveform models.

To mitigate these costs, several acceleration strategies have been developed. Reduced-order modelling and surrogate waveform techniques replace full waveform-generation codes with interpolated models trained on a sparse grid of parameter points, greatly reducing evaluation time [130, 131]. Multi-band likelihood methods split the frequency domain into coarse and fine bands, reducing the computational cost of likelihood evaluations [132]. Nonetheless, there remains a tension between accuracy, robustness, and computational cost, particularly for low-latency PE suitable for EM follow-up [40].

3.2.2 Machine Learning Approaches to Parameter Estimation

ML methods for PE seek to reduce the computational cost of inference by shifting much of the computation from per-event sampling to an upfront training stage. The central idea is to train a flexible neural network to approximate either the posterior $p(\theta \mid d)$ directly, or related objects such as the likelihood or conditional summary statistics, using simulated pairs (θ, d). Neural Posterior Estimation (NPE) frameworks, often built on normalising flows or invertible neural networks, have shown that accurate approximations to GW posteriors can be obtained with orders-of-magnitude speedups at inference time [118]. In these approaches, a flow-based model learns to transform a simple base distribution (e.g., a multivariate Gaussian) into a complex posterior-like distribution conditioned on observed data. Once trained, the model can produce approximate posterior samples for new events in fractions of a second.

Complementary work has explored conditional variational autoencoders, conditional neural processes, and diffusion-based generative models as tools for fast PE and waveform emulation [118, 133, 134]. These models can interpolate across waveform families and capture nontrivial parameter correlations, potentially surpassing the efficiency of traditional surrogate-modelling approaches. In some cases, they can be integrated directly into Bayesian pipelines as learned likelihood or proposal distributions. A key attraction of ML-based PE is its suitability for low-latency applications. If parameter estimates (or even approximate sky maps and distance posteriors) can be produced within seconds of a detection, they can inform rapid EM follow-up strategies and help prioritise telescope resources. This has motivated the development of end-to-end neural PE frameworks capable of operating on streaming data in parallel with classical sampling codes [118]. At the same time, ML methods for PE face several challenges. Training requires large, representative sets of simulated signals spanning the relevant astrophysical and detector configurations. Mismatches between training and real data, such as calibration uncertainties, non-stationary noise, or unmodeled physics, can introduce biases

in inferred parameters. Furthermore, rigorous uncertainty quantification and validation remain active areas of research: while ML models can produce samples that resemble posterior draws, demonstrating that these samples faithfully capture the true posterior (e.g., via coverage tests) is non-trivial.

Classical vs ML Approaches to PE

The contrast between classical Bayesian PE and ML-based PE can be summarised along lines similar to those discussed for detection:

- **Accuracy and robustness**: Classical sampling-based PE methods, when supplied with accurate waveform models and appropriate noise assumptions, produce posteriors that are, in principle, asymptotically exact. Their main limitations arise from waveform systematics, noise-model uncertainties, calibration uncertainties, and computational constraints. ML-based PE methods approximate the posterior directly and, in practice, can achieve good agreement with classical posteriors for many events, but they offer weaker formal guarantees and more complex validation.

- **Computational profile**: Classical PE methods incur most of their computational cost per detected event, with little opportunity to reuse previous calculations. In contrast, ML-based PE approaches typically require a computationally intensive training stage performed once in advance, after which inference for individual events can be carried out extremely rapidly. This strategy, often referred to as amortised inference, distributes the cost of learning the inference procedure across many subsequent analyses. It is particularly advantageous when large numbers of similar signals are expected, as in CBC populations observed during advanced-detector observing runs and in future third-generation detectors.

- **Suitability for low-latency and early-warning**: For early-warning applications, where preliminary parameter information is required

before or shortly after merger, classical PE methods are challenged by their runtime. Reduced-order and rapid PE methods exist, but still typically operate on timescales of minutes. ML-based PE can, in principle, deliver informative approximate posteriors in seconds, making it a natural partner to low-latency detection and early-warning triggers.

- **Dependence on simulations and priors**: Both classical and ML-based PE depend on waveform models and prior assumptions, but ML methods also depend critically on the distribution of training simulations. If the training set does not adequately cover rare but scientifically important regions of parameter space (e.g., highly asymmetric masses or extreme spins), ML posteriors may be biased or overly confident in those regimes. Classical samplers, while still model-dependent, can more flexibly explore surprising regions of parameter space if the likelihood supports them.

From the perspective of this thesis, ML-based PE serves primarily as contextual background: it illustrates how deep learning is reshaping multiple facets of GW data analysis beyond detection alone. The main focus hereafter will be on detection, in particular early-warning detection, where sequence models such as LSTMs can exploit the temporal structure of inspiral signals to trigger on pre-merger data. Classical and ML approaches to early-warning detection are therefore examined in detail in Sec. 3.3.

3.3 Early-Warning Detection of Compact Binary Coalescences

The prospect of issuing GW alerts prior to the merger of a compact binary has emerged as a central objective in multi-messenger astrophysics. For BNS and NSBH systems, the long inspiral phase provides a temporal window during which the evolving GW signal may be identified early enough to enable negative-latency EM alerts. Such capability would allow observatories to reposition their instruments before or around the time

of merger, when the earliest electromagnetic counterparts may appear. Prompt gamma-ray emission from a short GRB can occur within seconds of the merger, while optical and near-infrared kilonova emission develops over timescales of hours to days as radioactive ejecta expand and cool. X-ray and radio afterglows associated with relativistic outflows can emerge later and evolve over days to months. Early-warning alerts are therefore especially valuable because they increase the chance of capturing the earliest phases of the electromagnetic emission and of establishing a well-constrained multi-messenger timeline for the event [44, 135]. Achieving early-warning, however, requires algorithms capable of detecting weak, gradually accumulating signatures embedded in detector data that are largely characterised by approximately stationary Gaussian noise, but which also contain transient noise artefacts and slow non-stationarities.

3.3.1 Classical Early-Warning Pipelines

Classical approaches to early-warning are grounded in the matched-filtering paradigm. They attempt to trigger on the partial inspiral signal before the waveform reaches its maximum amplitude. While matched filtering is statistically optimal under the assumptions of Gaussian, stationary noise and perfectly known signal models [21, 136], the low-frequency regime relevant for early-warning challenges these assumptions. In practice, detector noise is only approximately stationary on relatively short timescales (typically tens of seconds to minutes) and exhibits slow drifts as well as transient noise artefacts [137]. In addition, the accumulated SNR in the early inspiral is small. As a result, classical pipelines require substantial modifications to operate reliably in an early-warning mode. The primary low-latency pipelines used by the LVK collaboration—GstLAL, MBTA, PyCBC Live, and SPIIR—include several pipelines that have been adapted or investigated for early-warning operation. Their operational principles, strengths, and limitations are reviewed in the following subsections.

The GstLAL pipeline combines time-domain whitening with a likelihood-ratio ranking statistic [119, 138]. Its architecture supports streaming

matched filtering by constructing whitened template waveforms and evaluating likelihood ratios continuously as data arrive. In the early-warning context, GstLAL accumulates SNR over the inspiral-only portion of the waveform and issues a trigger once the ranking statistic exceeds a calibrated threshold [49]. Early simulations for design-sensitivity detectors suggested that GstLAL can identify BNS systems several seconds before merger for sufficiently nearby systems in which enough SNR has accumulated before merger [53, 54]. However, significant challenges arise due to the low-frequency (20–60 Hz) noise structure of the detectors. Transient glitches, narrow instrumental spectral features such as calibration lines and power-mains-related lines, and other non-Gaussian noise features can lead to elevated FARs at early times. To control these rates, GstLAL employs conservative likelihood-ratio thresholds, which reduce sensitivity in the pre-merger regime. During O3 and O4, although GstLAL provided low-latency post-merger triggers for BNS and NSBH systems, its achievable early-warning times depend strongly on the source mass, the detector's low-frequency sensitivity in a given observing run, and the details of the search algorithm. In practice, warning times are therefore highly variable and depend strongly on source properties, detector sensitivity, and the details of the low-latency search. Studies and low-latency alert analyses for the O3–O4 era indicate that only a subset of favourable BNS systems are expected to be identified before merger, with warning times typically ranging from a few seconds to at most several tens of seconds under favourable conditions [53, 139].

The MBTA pipeline [24, 140] uses a frequency-domain decomposition of the matched-filter integral. By dividing templates into low-frequency and high-frequency bands, MBTA reduces computational cost while preserving sensitivity. In the early-warning regime, the low-frequency band becomes especially important, as it contains much of the inspiral evolution. Early-warning modes of MBTA have been explored in simulation studies for earlier LIGO observing runs prior to the Advanced LIGO era [140]. These studies indicated that the multi-band approach naturally lends itself to pre-merger triggering; however, its susceptibility to

low-frequency transients remains similar to that of GstLAL, and during O3 the achievable low-latency performance of MBTA remained strongly dependent on detector noise conditions and source properties [24].

PyCBC Live extends the classical PyCBC matched-filter search into a low-latency streaming environment [23]. Its early-warning capabilities depend on the accumulation of matched-filter SNR from the early inspiral. While PyCBC Live demonstrated robust post-merger performance during the O3 run, early-warning operation faced similar constraints to GstLAL and MBTA. In the pre-merger regime, the pipeline must contend with reduced SNR and weaker diagnostic power from the χ^2 signal-consistency tests [122], which are essential for glitch rejection. Consequently, early-warning triggering has been limited to systems with high network SNR and favourable source orientations relative to the detector network, which produce stronger detector responses.

SPIIR pipeline [51, 121] approximates matched filtering through a bank of IIR filters designed to mimic the chirp waveform. This time-domain architecture enables low-latency responses suitable for real-time detection. Because SPIIR updates its detection statistic continuously without requiring repeated Fourier transforms, it can track inspiral evolution in real time with reduced computational overhead.

Simulation studies based on anticipated O4 detector sensitivities demonstrated that SPIIR could provide early-warning triggers for some BNS systems at least 10 s before coalescence [141]. These results demonstrate the potential of classical matched-filter-based methods for pre-merger detection. Nevertheless, a fundamental challenge is that detection decisions must be made using only a partial inspiral signal, before the waveform has accumulated its full SNR. This limited signal information reduces the available discriminative power and increases vulnerability to low-frequency glitches and non-stationary noise. For matched-filter-based methods such as SPIIR, this challenge becomes particularly important when operating deep in the low-frequency regime.

3.3.2 Limitations of Classical Early-Warning

Although classical pipelines have achieved substantial progress, several structural limitations restrict their early-warning performance:

- **Low accumulated SNR**: At early times in the inspiral, only a small fraction of the total signal power has accumulated in the matched filter, and the accumulated SNR therefore remains low. This reduces the effectiveness of signal-consistency tests such as the reduced χ^2 statistic [122], which typically become more informative closer to merger, where the signal power and accumulated SNR are larger.

  Non-stationary noise: Below ~ 60 Hz, detector sensitivity can be affected by time-varying environmental and technical noise sources, including seismic disturbances and instrumental noise. [142]. Although the noise can often be treated as approximately stationary over short analysis segments (typically tens of seconds to a few minutes), its statistical properties drift on longer timescales due to environmental and instrumental variations, and transient noise artefacts may occur on millisecond-to-second timescales. These non-stationary features can significantly elevate FARs for classical matched-filter searches operating in the low-frequency regime.

- **Template-bank dependence**: Early-warning sensitivity depends critically on the density and coverage of the template bank. In practice, template banks must span a range of source parameters, including component masses and spin configurations. Incorporating additional physical effects such as spin-induced precession can increase the dimensionality of the parameter space and the computational cost of the search [34, 143]. Consequently, early-warning searches must balance waveform fidelity against the computational constraints associated with low-latency matched filtering.

- **Conservative thresholding**: Classical early-warning pipelines estimate the background distribution of their ranking statistics using

pipeline-specific background-estimation procedures and set detection thresholds to achieve a desired FAR [119]. Because only a small fraction of the inspiral signal has accumulated prior to merger, the available SNR is limited at early times. Consequently, detection statistics must rely on weaker evidence, making it more difficult to distinguish signals from noise fluctuations and limiting the achievable warning time.

Collectively, these limitations motivate the exploration of alternative approaches that may extract additional structure from the early inspiral signal, potentially complementing traditional ranking statistics based primarily on accumulated SNR and signal-consistency tests.

3.3.3 Machine Learning Approaches to Early-Warning

ML methods offer a different paradigm for early-warning detection. Rather than relying on template-based correlations and analytically derived ranking statistics, ML approaches learn patterns directly from training data containing examples of signals and noise, as discussed in Chapter 2. ML approaches have been explored as potential alternatives or complements to matched-filter pipelines, particularly in contexts where some of the practical limitations discussed in Subsec. 3.3.2 may affect performance. Existing ML-based early-warning studies can be broadly grouped into several methodological families, including convolutional architectures, recurrent and sequence models, forecasting or predictive frameworks, and multi-channel or noise-aware approaches. These approaches differ in how they model the inspiral signal and handle detector noise. Each family is reviewed below. The earliest ML-based GW detection studies employed CNNs, leveraging their ability to extract localised features from time-domain or time-frequency representations. Early studies demonstrated that deep CNNs trained on whitened time series data can achieve detection sensitivity comparable to matched filtering for BBH signals in simulated Gaussian noise. Subsequent work extended CNN architectures to non-Gaussian real LIGO noise [144], multi-detector inputs [62], and spectrogram-based representations [117].

While these early CNNs studies were not explicitly designed for early-warning, they revealed two properties essential for pre-merger detection: (i) CNNs can approximate the matched-filter decision boundary without requiring template banks, and (ii) once trained, inference is extremely fast, enabling real-time operation. Baltus et al. [145] explored CNN architectures for long-duration BNS inspirals, showing that compact CNNs trained on early-time signal segments can trigger tens of seconds before merger in simulated design-sensitivity noise. This work provided the first explicit demonstration that ML-based methods could potentially surpass classical pipelines in achievable warning time under idealised conditions. However, CNNs operate using local convolutional filters that analyse limited time windows of the input data. As a result, they do not explicitly model long-range temporal dependencies, which can limit their performance for very long-duration signals.

RNNs, particularly LSTM networks, provide a natural architecture for time series analysis. As discussed in Chapter 2, their gating mechanisms allow information to be retained over extended temporal sequences, making them well suited to modelling the slowly evolving chirp of a BNS inspiral. Krastev [146] explored CNN networks trained on raw time series strain data and demonstrated that such models can identify CBC signals prior to merger in simulated Gaussian noise.

Yu et al. [147] investigated neural-network-based early-warning detection of compact binary coalescences in nonstationary noise backgrounds. Their approach combines noise mitigation with signal detection to improve low-frequency sensitivity, enabling earlier identification of inspiral signals. These results demonstrate the potential of neural networks for early-warning detection under challenging noise conditions. Beyond LSTMs, GRUs and transformer-based architectures have been explored. GRUs simplify the gating structure while retaining the ability to model long-term dependencies, offering computational advantages. Transformer models, while powerful, generally require large training datasets and have yet to demonstrate stable early-warning performance in non-stationary detector noise.

A distinct approach to ML-based early-warning involves identifying the evolving chirp patterns of inspiralling binaries before merger. Rather than relying solely on time-domain classification, these models analyse time-frequency representations of the detector strain to identify emerging signals. Wei and Huerta [57] proposed a deep learning forecasting approach based on a ResNet-50 architecture to identify chirp patterns in spectrograms of detector strain. In tests involving simulated BNS signals injected into real detector noise, their model identified signals up to 30 s before merger. When applied retrospectively to GW170817, the model identified the signal approximately 10 s prior to coalescence.

Martins et al. [148] investigated a WaveNet-based approach for early-warning detection of binary neutron star mergers, with particular emphasis on computational efficiency and low-latency implementation. Using simulated gravitational-wave signals embedded in Gaussian noise representative of the O3 observing run, their model improved detection accuracy compared with an existing machine-learning-based early-warning approach. They also explored deployment on field-programmable gate arrays (FPGAs), highlighting the potential of hardware acceleration for reducing computational latency in real-time detection pipelines.

Strengths and Limitations of ML-Based Early-Warning

ML-based early-warning approaches offer several compelling advantages:

- **Enhanced early-time sensitivity**: ML models can learn temporal correlations in the evolving inspiral waveform beyond the instantaneous SNR. Recurrent architectures such as LSTMs may therefore identify consistent sub-threshold structure before the coherent SNR becomes sufficiently large for classical detection.

- **Low-latency inference**: Once trained, ML models operate with millisecond-level inference speeds, far faster than template-bank matched filtering.

- **Flexibility in waveform modelling**: ML can incorporate a wide variety of simulated waveforms, including tidal effects, eccentricity,

and precession, without the need for explicit template-bank expansions.

However, several challenges remain:

- **Generalisation across noise conditions**: ML models are sensitive to the statistical properties of their training data. Unlike classical methods that often assume stationary noise, ML models can, in principle, learn non-stationary detector behaviour when trained on representative real data. However, their performance may still degrade if the noise conditions encountered during deployment differ significantly from those present in the training set.
- **Calibration of detection statistics**: Unlike matched filtering, ML outputs do not have a direct physical interpretation, such as SNR. However, in practice, both ML and matched-filter ranking statistics require empirical calibration through background studies to estimate FARs.
- **Limited deployment readiness**: ML-based early-warning frameworks have yet to achieve the level of robustness, interpretability, and community validation required for integration into LVK operational pipelines.

These limitations underscore the need for rigorous comparative studies between ML-based and classical approaches, including controlled validation with realistic noise and large-scale injection campaigns.

3.3.4 Comparative Assessment of Classical and Machine Learning Approaches

The development of early-warning detection frameworks has produced two broad methodological families: classical matched-filter-based pipelines and ML approaches. Although both aim to detect compact-binary inspirals as early as possible, their underlying assumptions, operational characteristics, and performance regimes differ fundamentally. A

meaningful assessment must therefore compare not only detection sensitivity but also statistical interpretability, robustness to noise, computational requirements, and readiness for deployment within large-scale GW infrastructures.

Classical early-warning pipelines derive their strength from well-established statistical foundations. For a known waveform in stationary Gaussian noise, the likelihood-ratio test reduces to matched filtering [21, 136]. In practical CBC searches, however, the signal parameters are unknown, and searches are performed over template banks with additional ranking statistics and empirical background calibration [119]. This theoretical grounding enables the construction of interpretable ranking statistics and reliable false-alarm calibration using pipeline-specific background-estimation procedures [119]. These properties explain why pipelines such as GstLAL, MBTA, PyCBC Live, and SPIIR continue to serve as established operational search pipelines for CBC detection in LVK observing runs. However, achieving reliable early-warning detection remains challenging because the available signal power is limited during the early inspiral, while nonstationary low-frequency detector noise can further reduce detection sensitivity [142].

ML approaches, in contrast, are not constrained by explicit matched-filter template banks. They learn discriminative structure directly from training data and can identify patterns associated with evolving inspiral signals [146]. Sequence architectures such as LSTMs may therefore begin accumulating evidence earlier in the inspiral, although their performance remains fundamentally limited by the available SNR. CNNs have also demonstrated the potential for detecting pre-merger inspiral signals in simulated environments [145].

Nonetheless, the statistical behaviour of ML detectors depends strongly on the representativeness of their training datasets, and calibrating their outputs to meaningful false-alarm probabilities remains an important challenge. Performance can degrade when models encounter noise transients or changes in the statistical properties of the detector noise that were not present in the training data. This highlights the importance

of evaluating ML methods under realistic noise conditions and carefully calibrating their detection statistics [127]. A second point of comparison arises in computational efficiency. Classical matched-filter pipelines scale with the size of the template bank, which grows rapidly when additional physical effects, such as spin precession or higher-order waveform structure, are incorporated [143].

While algorithmic optimizations (e.g., multi-banding, reduced-order modelling, and streaming IIR filters) mitigate this burden, the computational cost can remain substantial. ML models, once trained, can operate with low inference latency and are highly compatible with streaming workflows or GPU acceleration. However, their training phase demands extensive datasets and significant computational resources. For long-term deployments, ML frameworks may also require domain adaptation or continual learning to maintain performance across changing detector conditions. Finally, deployment readiness constitutes a central differentiator. Classical early-warning pipelines have been used operationally for several observing runs and are integrated into LVK alert infrastructures [139]. ML-based early-warning methods, while promising, have not yet been validated at the scale, reliability, and transparency required for routine production use. Integration challenges include calibration of detection scores, interpretability of internal representations, and rigorous validation of model performance under realistic detector-noise conditions.

Overall, classical early-warning pipelines remain the most mature and operationally validated approach, offering interpretable ranking statistics and reliable false-alarm calibration. ML methods, by contrast, provide attractive advantages in speed, flexibility, and the potential to recognise weak pre-merger structure earlier in the inspiral. However, their practical use remains limited by challenges in calibration, robustness to changing noise conditions, and deployment readiness. At present, ML methods are therefore best viewed as promising complementary tools rather than replacements for classical pipelines.

3.3.5 Thesis Contribution and Position

The review presented in this chapter demonstrates that early-warning detection of CBCs remains a challenging and unsolved problem at the intersection of detector noise characterisation, statistical signal processing, and ML. Classical matched-filter pipelines offer mature, interpretable, and operationally validated frameworks, yet their early-time sensitivity is fundamentally constrained by low accumulated SNR, non-stationary low-frequency noise, and conservative threshold requirements.

ML approaches, particularly sequence models such as LSTM networks, can integrate information over long inspiral durations and may therefore facilitate earlier signal identification in some scenarios. However, their deployment is currently hindered by calibration issues, generalization under realistic noise, and the absence of established statistical guarantees. This thesis addresses these gaps by developing a dedicated LSTM-based early-warning detection framework for CBC systems, including both BBH and BNS systems. The architecture is designed to model the quasi-monotonic frequency evolution of inspirals and to output a time-dependent detection probability at each timestep. This approach allows systematic quantification of warning time under controlled false-alarm constraints and enables systematic comparison with classical matched-filter pipelines.

Chapters 4 and 5 evaluate the LSTM model through large-scale injection campaigns in simulated Gaussian noise. Chapter 4 establishes a proof-of-concept using BBH signals to assess baseline detection performance, while Chapter 5 focuses on BNS systems to investigate early-warning capabilities, including detection-time distributions at fixed false-alarm thresholds and performance stability across varying SNR conditions. Chapter 6 extends the analysis to realistic O3 Replay detector data, examining the model's generalisation behaviour under domain shifts and non-stationary noise conditions. By situating these results within the methodological landscape reviewed in Sections 3.1-3.3, the thesis clarifies the potential role of LSTM-based early-warning frameworks as comple-

mentary tools to classical pipelines in future LVK observing runs and forthcoming third-generation detector networks.

CHAPTER 4

LSTM for Early-Warning Detection of Gravitational Waves

4.1 Introduction

The previous chapters introduced the main concepts of GW detection, machine learning, and low-latency analysis that motivate the work presented in this thesis. Building on this background, this chapter investigates whether a recurrent neural network can identify compact-binary GW signals sufficiently early to provide advance warning before merger.

Early-warning detection is particularly relevant for compact binaries containing neutron stars, where a GW trigger prior to coalescence could provide additional time for electromagnetic observatories to prepare follow-up observations. Existing low-latency searches, including matched-filtering pipelines such as MBTA, have demonstrated the feasibility of identifying compact-binary signals with low latency [120, 149]. Deep-learning approaches have also been explored for low-latency and early-warning GW analysis [145, 148, 150]. These studies motivate the investigation of recurrent neural networks as an alternative approach for identifying evolving inspiral signals directly in the time domain.

In this chapter, we use a LSTM network [103] to investigate early-warning detection of transient GW signals from BBH systems. The analysis is performed using data from a single detector and is intended as a preliminary proof-of-concept study before extending the method to longer-duration, lower-mass systems. BBH signals provide a useful starting point because their shorter duration allows the behaviour of the network, the training procedure, and the temporal response of the model to be examined without the substantially larger data volumes associated with BNS inspirals.

The broader objective of the method is to develop an approach that can subsequently be applied to lower-mass systems, particularly BNS and

NSBH mergers, which are of greater interest for multi-messenger observations because of their potential electromagnetic counterparts. The LSTM architecture is well suited to this problem because it processes sequential time-domain data while retaining information through its recurrent hidden state. This allows the network to model temporal dependencies in an inspiral waveform and to respond to signal features that evolve before coalescence.

The present study focuses on signal detection and the timing of the network response rather than source localisation. Consequently, the method does not provide an early sky-localisation estimate, which would be required for a complete real-time multi-messenger alert system. The results presented here therefore establish the initial detection framework that is subsequently extended to longer-duration BNS signals in Chapter 5 and evaluated under realistic detector-noise conditions in Chapter 6.

The chapter is organised as follows. In Sec. 4.2, we introduce the LSTM architecture used in this study. Section 4.3 describes the data generation and training procedure, while Sec. 4.4 presents the design and development of the model. The results are presented in Sec. 4.5, followed by the conclusions in Sec. 4.6.

4.2 Recurrent Neural Networks

An RNN is a type of ANN designed to process sequential data within the scope of deep learning [105]. This distinctive architecture empowers RNNs to grasp and leverage information from preceding time steps, rendering them particularly effective in analysing time series data. This type of network has achieved excellent performance with sequential data over the past few years. For example, RNN applications are used in Apple's Siri and Google voice search [151]. In the context of GW detection, RNNs offer significant advantages. By training on a large dataset of labelled GW signals, RNNs can learn to recognise and classify different waveforms. This capability is crucial for identifying and distinguishing GW signals from background noise. They also benefit from a reduced relative size and complexity of the network compared to other deep learning

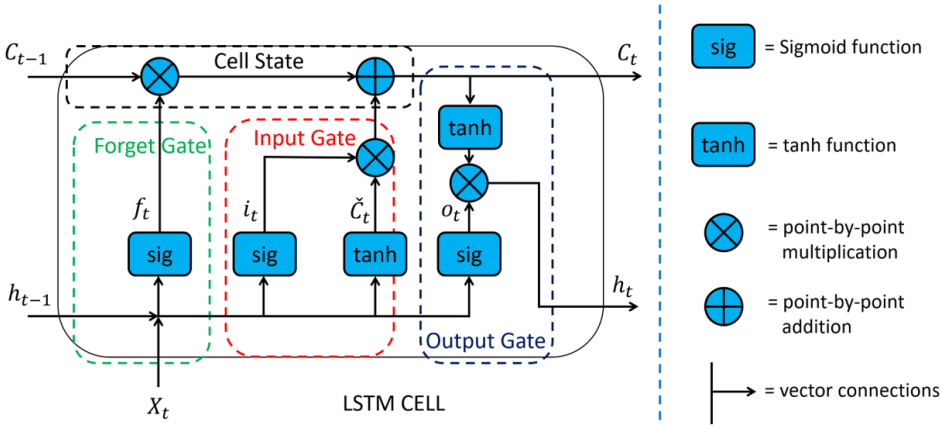


Figure 4.1: Simple structure of LSTM [154], where x_t is a current input, and h_{t-1} is the output of the last LSTM unit and c_{t-1} is the memory of the last LSTM unit, c_t is the updated memory, and h_t is the current output. The sigmoid and Tanh functions are the activation functions.

approaches, such as CNNs, where the input layer is receptive to the entire signal length. The purposeful retention of relevant information in an RNN allows it to focus on short sections of the input time series sequentially.

A LSTM network is a class of RNN, which was first introduced in 1997 [105]. It is designed to solve many tasks that were previously unsolvable using RNNs [152]. Moreover, the use of LSTMs can avoid gradient descent problems [103], which are often found in RNNs. It has achieved impressive results in various applications, such as language modelling, speech-to-text transcription, and machine translation [153]. The fundamental concept of an LSTM is built around a cell state that functions as long-term memory. The cell state acts as a transport highway that carries relevant information step by step through the sequence. Alongside it, the hidden state serves as the short-term memory, capturing the most recent information needed for the current time step's output. It consists of three gates: input gate, forget gate, and output gate, as shown in Fig. 4.1.

The gates control the information that goes through the LSTM. The forget gate decides which data should be discarded from the cell. The input gate determines which values from the current input to use to update the memory cell. Finally, the output gate decides which output

will be presented based on the input and the memory of the cell. The following set of equations can illustrate operations in an LSTM layer as follows:

$$i_t = \sigma\left(w_i \cdot [h_{t-1}, x_t] + b_i\right), \tag{4.1}$$

$$f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right), \tag{4.2}$$

and

$$o_t = \sigma\left(w_o \cdot [h_{t-1}, x_t] + b_o\right), \tag{4.3}$$

where i_t is the input gate, f_t is the forget gate, o_t is the output gate, σ is the sigmoid function, and w is a set of weights associated with each gate performing a linear transformation represented by a matrix. For the respective gate neurons (x), h_{t-1} is the output of the previous LSTM block at timestamp $t-1$, x_t is the input at the current timestamp, and b_x represents the biases for the respective gates (x). A bias is applied as an intercept in a linear equation; it is a parameter in the neural network, together with the network weights, that are used to change the output. The following set of equations then describes the cell state of the LSTM layer:

$$\tilde{c}_t = \tanh\left(w_c \cdot [h_{t-1}, x_t] + b_c\right), \tag{4.4}$$

$$c_t = f_t * c_{t-1} + i_t * \tilde{c}_t, \tag{4.5}$$

and

$$h_t = o_t * \tanh\left(c_t\right), \tag{4.6}$$

where $*$ represents the element wise multiplication of the vectors, c_t is cell state (memory) at timestamp t, and $\tilde{c}_t$ is the potential update to the cell state at timestamp t. The cell state gives the LSTM power to learn long strings of sequential data successfully [86]. However, various LSTM models have different architectures depending on the number of hidden layers.

In a stacked LSTM architecture (as used in the analysis presented in this chapter), the arrangement is such that several LSTM layers are stacked on top of each other, creating a deeper network structure. In

this configuration, each LSTM layer of the stack processes the input sequence and then forwards its output to the subsequent layer. This layered approach enables the network to discern and learn intricate features at varying degrees of abstraction. Essentially, the output of one LSTM layer becomes the input for the next, allowing for a more nuanced understanding and processing of the sequential data.

4.3 Data Generation Process

In this section, we describe the process used to simulate GW signals and detector noise. Since our LSTM model requires extensive training, validation, and testing, we generate a large dataset of simulated signals embedded in different realisations of detector noise to ensure robustness and generalisation. We generate simulated GWs using the PyCBC [155, 156] library. This study focuses on BBH signals rather than BNS or NSBH systems, as the higher masses of BBH systems produce shorter-duration signals that are well suited for our analysis. We randomize the parameters of our simulated non-spinning BBH signals according to the distributions listed in Tab. 4.1. Each parameter in our analysis is carefully constructed to fall within a specified range, ensuring comprehensive, randomised sampling across all relevant parameters. This approach is designed to create a robust and varied dataset that accurately reflects a realistic distribution of signal parameters.

Our analysis focuses on systems with masses ranging from $5M_{\odot}$ to $95M_{\odot}$, based on astrophysical observations suggesting that black hole masses typically fall within this range. The choice of this range allows us to capture the majority of likely black hole binary systems observed in the universe. We employ a power-law mass distribution for selecting the primary mass m_1, ensuring $(m_1 > m_2)$. The power law is of the form:

$$p(m_1) = \left(\frac{m_{\rm min}^{1-\alpha} - m_{\rm max}^{1-\alpha}}{\alpha - 1} m_1^{-\alpha} \right), \tag{4.7}$$

where $m_{\rm min} = 5M_{\odot}$, $m_{\rm max} = 95M_{\odot}$, and and $\alpha = 1.6$ is the power-law index. We selected this power-law prior because it is motivated by astrophysical evidence suggesting that the mass distribution of black holes

Table 4.1: The prior distribution used on the BBH signal parameters with zero spin, the distance is determined by the choice of SNR and the detector PSD.

Parameter	distribution	Min.	Max.
Power-law distribution P (m_1)	$\propto m_1^{-\alpha}$	$5M_\odot$	$95M_\odot$
Mass 2 (m_2)	Uniform	$5M_\odot$	m_1
Phase at Coalescence ($\phi_\circ$)	Uniform	0	2π rad.
Right Ascension (α)	Uniform	0	2π rad.
Sine Declination ($\sin(\delta)$)	Uniform	-1	1
Cosine Inclination ($\cos(\Theta_{jn})$)	Uniform	-1	1
Polarisation (ψ)	Uniform	0	π rad.
Coalescence time (t_c): Training+Validation	Uniform	4	9 seconds.
Coalescence time (t_c): Testing	Uniform	8	13 seconds.
Optimal SNR (ρ_{opt}): Training+Validation	Uniform	7	20
Optimal SNR (ρ_{opt}): Testing	Uniform	2	20

follows such a trend, particularly for the more massive components. The choice of α reflects the steepness of the distribution, with higher values implying that lower-mass black holes are more common. The chirp mass is defined as

$$\mathcal{M}_c = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}},$$

where m_1 and m_2 are the component masses of the binary. The chirp mass is a key parameter governing the leading-order inspiral evolution of the system.

For the sky location, inclination, polarisation, and phase, we adopt uniform priors to reflect our complete lack of prior knowledge about these parameters. Specifically, the sky position is uniformly distributed over the celestial sphere to ensure an isotropic source distribution. The inclination angle is drawn from a uniform distribution in $\cos(\Theta_{jn})$, which appropriately accounts for the isotropic orientation of binary systems relative to the observer. Similarly, the polarisation angle ψ and the phase at coalescence $\phi_\circ$ are sampled uniformly across their respective intervals, implying no preferred polarisation state or initial phase. These assumptions ensure that the simulated population covers all possible orientations and viewing angles without bias. The simulated signals are then projected onto a single detector—the H1 interferometer at LIGO Hanford Observa-

tory—whose antenna response depends on the randomised sky position and polarisation of each event. The optimal SNR of the simulated signals is therefore given by:

$$\rho^2_{\rm opt} = \frac{4}{d^2}\int_0^\infty \frac{|\tilde{h}_{d=1}(f)|^2}{Sn(f)}\,df, \tag{4.8}$$

where $\tilde{h}_{d=1}(f)$ is the Fourier transform of the signal originating from a distance of 1 Mpc, and $Sn(f)$ is the detector noise PSD. We control the SNR distribution by assigning the distance d (in units of Mpc) and correspondingly scaling the signal amplitude based on the desired SNR. We have chosen to use an SNR distribution that is uniform in the range between 7 and 20 for training the model, and for testing the model, we use a range between 2 and 20. SNR values between 7 and 20 are chosen for the training in order to allow the model to efficiently learn the salient characteristics of the signals. We generated a 12-second waveform with different merger times, with the merger occurring between 4 and 9 seconds. Then we took the first 8-second window, where the merger time can be inside or outside these 8 seconds, to train the model, while testing on longer data (12 seconds) with the merger located between 8 and 13 seconds.

We generate signals using the SEOBNRv4 opt approximant [157]. To simulate the signal from the SEOBNRv4 opt waveform, we initially employ different sampling frequencies depending on the system's mass. For systems with a mass less than 25 $M_\odot$, a sampling frequency of 4096 Hz is used. Conversely, for systems with a mass greater than 25 $M_\odot$, the sampling frequency is set to 2048 Hz. Following the initial simulation, we resampled the signal to the sampling frequency of 1024 Hz. This resampling process standardises the data for subsequent analysis and ensures consistency across our prior mass range, but it will suppress high-frequency information (close to the merger) for the lower-mass systems. We compute the signal's plus and cross-polarisation and, from this, construct $h(t)$ by applying the detector antenna pattern, a function of the randomised sky location and polarisation angles.

We generate Gaussian noise using the advanced LIGO aLIGOZeroDetHighPower [158] PSD. The `aLIGOZeroDetHighPower` PSD represents an idealised Advanced LIGO design-sensitivity noise curve and is used here to provide a controlled and reproducible baseline for the proof-of-concept analysis. Figure 4.2 shows the PSD over the frequency range used in this chapter, from the lower-frequency cutoff of 20 Hz to the Nyquist frequency of 512 Hz, corresponding to the 1024 Hz sampling rate. The use of this idealised PSD allows the model to be trained and evaluated under stationary Gaussian-noise conditions, without additional variability arising from real detector data. However, it does not capture the non-stationary and non-Gaussian features present in operating interferometers. In contrast, the measured sensitivity of an operating detector varies with time and contains frequency-dependent spectral features that are absent from the smooth design-sensitivity model. Consequently, training exclusively on this idealised noise model may lead to a domain mismatch when the network is applied to real detector data, potentially reducing its generalisation performance. The effect of this mismatch, and the model's behaviour under more realistic detector-noise conditions, are investigated further in Chapter 6. Then, we generate signals using random parameters drawn from our prior and add them to the noise; examples of training data time series are shown in Fig. 4.3 and 4.4. These diverse data samples are critical for effectively training the model to distinguish between the presence of noise and the presence of signal plus noise. We whitened the data before using it in our LSTM algorithms. Where the whitened waveform is given by:

$$h_{\mathrm{w}}(t) = \mathcal{F}^{-1}\left(\frac{\mathcal{F}\{h(t)\}}{\sqrt{S(f)}}\right), \tag{4.9}$$

where $\mathcal{F}\{h(t)\}$ is the Fourier transform of the original time series $h(t)$ and $S(f)$ is the PSD at the frequency f. While $\mathcal{F}^{-1}$ is the inverse Fourier transform to return to the time domain.

For training, validation, and testing, the LSTM model must be provided with labels indicating whether a signal is present (1) or absent (0)

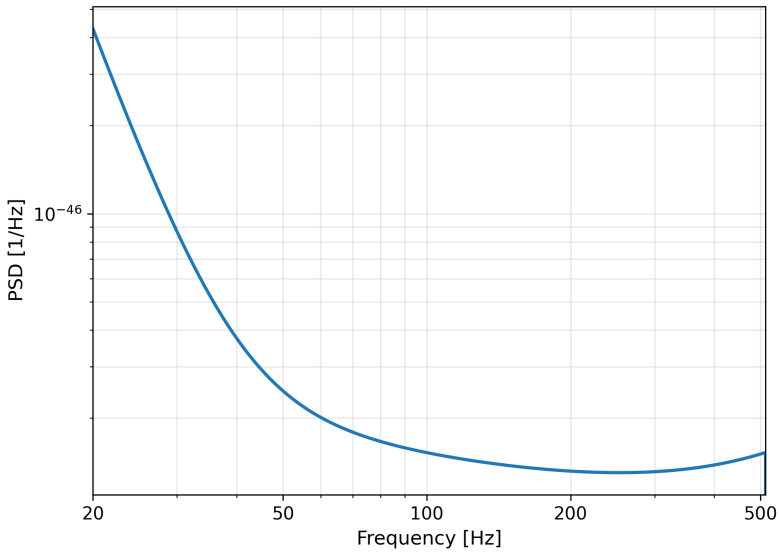


Figure 4.2: Power spectral density of the `aLIGOZeroDetHighPower` noise model used to generate the simulated Gaussian noise in this chapter. The curve is shown from the adopted lower-frequency cutoff of 20 Hz to the Nyquist frequency of 512 Hz, corresponding to the 1024 Hz sampling rate.

at each time step. We construct these labels based on the optimal CSNR of time series containing signals and the merger time of those signals. We define the optimal CSNR as:

$$\rho_{\mathrm{C}}^2(t) = \begin{cases} 2\int_0^t |h_{\mathrm{w}}(t')|^2\, dt' & \text{if } t \leq t_c \\ 0 & \text{otherwise} \end{cases}, \tag{4.10}$$

where $h_{\mathrm{w}}(t)$ is the whitened signal time series, and the CSNR is reset to zero following the merger at $t = t_c$. Since the LSTM produces a prediction every 64 time steps (a prediction rate of 16 Hz), we have 128 labels per time series in the training dataset and 192 per time series in the testing dataset. Time steps are labelled as not containing a signal (0) if $\rho_{\mathrm{C}} < 5$ and labelled as containing a signal (1) if ≥ 5. For time series not containing any signal, the optimal CSNR is zero, and all time steps are labelled as the noise-only class (0). For ROC curves analysis, we use a different labelling scheme where entire time series are given labels and not each time step, where the whole time series has an associated label,

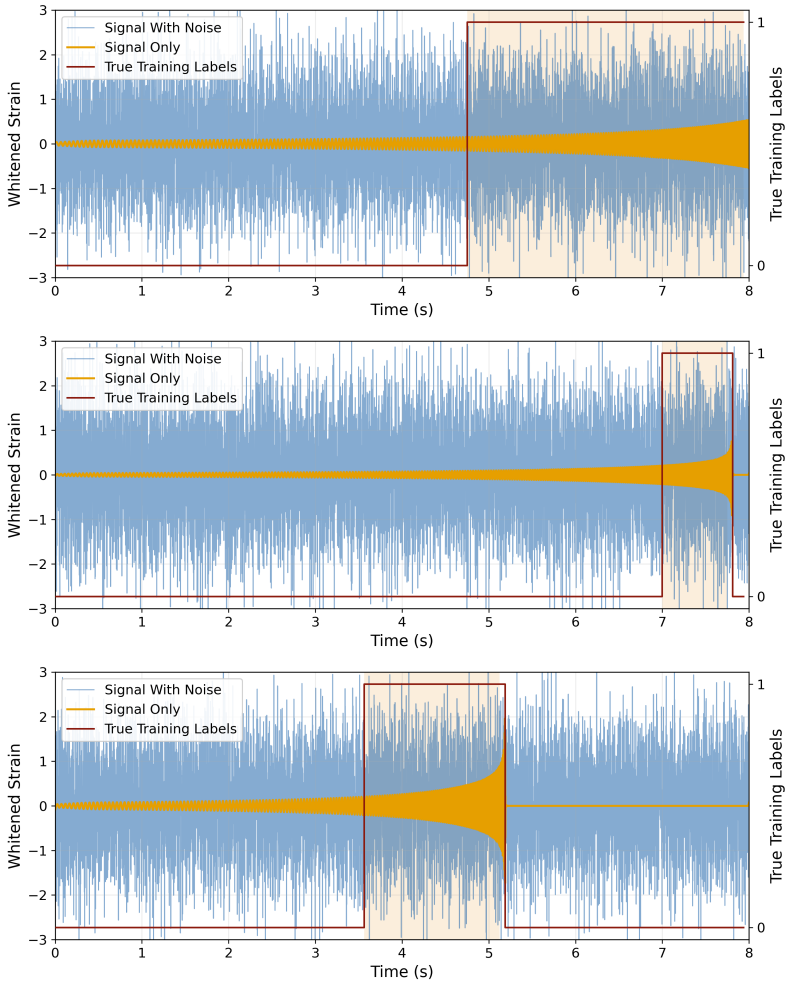


Figure 4.3: Example time series of simulated data used for training the LSTM model, showcasing various combinations of signal and noise. Each plot captures a unique scenario with signals merging in different regions of the time series and an example featuring a noise-only case. In each plot, the orange curve represents the whitened signal, the blue curve represents the whitened signal combined with whitened noise, and the purple curve indicates the training label (sampled at 16 Hz) defined via the optimal Cumulative Signal-to-Noise Ratio (CSNR), calculated using Eq. 4.10.

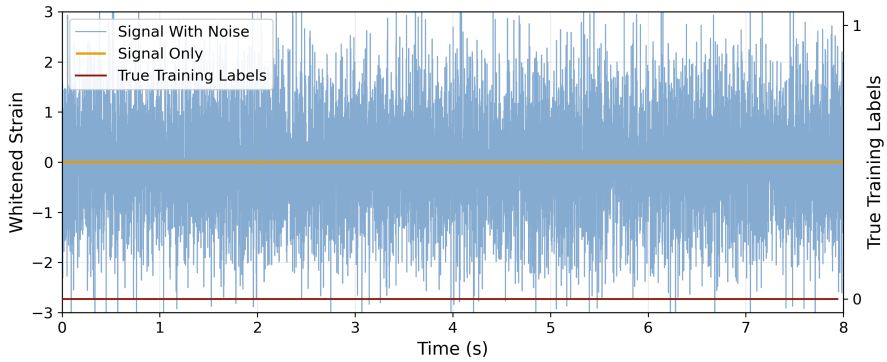


Figure 4.4: Example time series of simulated data used for training the LSTM model. This plot captures an example featuring a noise-only case. In the plot, the orange curve represents the whitened signal, the blue curve represents the whitened signal combined with whitened noise, and the purple curve indicates the training label (sampled at 16 Hz) defined via the optimal CSNR, calculated using Eq. 4.10.

(1) or (0), corresponding to time series containing or not containing any signal, respectively.

Additionally, 25% of our data contains only noise, and the rest contains signal plus noise. Data are first whitened and then normalised using the Z-score [159], where Z-score normalisation is a statistical technique employed to standardise the distribution of a dataset's features by adjusting their values to have a mean of zero and a standard deviation of one. Normalisation is indispensable for data pre-processing in ML, where it plays a crucial role in preparing data for analysis and optimising model performance. The training set consists of a 1 million time series, which includes either signals with noise or noise only. The validation set comprises 100,000 time series, and the test set contains 20,000 time series. This study uses simulated Gaussian noise from a single advanced LIGO detector and non-spinning BBH signals to provide a clear proof of concept. Real interferometer data contain non-Gaussian noise transients and spectral variations that may affect sensitivity. However, the model's lightweight architecture ($\sim 3.9 \times 10^4$ parameters) and time-domain design make it straightforward to adapt through domain adaptation, fine-tuning on real noise, and glitch-aware training.

4.4 The LSTM Model

In this section, we explore the design and implementation of a four-layer stacked LSTM model. We discuss the architecture and the training process, including optimisation strategies. We then present the testing phase, during which we evaluate the model across different scenarios. Finally, we discuss how to approximately compare the performance of our LSTM model with traditional matched filtering techniques.

4.4.1 The LSTM Design

Using the PyTorch [160] library, we adopt a stacked LSTM configuration comprising four layers, each using 32 neurons. The input dimension is 64, meaning that 64 time series samples are read as input to the network at each step. The final LSTM output is passed to a dense layer with 16 neurons and then an output layer of dimension 2. Since we are dealing with a classification problem, we use the Binary Cross-Entropy (BCE) with logits as the loss function. The BCE with the logits loss function measures an effective average difference between the predicted probabilities (after applying a sigmoid function to the network outputs) and the true labels in a binary classification scenario. Hence, the network's output statistics are unbounded but are converted to probabilities in the range $(0, 1)$ within the loss function, where the model outputs Z are logits, which are unbounded, i.e., $(-\infty, \infty)$. In order to compute the probability for each class, we apply the sigmoid function to the output, i.e., $\sigma(z)$, which bounds it from $(0, 1)$, where the $\sigma(z)$ is defined as:

$$\sigma(z) = \frac{1}{1 + \exp(-z)}. \tag{4.11}$$

Before our analysis, the data were split into training, validation, and test sets. Specifically, 90% of the data were used for training, and the remaining 10% for validation. A separate test set was used for final evaluation. When in operation, the model will iteratively take in input in blocks of 64 time series samples (equivalent to 16 steps over 1 second of data). Together with this input, the LSTM will combine information passed to it from the previous analysis step in the form of the cell state

memory and process this input through the LSTM network to obtain a prediction for the presence or absence of a signal at that step. It will also update its memory state for passing on to the subsequent step. Therefore, the primary output of the model is a lower-sample-rate (16 Hz) time series of predictions.

In building our model for sequence data analysis, we strategically chose four stacked LSTM layers to handle long-term data dependencies, enhancing its pattern recognition capability. Opting for the sigmoid function as the output layer aids in binary classification, providing probabilities for data categorization. We initiated the model with 32 neurons in the input layer, striking a balance between capturing complex patterns and avoiding overfitting, These decisions reflect our careful consideration of the model's complexity and operational feasibility. Our model is designed to reduce the number of trainable parameters, achieving an optimal balance between computational efficiency and model performance. While CNNs, widely used for CBC detection, often require millions of trainable parameters, our LSTM-based approach uses just 38705 parameters—a substantial reduction. This significantly reduces computational load and shortens training time without compromising detection accuracy. Furthermore, the streamlined architecture enhances the model's operational viability, making it lightweight. A key objective of our work is to achieve performance comparable to that of more complex models while using far fewer trainable parameters. This makes our approach particularly well-suited for real-time applications and resource-constrained environments, such as Field-Programmable Gate Array (FPGA) implementations, where computational efficiency is critical.

4.4.2 Training the LSTM Model

The LSTM model required 45 hours to fully train and was run on an NVIDIA GeForce RTX 3090 Family GPU with 24 GB of memory. The number of epochs used during the training is 100, with a batch size of 1000 and a learning rate of 1×10^{-4}. In our model, a minimum of 7×10^5 training time series were required to prevent overfitting, and to

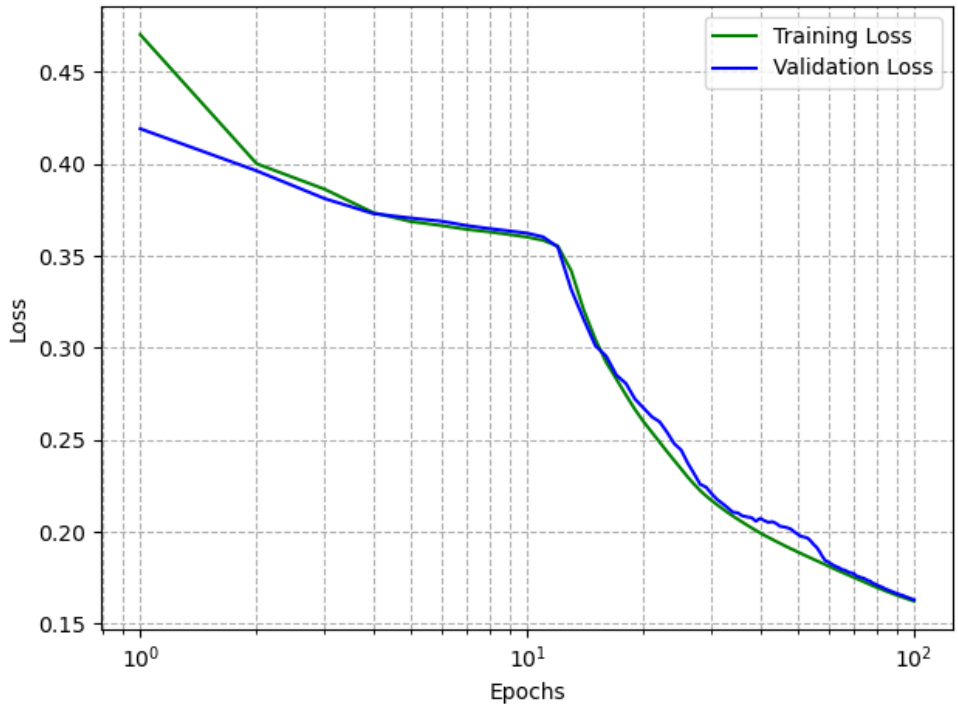


Figure 4.5: Evolution of the average training and validation BCE loss during training as a function of epoch.

obtain our most sensitive results, 1 million training time series were used. The model's performance was observed by analysing the loss curves for the training and validation datasets. The loss curves in Fig. 4.5 indicate effective training, with the training loss steadily declining, indicating that the model efficiently learns from the training set. Simultaneously, the validation loss curve descends in step with the training loss, with little deviation. This parallel fall shows that the model was not over-fitting to the training set and is a sign of its ability to generalise to unseen data. Training was stopped after 100 epochs (equivalent to $\sim$ 1.5 days of training time), although the training had not yet converged. Further training would likely improve the model, but those improvements would be minor given the final gradients of the loss curves.

The choices made regarding training hyperparameters were deliberate; for instance, the batch size was selected to balance the computational

load with the model's ability to generalise from the training data. The learning rate was chosen to ensure steady progress without overshooting the minima in the loss landscape. The training data volume was sufficient to represent the problem space without excessive training time or prohibitive memory requirements.

4.4.3 Testing the LSTM Model

To evaluate the efficacy of our model, we conducted a series of tests using an independent test dataset. These tests were designed to assess the model's performance using several metrics. The test dataset, distinct from the training and validation datasets, comprises 20,000 time series, each with a 12-second time series, and, for the signal population features, a uniform distribution of SNR ranging from 2 to 20. We used 20,000 time series to gather enough data to estimate FARs down to a level of $1\,\mathrm{day}^{-1} \equiv 1.157 \times 10^{-5}\,\mathrm{Hz}$. Each series has 192 labels, resulting in a total of 3,840,000 data points. After excluding the first 4 seconds, we are left with 2,560,000 data points, including 2,187,168 classified as noise (equivalent to 1.58 days) and 372,832 classified as signal. This level of background noise yields sufficient data points to achieve a FAR of 1 per day.

Reaching this FAR is essential for testing the practicality of our model in real-world scenarios. Based on rate predictions for BBHs [7], where it is expected that in the current 4^{th} advanced detector observing run observes $\mathcal{O}(100)$ such events, a FAR of 1 per day could be considered tolerable given the expected comparable true alarm rate. In early-warning analysis, we use the LSTM model to analyse test data operating with small time steps (1/16 second) as described in Subsec. 4.5.1. For each of these time steps, we calculate the LSTM output statistic. If this statistic exceeds a specific false alarm threshold, based on the predetermined distribution of the background, we classify it as an early-warning trigger. However, it's essential to note that we only label it as a trigger if it's the first time it has surpassed this threshold within a 12-second-long signal test data time series.

4.4.4 Early-Warning Comparison with Matched-Filtering

To provide an approximate benchmark for comparing our LSTM early-warning results, we have approximated the sensitivity of a non ML search method, similar to matched filtering. We calculate the CSNR using an exact template to provide early-warnings, akin to how matched filtering would accumulate this statistic in real time,and compute it using the optimal CSNR as defined in Eq. 4.10. Our approach also broadly equates realistic false-alarm thresholds to a set of specific CSNR values. As a result, we explore various CSNR thresholds to compare the LSTM model results (which are defined at specific FARs). We approximate the early-warning time as the duration between the merger and when the CSNR crosses these thresholds. This comparison aims to approximately assess the effectiveness of our approach in early signal-detection scenarios. This matched-filter comparison serves as an idealised baseline under controlled conditions and is intended only as a reference for the LSTM results.

4.4.5 Detection Sensitivity Comparison with Matched-Filtering

We also aim to approximately compare the raw sensitivity of the LSTM analysis with that of existing matched-filter methods (irrespective of the early-warning output). In this instance, we do not use existing matched-filtering algorithms for early-warning detection, such as those discussed in the previous chapter. Instead, we opt for a simpler comparison. We compared the LSTM results with a hypothetical scenario in which the exact mass template is known, without the need to search a template bank. Here, the detection statistic is the matched-filter SNR, computed using PyCBC [155] and maximised over an 8-second window containing the signal's true arrival time. In our exact template-matched-filter analysis, random templates from our training mass prior are generated when applying this single-template approach to instances of noise alone. To be consistent, we also maximise over the same 8-second window when obtaining the corresponding statistic from the LSTM analysis.

This approximate comparison sets an unrealistic upper bound for sensitivity, but it helps contextualise our results. As stated in the previous section, a similar method is used as a benchmark for the LSTM early-warning analysis, where we use a single true mass template for each test signal and compute the optimal CSNR as a function of time. Our aim in both cases is to determine whether our proof-of-principle LSTM analysis can approach current detection capabilities and early-warning times.

4.5 Results

After completing the training and validation steps, we evaluate the performance of the LSTM model on a test dataset, as described in Sec. 4.4. Our primary focus has been on two key goals: first, approximating the sensitivity of the LSTM model relative to the matched-filter approach; and second, investigating the effectiveness of LSTM models for detecting signals during the inspiral stage before the merger.

4.5.1 LSTM Model Detection Sensitivity

In our analysis, we compared the LSTM model to an approximate matched-filtering approach for detecting GW signals using ROC curves. These curves provide a visual comparison of how well each method identifies true positives as a function of false alarm thresholds. The ROC curve in Fig. 4.6 shows how the LSTM model and matched filtering compare in terms of sensitivity when applied to the testing dataset. Our matched-filtering approach, which we used with a single exact template, provides an unrealistic upper bound on sensitivity, since a full template-bank analysis would suffer from a higher FAR. The ROC curve for the LSTM model, still in early development, shows that at a fiducial FAR of 1 per day has a true alarm probability ~ 2 times lower than the upper-bound. We note that a rigorous matched-filter analysis using a complete template bank would likely yield results somewhere between the 2 ROC curves shown. The fact that the LSTM result is within a factor of 2 of the best-case sensitivity is important because the LSTM model could be further improved and offers faster, more efficient processing. In this

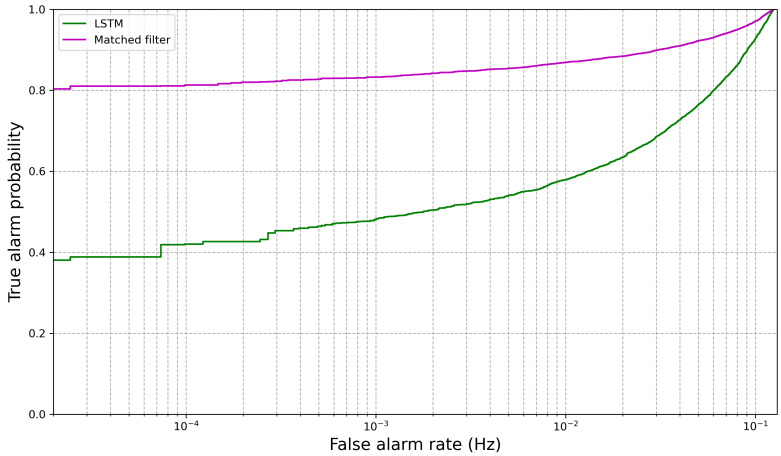


Figure 4.6: ROC curve comparing the detection sensitivity of the LSTM model (green line) with that of a matched filtering approach using an exact template (magenta line) where both statistics have been maximised over an 8-second window. The ROC curve demonstrates the true alarm probability as a function of the FAR. Note that the test dataset consisted of 14,900 cases of (noise + signal) with signal SNR uniformly distributed over [2, 20], and 5,100 cases of noise only.

analysis, the detection statistic is defined as the maximum output value (with a window of $[4, 12]$ seconds containing the merger time) produced by the matched filter and LSTM analyses, respectively, for each time series. This methodology aligns with the matching filtering approach. For this particular analysis, the dataset comprises 20,000 time series, of which 25% consist of noise only, while the remaining 75% contain signals. Since in this analysis each time series contains only noise or signal plus noise, the time series has an associated label of 0 or 1 (rather than each 1/16 second having a label based on the CSNR). This results in a total of data labelled as only noise as 0.46 days, limiting our ROC results to a minimum FAR of 2.5×10^{-5} Hz.

In addition to the ROC analysis, we report precision, recall, F1 score, and Precision-Recall Area Under the Curve (PR-AUC) to provide complementary insights into the model's behaviour. Precision measures the fraction of predicted positives that correspond to true signals, while recall quantifies the fraction of real signals that are successfully de-

tected. The F1 score balances missed detections and false alarms, and the PR-AUC summarises the overall quality of the precision–recall curve. On our test set, the LSTM achieves a precision of 0.798, recall of 0.544, F1 score of 0.647, and PR-AUC of 0.707. These metrics indicate that the model performs reliably in the early inspiral regime: it is selective and robust against noise-induced triggers, as reflected by its high precision and PR-AUC, while its moderate recall is consistent with the reduced signal power far from merger. Overall, the LSTM prioritises accurate detections while still identifying a substantial fraction of true signals. It is important to note that this comparison does not imply that the LSTM surpasses matched filtering. Instead, it serves to assess how the LSTM's sensitivity compares under simplified conditions, while highlighting its potential advantages in real-time, template-free processing.

4.5.2 Early-Warning Times from the LSTM Model

Here, we present the results of our early-warning analysis, which compares the performance of the LSTM model and CSNR for early-warning detection. As a proxy for comparing both analyses at common FARs, we instead provide comparative results assuming 3 different CSNR thresholds (6, 8, and 10) for the approximate matched-filter benchmark. This is because we have performed an unrealistic single-template matched-filter analysis and cannot therefore accurately estimate sensitivity at specific FAR values. In Fig. 4.7, we observe a significant positive correlation between the early-warning times calculated using the CSNR approximation and those determined by the LSTM model.

Most importantly, it is found that, in terms of early-warning capabilities, a CSNR threshold of 10 yields performance comparable to that of the LSTM. Moreover, it is noteworthy that while the LSTM model maintains a fixed FAR of one per day, the early-warning time provided by the LSTM is around half that provided by the CSNR approach at a CSNR threshold of 6, and it reaches roughly two-thirds at a threshold of 8. This comparison highlights that the LSTM model consistently provides earlier warnings across different CSNR thresholds. As shown in

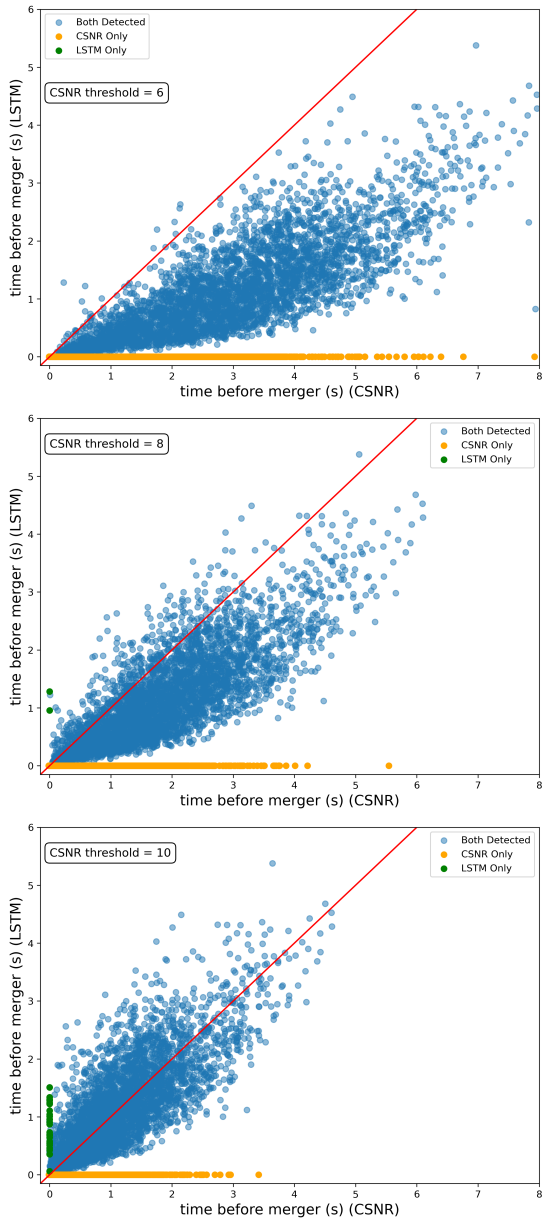


Figure 4.7: Early-warning detection analysis at a fixed FAR of one per day for the LSTM model: Plots show the performance of the LSTM model and CSNR with thresholds, set at 6, 8, and 10, respectively, for early-warning detection (time before merger). The axes report the time before merger(s): x-axis for CSNR and y-axis for the LSTM. Green dots signify early-warnings detected solely by the LSTM model, not identified by CSNR. Orange dots indicate early-warnings detected only by CSNR. Blue dots represent common detections by the LSTM model and CSNR.

Table 4.2: Comparison of the numbers of early-warning detections from the LSTM model at a false alarm threshold of one per day compared with detection at different CSNR thresholds (6, 8, and 10).

Threshold Value CSNR	6	8	10
Early-warning detected by LSTM and CSNR	11612	9984	8356
Early-warning detected by LSTM only	0	2	27
Early-warning detected by CSNR only	7849	6221	4593
Early-warning not detected by either LSTM or CSNR	3288	4916	6544
Ratio of events detected by LSTM over CSNR	0.60	0.62	0.65

Tab. 4.2, while the LSTM model detects fewer unique events compared to the CSNR approach, particularly at lower thresholds, its performance improves significantly at higher thresholds. At a CSNR threshold of 10, the number of early-warnings detected by both methods becomes more comparable, with 8,356 events detected in common. In this case, the LSTM model captures approximately 65% of the detections made by the rough CSNR method, demonstrating that the LSTM achieves reasonably strong performance despite its more selective detection strategy. This level of agreement, especially at higher thresholds, underscores the complementary nature of the two approaches and suggests that the LSTM model can serve as a reliable early-warning tool, even if it is more conservative.

The LSTM model can produce early-warnings up to 5.3 seconds before merger at a one-per-day FAR, depending on the SNR and the system's chirp mass. This early-warning is generated based on a specific threshold parameter selected to achieve a predetermined one-per-day FAR. Fig. 4.8 shows the relationship between the SNR and the achievable early-warning time across different chirp-mass ranges, with the black dashed curve marking the theoretical maximum time before merger extracted from our dataset and representing the upper limit on how early any method could detect the signals. This dashed curve is computed by identifying, for each SNR value, the maximum time before merger present in the dataset, providing an empirical envelope that no detection can physically exceed. The results reveal a consistent trend: higher SNR values lead to earlier detections that approach the theoretical limit,

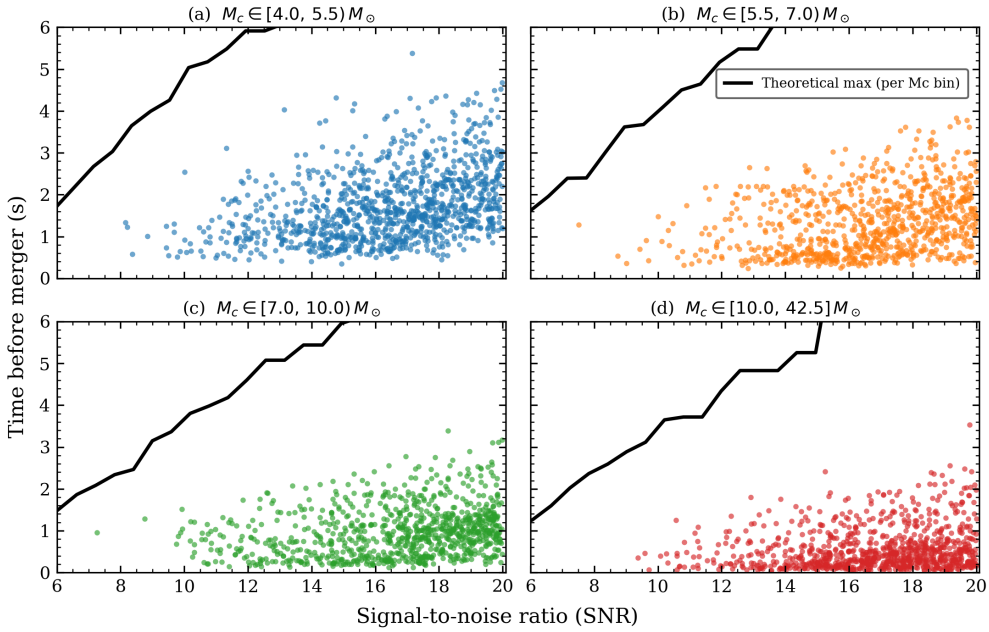


Figure 4.8: Illustrating the LSTM model's capability for early event warnings based on the signal, as a function of optimal SNR, this graph spans SNR values ranging from 6 to 20. The x-axis represents the SNR, while the y-axis denotes the amount of advance time the LSTM model can predict an event before it occurs. Additionally, the colours in the plot correspond to the chirp mass of the signals, providing a visual correlation between the mass and the model's predictive timing. These results correspond to a FAR of one per day. The black curve shows the theoretical maximum early-warning time derived from the true signal labels.

whereas low-SNR events cluster near zero lead time, indicating reduced sensitivity. As the SNR increases, the LSTM's detections progressively move closer to the envelope, illustrating that the model benefits directly from stronger signals but remains constrained by the theoretical early-warning boundary. The chirp mass further influences this behaviour by shifting the accessible SNR range: lower-mass systems typically achieve higher SNRs and therefore longer lead times, while higher-mass binaries populate lower-SNR regimes and yield shorter warnings, in agreement with the theoretical envelope.

The scatter plot in Fig. 4.9 shows the time before merger at which each signal is detected by the LSTM model, plotted as a function of

chirp mass. Each point corresponds to a simulated event and is coloured by its optimal SNR. The detection time is defined as the earliest moment at which the LSTM output exceeds the threshold associated with a FAR of one per day. To summarise the distribution of achievable early-warning times, the black curve shows the 95^{th} percentile of the theoretical early-warning time, computed from the true signal labels and a CSNR threshold, and evaluated within fixed chirp-mass bins. This percentile envelope provides an upper bound on the effective in-band signal duration available in the dataset, serving as a reference independent of the LSTM model's detection performance. A clear trend is evident in both the scatter points and the theoretical envelope: lower–chirp-mass binaries are detected several seconds before merger due to their long inspiral durations in the detector's sensitive band. As the chirp mass increases, the inspiral phase shortens, reducing the available pre-merger warning time and causing detections to cluster closer to coalescence. At high chirp masses ($\mathcal{M}_c \sim 20$–$40\,M_\odot$), the theoretical early-warning time falls below one second, consistent with the rapid merger timescales of these systems. Overall, the figure illustrates the range of early-warning times accessible within the dataset and their dependence on chirp mass.

Figures 4.10, 4.11, and 4.12 illustrate the performance of the LSTM model when applied to different test data time series. Specifically, Fig. 4.10 demonstrates the model's capability to detect the signal prior to merger. The LSTM model successfully identifies the signal approximately 2 seconds before the merger, with a FAR of one per day. At that point, the optimal CSNR is approximately half of the total, i.e., around 8, which highlights the model's ability to make early detections with only partial signal accumulation. The characteristics of this detected signal include an SNR of 16.52 and a chirp mass of 4.81 $M_\odot$, merging at 11.77 seconds within the time window under consideration. In a separate instance shown in Fig. 4.11, the model at a one-per-day FAR detects another signal roughly 3 seconds prior to its merger. At that point, the optimal CSNR is approximately half of the total, i.e., about 10, which again demonstrates the model's ability to make confident detections with only partial

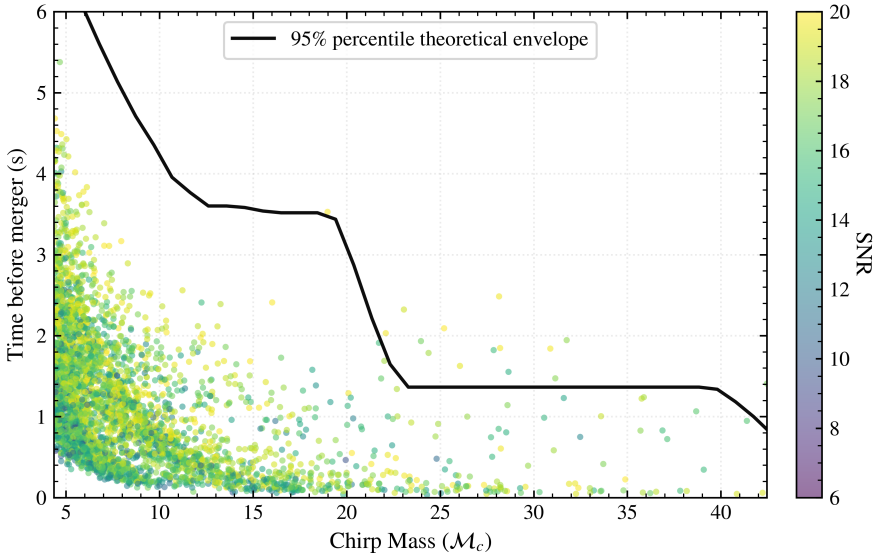


Figure 4.9: Time before merger as a function of chirp mass for all detected signals. Scatter points show individual events coloured by their optimal SNR. The detection time corresponds to the earliest moment at which the LSTM output exceeds the threshold associated with a FAR of one per day. The black curve shows the 95$^{\text{th}}$ percentile of the theoretical early-warning time, computed from the true signal labels in fixed chirp-mass bins.

signal information. This signal is distinguished by an SNR of 19.5 and a chirp mass of 5.17 $M_{\odot}$. In this case, the merger occurs at 12.15 seconds, falling outside the time window analysed. These examples underscore the LSTM model's efficacy in detecting temporal signals across varied observational windows. An additional feature of the LSTM output is that it does not increase monotonically as the signal approaches merger. Unlike the CSNR, which accumulates progressively as more signal power is observed, the LSTM detection statistic can rise and fall between successive timesteps. This reflects the fact that the network output is determined by both the local input and the information retained in the recurrent hidden state, rather than by a cumulative statistic constrained to increase with time. The observed fluctuations may therefore indicate that the model is responding to particular temporal features of the evolving waveform rather than simply tracking accumulated signal power. However, the internal representation learned by the network cannot be determined uniquely from the present analysis, and any detailed physical interpretation of these fluctuations remains speculative. Fig. 4.12 presents the performance of the LSTM model in recognising the absence of the signal, where only noise is present. It is evident from the LSTM output that, in the absence of a signal, the model's predictions remain low, indicating its effectiveness in differentiating between noise and actual signals. This capability is crucial for reducing false positives in signal detection and enhancing the model's reliability in practical applications. The small increases in the LSTM output observed in the noise-only example may arise for several reasons. One possibility is that random fluctuations in the Gaussian noise temporarily resemble features encountered during training, producing a weak response from the network. In addition, because the LSTM output depends on both the current input and information retained in its recurrent hidden state, small variations may also reflect the internal temporal dynamics of the model. Another possibility is that the finite range over which signals are positioned within the input time series introduces a temporal prior that the network may partially exploit, such that the model develops some sensitivity to the typical timing of injected

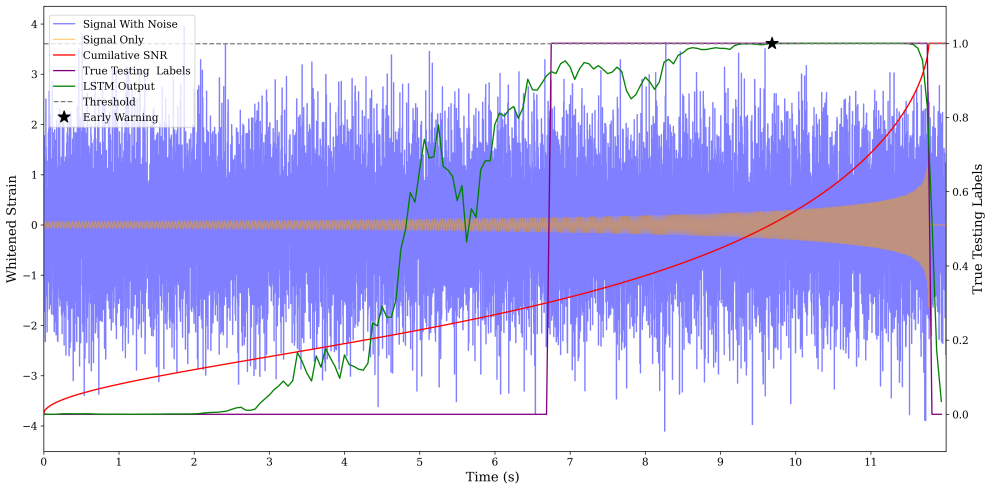


Figure 4.10: Example output of the LSTM model for a BBH test signal injected into Gaussian noise. The orange line shows the clean GW signal, while the blue line represents the corresponding whitened strain including noise. The green line denotes the LSTM output probability, and the gray dashed line marks the detection threshold corresponding to a FAR of one per day. The black star indicates the early-warning detection time, occurring approximately ~ 1.9 s before merger. The red line represents the CSNR, and the purple line shows the ground-truth labels identifying the true signal interval.

signals. Transient noise structure that resembles portions of the inspiral waveform may also contribute to these fluctuations. The present analysis does not allow these possibilities to be distinguished uniquely, and therefore the origin of the small output increases remains uncertain. A useful future validation test would be to evaluate the trained model on signals injected over a substantially wider and independently randomised range of times, in order to determine whether the network response remains primarily associated with the physical waveform morphology rather than with the typical temporal position of the signal within the input window.

The distribution of the LSTM model output statistic Z (prior to the BCE sigmoid activation to convert to a probability) with test data as input is presented in Fig. 4.13. Displaying the raw output provides a clearer view of the distribution's tails. The signal distribution exhibits

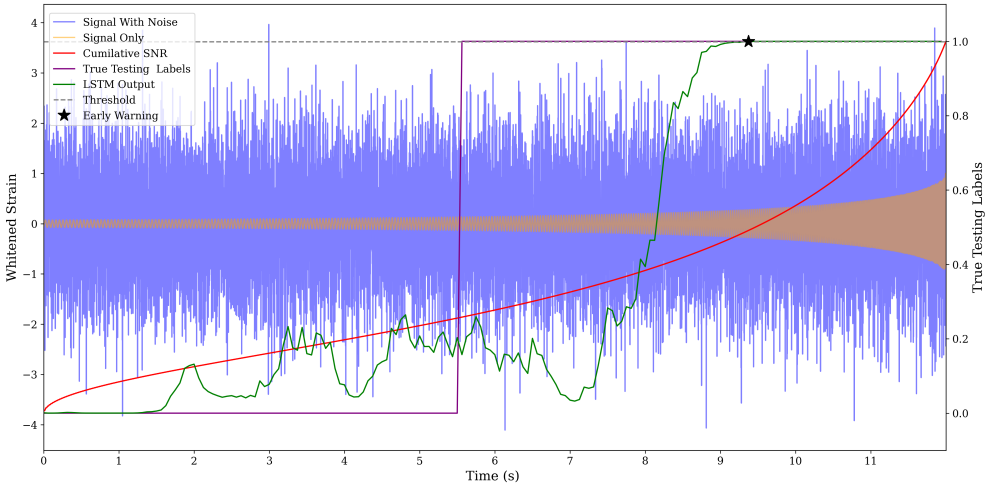


Figure 4.11: Example output of the LSTM model for a BBH test signal injected into Gaussian noise. The blue line shows the whitened strain containing both the GW signal and noise, while the orange line represents the true injected signal. The red curve indicates the CSNR, which increases as the merger approaches. The green curve shows the LSTM model's output probability for signal detection, and the purple line denotes the ground-truth labels used for supervised training. The dashed horizontal line represents the detection threshold corresponding to a FAR of one per day, and the black star marks the first threshold crossing — the model's early-warning detection, which occurs approximately ~ 3.4 seconds before merger.

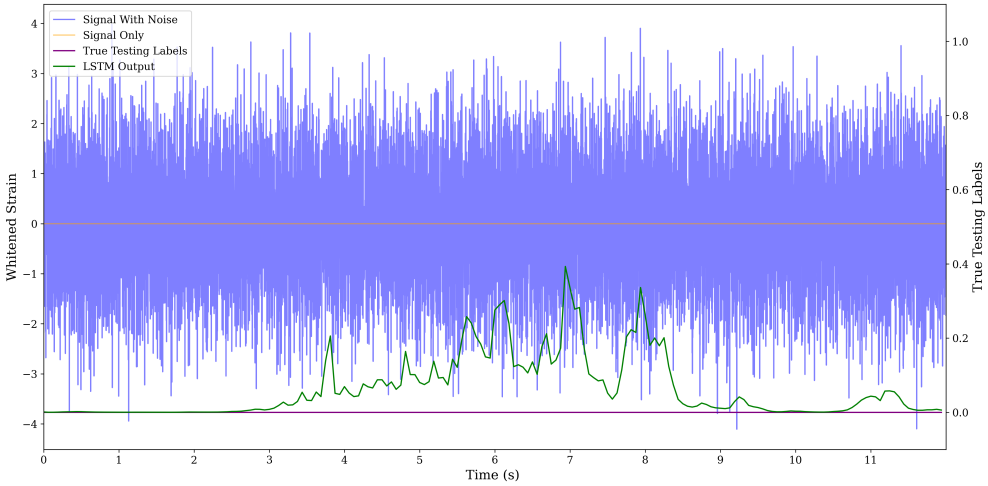


Figure 4.12: Example output of the LSTM model for a test case containing only Gaussian noise. The green line shows the LSTM output probability, which remains near zero throughout, indicating no detection at a FAR of 1 per day. The blue line represents the whitened strain (noise only), the orange line corresponds to the clean signal placeholder (no injected signal in this case), and the purple line marks the ground-truth labels.

a rightward shift, indicating higher values of the output statistic. In contrast, the noise distribution is shifted to the left, reflecting the model's effectiveness at distinguishing between signal and noise. The overlap in the central region suggests the presence of signals with low SNR, where the model's distinction between signal and noise becomes less pronounced and also indicates a subset of signals that are challenging to classify. These are instances where the signals are relatively weak and, hence, more susceptible to misclassification as noise. Conversely, the tail on the far right signifies signals with high SNR, which the LSTM model has identified with a high degree of confidence. These central cases may represent signals partially obscured by noise at the beginning of the signal time series. Yet, such events will yield robust detections where the signal's strength is pronounced enough to be clearly discernible from the noise.

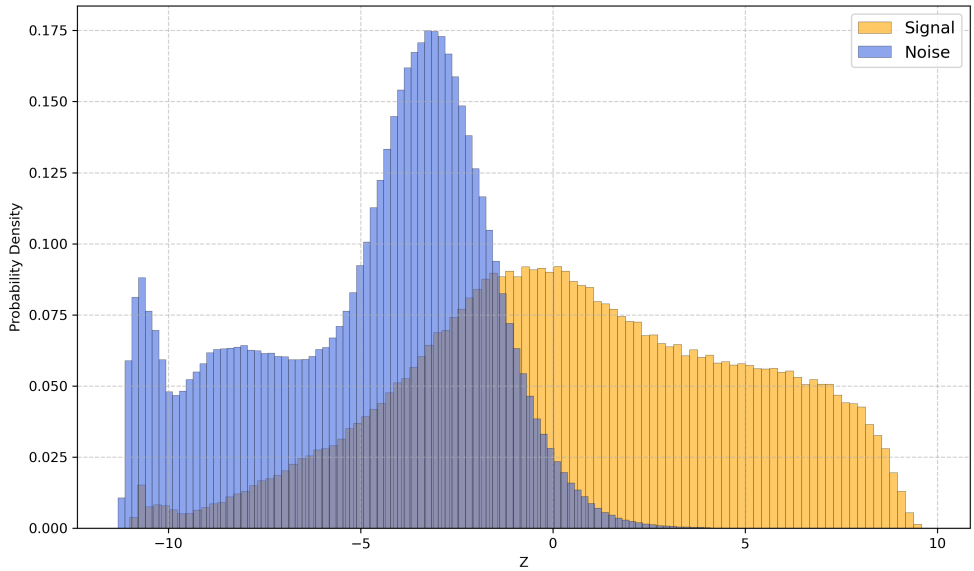


Figure 4.13: Distribution of the LSTM model output statistic Z for samples labelled as signal and noise in the test dataset. The blue histogram corresponds to noise samples, while the orange histogram corresponds to signal samples.

4.6 Conclusion and Discussion

In this chapter, we presented an LSTM-based early-warning system for BBH signals. We have demonstrated that BBH events can be identified even when the data contain only a portion of the early inspiral. The methodology we have described has the potential to be extended and applied effectively to detect various types of mergers, including those involving BNSs and NSBH systems. In summary, using LSTM networks for GW early-warning detection from BBH signals has yielded encouraging results. Early detection of GW signals is a crucial prerequisite for time-sensitive astronomical observations, and the LSTM model has demonstrated its ability to achieve this. One of the most significant benefits of using LSTMs is its efficiency in terms of computational resources. Unlike more complex models that may require substantial GPU memory, LSTMs are lightweight and can be run with relatively low memory overhead. This makes them particularly suitable for applications where low-

latency is crucial. With the ability to operate effectively in low-latency scenarios, LSTM networks can facilitate rapid data processing, which is essential for prompt scientific response and follow-up observations.

Additionally, the LSTM model performs exceptionally well in terms of flexibility across various data ranges. Once trained on a dataset, the model is fast to test across different data ranges and exhibits strong generalization. This flexibility allows quick adjustment to new data without requiring extensive retraining. This is very helpful in the dynamic field of GWs astronomy, where the swift detection and reporting of events can have significant scientific ramifications. For instance, in the context of multi-messenger follow-ups, the ability of LSTMs to promptly identify potential GWs signals can enable astronomers to coordinate observational resources across the EM spectrum and other cosmic messengers, such as neutrinos or cosmic rays. This coordinated approach can lead to a more comprehensive understanding of the events, further cementing the role of LSTM models as a potentially valuable asset in the era of time-domain astronomy.

The LSTM model introduces a prospective methodology for the early detection of GWs. Moreover, this model is especially promising for systems like BNSs, where there is a chance to detect EM or neutrino counterparts and a significantly longer early-warning period. The analysis presented in this chapter does not include several aspects that are important for extending the framework towards more realistic applications. These elements include integrating multiple detectors, accounting for spinning systems, and incorporating real detector noise. Moreover, incorporating real detector noise is critical for improving the model's practical applicability and resilience in real-world scenarios. When applied to real detector data, the presence of non-Gaussian noise and transient glitches is expected to introduce additional challenges. In particular, merger-mimicking glitches could transiently affect the network output, leading to delayed or false triggers. One practical mitigation strategy is to retrain the model on real data that includes a representative range of glitches, enabling the network

to learn to distinguish genuine inspiral signals from transient noise artifacts. Although only a limited number of labelled glitches are currently available for training, this limitation can be mitigated by leveraging existing labelled datasets such as those from the Gravity Spy project [9]. Following the approach adopted in SAGE [129], these labelled glitches could be incorporated to train the model on three distinct classes—noise, noise plus signal, and noise plus glitch—thereby enhancing its robustness to non-astrophysical transients and improving its overall reliability in real detector conditions.

The existing framework could be extended to include these elements. While this proof of concept is demonstrated for a single detector, the approach extends naturally to a detector network. One straightforward option is to run the LSTM independently on each detector and require time-coincident triggers, which reduces false alarms without retraining. A joint training approach to multi-detector data could further exploit antenna-pattern diversity to improve robustness. While this study focuses on BBH signals as a proof of concept, a natural extension of this work is to apply the framework to other GW sources, particularly BNS and NSBH systems, which are potential EM emitters. These sources introduce additional challenges due to their much longer inspiral phases, resulting in increased data volumes and computational demands for ML–based analyses. However, the lower-frequency nature of their early inspiral offers opportunities for narrow-band data compression and more efficient model training. The lightweight, recurrent design of the LSTM model makes it especially suitable for such applications, as it can process time series data sequentially and integrate information over long durations in real time. The extension of this framework to longer-duration, lower-mass systems is investigated further in Chapter 5, where the methodology is adapted to BNS signals.

A further extension of this framework would be low-latency sky localisation using data from multiple detectors. This could involve parallel or jointly trained LSTM networks that combine timing and amplitude information across the detector network to provide rapid, approximate

estimates of a source's position and uncertainty on the sky. In the longer term, the framework could be integrated with probabilistic PE methods, such as normalizing flows—which have been successfully applied in gravitational-wave inference [118] to produce low-latency posteriors for both sky location and other source properties. It may also be possible to leverage existing rapid sky-localization tools, such as BAYESTAR [40], which could be triggered independently by alerts from the LSTM, similar to the current GstLAL approach. For BNS or NSBH systems, the latency of BAYESTAR—typically a few to tens of seconds—would be on a similar timescale to the expected interval between the LSTM trigger and the merger, making it suitable for rapid sky localization immediately following an early-warning detection. Such an approach could provide a practical pathway toward a fully integrated, low-latency detection and localisation framework for MMA. The behaviour of the trained model under more realistic detector-noise conditions is investigated further in Chapter 6.

CHAPTER 5

Early-Warning Detection of Binary Neutron Star Systems Using Long Short-Term Memory Networks

5.1 Motivation and Overview

The early-warning detection of GWs is a central objective in the development of low-latency GW data analysis pipelines, as motivated by the requirements of MMA discussed in Chapter 1. ML-based approaches, as introduced in Chapter 2, have emerged as promising tools in this context, enabling the sequential processing of time series data and producing detection statistics in real time once trained. In Chapter 4, an LSTM–based framework was developed for the early-warning detection of BBHs. In this chapter, that framework is extended and adapted to BNS mergers, which represent a key target for early-warning detection in GW astronomy.

Early-warning detection is especially valuable for BNS systems, as highlighted by the role of low-latency detection in enabling multi-messenger follow-up discussed in Sec. 1.9. It may allow EM observatories to be alerted prior to a merger, increasing the likelihood of observing prompt and early-time emission. Such emission is often short-lived and can be challenging to capture without advance notice, particularly given the time required to localise the source and reposition telescopes [161]. The ability to identify a BNS inspiral before coalescence, therefore, has the potential to significantly enhance the scientific return of multi-messenger follow-up campaigns. Traditional matched-filtering-based pipelines have been adapted to provide early-warning alerts for BNS mergers; however, they remain computationally intensive because they require correlating the data with large template banks spanning a multi-dimensional parameter space [162], as discussed in Chapter 3.

This computational cost motivates the exploration of alternative and complementary strategies that can operate with low-latency and reduced computational overhead. ML–based approaches offer such a possibility,

provided they can robustly handle long-duration signals and low SNR conditions during the early inspiral phase. LSTM networks, introduced in Chapter 2, are particularly well suited to this problem, as they are designed to model sequential data and retain information over extended temporal contexts through their internal memory states [103]. This capability makes LSTMs a natural choice for capturing the slow, cumulative evolution of BNS inspirals in the time domain. However, applying LSTM-based models to long-duration inspiral signals poses specific challenges related to sequence length, class imbalance, and training stability, which must be addressed to achieve reliable early-warning performance.

In this chapter, the LSTM architecture introduced in Chapter 4 is retained and applied to BNS signals. Several adaptations are introduced to accommodate the longer inspiral duration and gradual signal build-up characteristic of BNS systems. These include adjustments to the temporal processing strategy to handle extended inspiral signals, while retaining the CSNR-based labelling strategy introduced in Chapter 4, which naturally captures the cumulative growth of inspiral signal power. In addition, the training duration is extended to ensure stable convergence when learning long-term dependencies across low-SNR data segments.

A key aspect of this work is the use of a significantly reduced sampling rate of 128 Hz, corresponding to a Nyquist frequency of 64 Hz. This choice is central to enabling the analysis of long-duration BNS inspiral signals within a computationally tractable framework. Quantitatively, a sampling rate of 128 Hz produces 12,800 samples for a 100-second input segment, compared with 25,600 samples at 256 Hz and 51,200 samples at 512 Hz. This reduces the sequence length by factors of two and four, respectively, thereby reducing the computational and memory requirements associated with processing long-duration BNS signals.

The corresponding Nyquist frequency of 64 Hz restricts the analysis to the low-frequency inspiral, excluding the higher-frequency signal content that becomes increasingly important closer to merger. This represents a deliberate trade-off between retaining information for early-warning detection and maintaining computational feasibility. By focusing

on the lower-frequency inspiral regime, the model is designed to investigate whether the gradual accumulation of signal information is sufficient for detection before coalescence, without requiring the high-frequency waveform evolution near merger.

Moreover, this approach is applicable to real detector data in a real-time setting. In practice, a low-pass filter can be applied to remove high-frequency components before downsampling to the desired rate, reducing the risk of aliasing. This provides a computationally efficient strategy for processing long-duration inspiral signals and supports potential real-time implementation.

This chapter is organised as follows. Sec. 5.2 describes the generation of simulated BNS signals and the construction of the training, validation, and test datasets. Sec. 5.3 presents the modifications to the LSTM model and training strategy required for long-duration inspirals. Sec. 5.4 outlines the training procedure and evaluation methodology. Sec. 5.5 presents the detection sensitivity and early-warning performance of the model on BNS signals, including an analysis of achievable warning times as a function of signal properties. Sec. 5.6 presents the detection efficiency of the model. Finally, Sec. 5.7 discusses the results in detail, examines the limitations of the present approach, and summarises the main conclusions of the chapter.

5.2 Generation of Binary Neutron Star Data

We simulate long-duration BNS signals and embed them in stationary Gaussian noise to train, validate, and test an early-warning LSTM detection model. The procedure parallels that used for the BBH analysis waveform generation, antenna projection, amplitude scaling by optimal SNR, addition of Gaussian noise, whitening, and labelling based on the CSNR, while adapting the waveform physics, parameter priors, sampling, and time windowing to the BNS regime.

5.2.1 Waveform Synthesis and Priors

Time-domain Inspiral-Merger-Ringdown (IMR) waveforms are generated using PyCBC [155, 156] and the IMRPhenomD_NRTidalv2 approximant, which extends a BBH phenomenological model with tidal phase corrections appropriate for neutron stars [18]. Each simulated signal is projected onto the H1 detector response. In this chapter, the analysis is restricted to the H1 detector to evaluate the performance of the LSTM model on long-duration, single-detector strain time-series data. Extension to a multi-detector network is beyond the scope of this thesis and is therefore reserved for future work. The detector response is defined as:

$$h(t) = F_+(t;\alpha,\delta,\psi)\, h_+(t) + F_\times(t;\alpha,\delta,\psi)\, h_\times(t), \tag{5.1}$$

where F_+ and $F_\times$ are the antenna pattern functions. The prior distributions used to generate the simulated BNS signals are summarised in Table 5.1. The priors for the extrinsic parameters—right ascension α, declination δ, polarisation ψ, inclination ι, and coalescence phase ϕ_c—are the same as those used in Chapter 4.

The component masses are sampled within the BNS range, $1.0M_\odot \leq m_2 \leq m_1 \leq 2.5M_\odot$, implemented by first drawing $m_1 \sim \mathcal{U}(1.0, 2.5)$ and then sampling $m_2 \sim \mathcal{U}(1.0, m_1)$ to enforce the ordering $m_1 \geq m_2$. The corresponding total mass and symmetric mass ratio are $M = m_1 + m_2$ and $\eta = m_1 m_2/M^2$, respectively. To control the overall signal amplitude, a target value of the inspiral SNR, $\rho_{\rm insp}(t_{\rm end})$, is randomly drawn. Here, $\rho_{\rm insp}(t)$ denotes the inspiral SNR accumulated up to time t, corresponding to the portion of the signal present within the analysed time series. The waveform amplitude (equivalently, the source distance) is then rescaled such that the accumulated SNR matches this target value. The inspiral SNR is defined using the standard optimal matched-filter inner product:

$$\rho^2_{\rm insp}(t) = 4\int_{f_{\rm low}}^{f(t)} \frac{|\tilde{h}(f')|^2}{S_n(f')}\, df', \tag{5.2}$$

where $\tilde{h}(f)$ is the Fourier transform of the waveform, $S_n(f)$ is the one-sided noise PSD of the detector, and $f(t)$ denotes the instantaneous GW

Table 5.1: The prior distributions used for the BNS signal parameters. The distance is adjusted to achieve the chosen inspiral SNR, ρ_{insp}.

Parameter	**Distribution**	**Min.**	**Max.**
Mass 1 (m_1)	Uniform	$1.0\,M_\odot$	$2.5\,M_\odot$
Mass 2 (m_2)	Conditional Uniform	$1.0\,M_\odot$	m_1
Right Ascension (α)	Uniform	0	2π
Sine Declination ($\sin\delta$)	Uniform	-1	1
Cosine Inclination ($\cos\iota$)	Uniform	-1	1
Polarisation (ψ)	Uniform	0	π
Coalescence Phase (ϕ_c)	Uniform	0	2π
Time to 64Hz (t_{64}) (Training,Validation, Testing)	Uniform	101 s	102 s
ρ_{insp} (Training + Validation)	Uniform	7	20
ρ_{insp} (Testing)	Uniform	2	20

frequency at time t. A lower cutoff frequency of $f_{\text{low}} = 20\,\text{Hz}$ is adopted. This choice provides a practical compromise between retaining as much of the early inspiral signal as possible and avoiding the rapidly increasing detector noise at lower frequencies. Below this frequency, the sensitivity of ground-based interferometers degrades significantly due to low-frequency environmental and instrumental noise, making the signal progressively more difficult to recover reliably.

In the present analysis, the waveform is generated at a sampling rate of 128 Hz, corresponding to a Nyquist frequency of 64 Hz. This defines the maximum frequency up to which the inspiral can be represented. However, since the signal is embedded in a finite-duration time series and randomly distributed within the data, it does not, in general, evolve to this limit. Accordingly, the SNR is computed as an inspiral SNR, $\rho_{\text{insp}}(t)$, accumulated over the portion of the signal present within the analysed time series. The upper limit of the integral is therefore determined by the smaller of the frequency reached by the signal within the available time window and the Nyquist frequency. In contrast to the analysis presented

in Chapter 4, where the SNR corresponds to the full waveform, $\rho_{\rm insp}(t)$ represents a time-dependent SNR accumulated from the inspiral portion of the signal, and does not include contributions from later stages of the waveform evolution. As a result, $\rho_{\rm insp}(t)$ corresponds to a band-limited SNR accumulated from the inspiral portion of the signal present within the analysed time series, excluding higher-frequency contributions from the late inspiral and merger phases, and therefore does not represent the full SNR up to coalescence.

To enable a consistent comparison of signals detected at different stages of the inspiral, an estimated total SNR, $\rho_{\rm tot,est}$, is defined. This provides an approximate measure of the full signal strength at merger, inferred from the inspiral SNR, $\rho_{\rm insp}(t)$, accumulated over the portion of the signal present within the analysed time series, and compensates for the truncated nature of the observed signal. The estimated total SNR is obtained by extrapolating the inspiral SNR, $\rho_{\rm insp}(t)$, to higher frequencies using the leading-order inspiral scaling $|\tilde{h}(f)|^2 \propto f^{-7/3}$ [2, 21]:

$$\rho^2_{\rm tot,est} = \rho^2_{\rm insp}(t_{\rm end}) \frac{\displaystyle\int_{f_{\rm low}}^{f^{\rm ext}_{\rm high}} \frac{f^{-7/3}}{S_n(f)}\, df}{\displaystyle\int_{f_{\rm low}}^{f(t_{\rm end})} \frac{f^{-7/3}}{S_n(f)}\, df}, \tag{5.3}$$

where $f(t_{\rm end})$ denotes the maximum frequency reached by the signal within the analysed time series, and $f^{\rm ext}_{\rm high}$ is an extended upper cutoff frequency chosen to approximate the inspiral contribution up to merger. This rescaling accounts for the additional SNR expected from higher-frequency portions of the inspiral that are absent in the truncated signal. This approach does not include contributions from the merger and ring-down phases. However, for BNS systems, the dominant contribution to the total SNR arises from the inspiral, and the omission of these later phases introduces only a small correction.

While it would in principle be possible to generate the full waveform at a higher sampling rate (e.g. 4096 Hz) and compute the total SNR directly, this is not adopted here in order to maintain consistency with

the data representation used for training and inference. In particular, the analysis is performed on 128 Hz time series, and the estimated SNR is therefore constructed from the information available within this band-limited and truncated signal. In this way, $\rho_{\mathrm{tot,est}}$ serves as a consistent proxy for the full signal strength, allowing events to be compared on a common scale despite only the inspiral portion of the signal being present within the analysed data. During training, the target inspiral SNR at the end of the training segment, $\rho_{\mathrm{insp}}(t_{\mathrm{end}})$, is drawn from a uniform distribution in the range $[7, 20]$. For testing, $\rho_{\mathrm{insp}}(t_{\mathrm{end}})$ is drawn from a wider range, $[2, 20]$, in order to evaluate the model performance in the low-SNR regime.

5.2.2 Sampling, Windowing, and Signal Alignment

BNS inspirals enter the detector band gradually, with the signal amplitude initially small and increasing over time as the system evolves towards merger. In this work, the waveform is generated with a duration of 103 seconds at a sampling rate of 128 Hz, corresponding to a Nyquist frequency of 64 Hz. The waveform generation is configured such that the time at which the signal reaches 64 Hz lies between 101 and 102 seconds. A fixed 100-second window is then extracted for training and evaluation. As a result, the analysed segment always terminates before the waveform reaches 64 Hz, ensuring that the model is trained exclusively on the early inspiral portion of the signal and does not include the late-time evolution close to this frequency.

The merger time is computed theoretically from the waveform model parameters. It is not used during preprocessing or model training, but is retained for subsequent analysis and visualisation. In particular, it provides a reference for interpreting the temporal evolution of the signal and for annotating results in post-processing. This configuration provides a clean and consistent input length of 100 seconds (12,800 samples at 128 Hz) for each training example.

5.2.3 Noise Generation and Whitening

Each simulated signal is injected into stationary Gaussian noise drawn from an empirical average H1 PSD, provided as a frequency–PSD table. The PSD is interpolated to match the discrete frequency bins corresponding to the chosen sampling rate and segment length. Gaussian noise realisations are generated using the `noise_from_psd` routine provided by the PyCBC library [155]. A fixed fraction (25%) of all examples are noise-only, while the remaining contain signal-plus-noise, consistent with the BBH configuration. Both signal and noise are whitened following the procedure described in Chapter 4 (see Eq. 4.9). This involves dividing the Fourier transform of the strain by the square root of the noise PSD and transforming back to the time domain. A Tukey window is applied in the time domain to reduce edge effects, with a tapering parameter of $\alpha = 0.1$. Each example is then standardised using Z-score normalisation [159].

The empirical average H1 PSD used to generate the stationary Gaussian noise in this investigation is shown in Fig. 5.1. The spectrum exhibits a frequency-dependent noise level and narrow spectral features inherited from the underlying detector noise estimate. The PSD is shown over the 20–64 Hz frequency range relevant to the 128 Hz sampled data, with 64 Hz corresponding to the Nyquist frequency.

The whitening and CSNR definitions follow the same procedure introduced in Chapter 4. The key difference in this chapter is that the waveform is generated at a reduced sampling rate. This fundamentally restricts the analysis to the early inspiral portion of the signal. In contrast to the approach presented in Chapter 4, the high-frequency evolution of the waveform in the final few seconds prior to merger is excluded by construction. This is a deliberate design choice motivated by the goal of early-warning detection. Signals that provide less than ~ 10 seconds of advance warning are of limited practical value, and therefore the exclusion of this late-time information represents an acceptable and intentional trade-off. At the same time, limiting the frequency content in this way

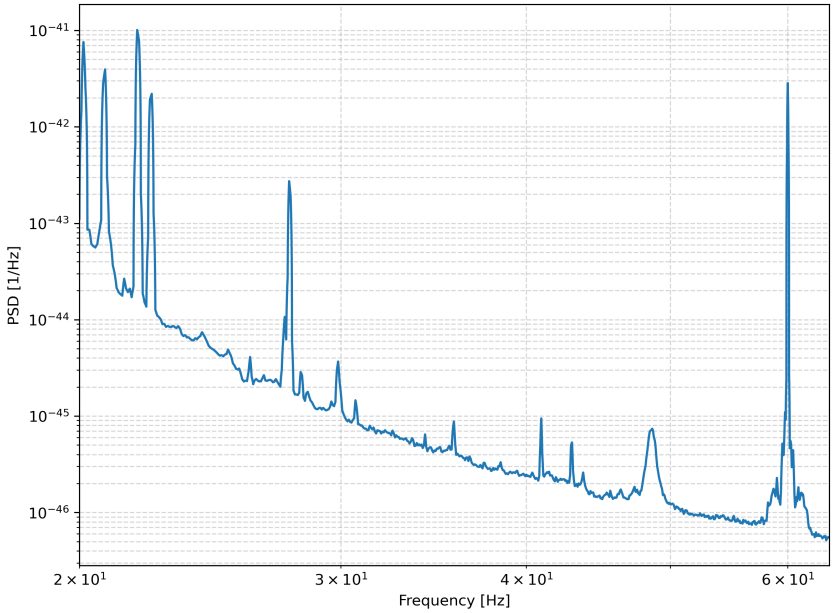


Figure 5.1: Empirical average H1 noise power spectral density used to generate stationary Gaussian noise for the simulations. The PSD is shown over the 20–64 Hz frequency range relevant to the 128 Hz sampled data, with 64 Hz corresponding to the Nyquist frequency.

is essential for enabling the efficient processing of long-duration BNS signals. As a result, the analysis focuses on the gradual buildup of signal power during the early inspiral phase, rather than on the short-duration, high-amplitude features near merger.

5.2.4 Labelling via CSNR

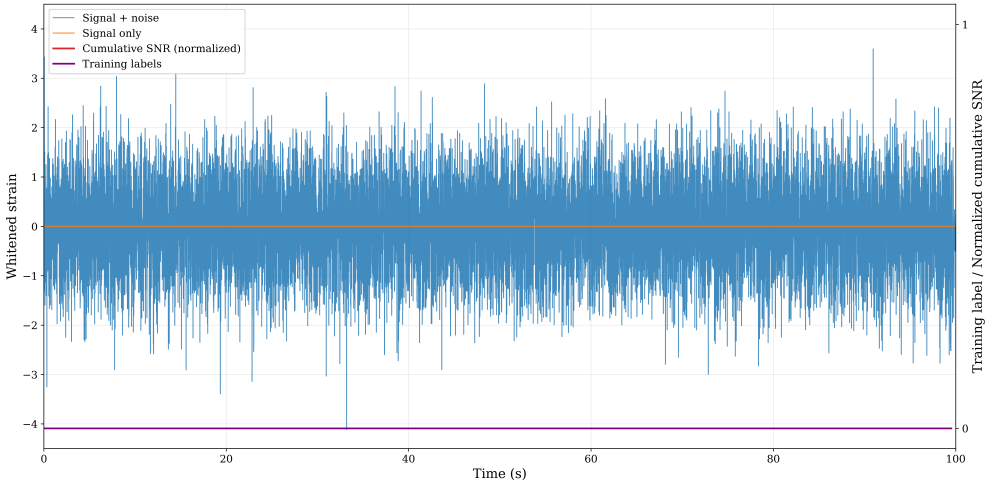
Following the procedure adopted for the BBH analysis, the CSNR is defined as in Chapter 4 (see Eq. 4.10). In the present analysis, only pre-merger data are considered, and therefore the CSNR is evaluated exclusively for $t < t_c$, corresponding to the inspiral portion of the signal. In this work, the CSNR is equivalent to the inspiral SNR accumulated up to time t, i.e. $\rho_{\mathrm{C}}(t) \equiv \rho_{\mathrm{insp}}(t)$. This quantity represents the accumulated signal power up to time t and provides a physically motivated measure of the gradual build-up of inspiral signal strength prior to merger. Since the analysed data are restricted to pre-merger segments ($t < t_c$), this condition is always satisfied and is therefore omitted in the label definition. Per-timestep binary labels are assigned using a fixed CSNR threshold of $\rho_{\mathrm{insp}}(t) \geq 5$:

$$y(t) = \begin{cases} 1, & \text{if } \rho_{\mathrm{insp}}(t) \geq 5 \\ 0, & \text{otherwise.} \end{cases} . \tag{5.4}$$

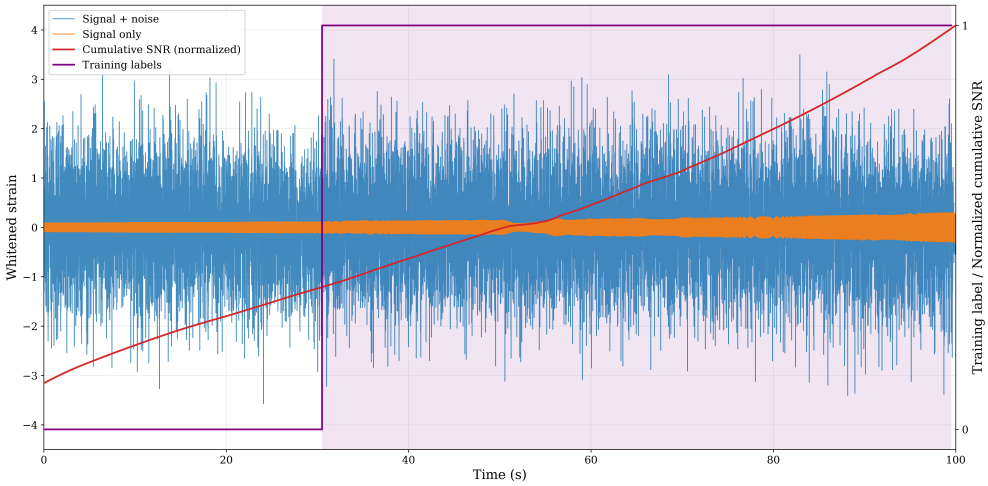
This procedure produces a binary label sequence across the 100-second pre-merger window at a labelling cadence of 2 Hz (one label every 0.5 s). This cadence is determined by the sampling rate of 128 Hz and the temporal resolution of the model outputs, with each label corresponding to 64 input time series samples. This results in 200 per-timestep labels for each training example. In addition to the time-resolved labels, a segment-level label is assigned to each 100-second data segment for event-based ROC evaluation, with 1 corresponding to signal-plus-noise and 0 corresponding to noise-only examples.

Representative examples of noise-only and signal-plus-noise training time series are shown in Fig. 5.2. These examples illustrate the structure of the whitened input data, the evolution of the inspiral SNR, $\rho_{\mathrm{insp}}(t)$, and the CSNR-based labelling scheme used for supervised training. In

panel (a), a noise-only example is shown. The whitened strain consists of stationary Gaussian noise, and the signal-only contribution is absent. The inspiral SNR, $\rho_{\text{insp}}(t)$, remains zero throughout the 100-second window, as no signal is present. Consequently, the CSNR does not exceed the predefined threshold, and all per-timestep labels remain zero. This example demonstrates the expected background behaviour, with all per-timestep labels remaining zero in the absence of any signal. In panel (b), a signal-plus-noise example is presented. The inspiral waveform is visible within the noisy background and gradually increases in amplitude as the system approaches merger. The inspiral SNR, $\rho_{\text{insp}}(t)$, correspondingly increases over time and eventually exceeds the threshold value of $\rho_{\text{insp}}(t) = 5$, at which point the per-timestep labels transition from zero to unity. This transition reflects the accumulation of sufficient signal power to justify a detection decision. The figure, therefore, illustrates how the labelling scheme encodes the progressive build-up of inspiral evidence prior to merger.



(a) Noise-only example. The inspiral SNR, $\rho_{\mathrm{insp}}(t)$, remains below the threshold throughout the window, and all per-timestep labels remain zero.



(b) Signal-plus-noise example for a representative BNS system with chirp mass $M_c = 1.09\,M_\odot$ and inspiral SNR $\rho_{\mathrm{insp}}(t_{\mathrm{end}}) = 13.13$. The inspiral SNR increases gradually during the inspiral phase and crosses the predefined threshold, after which the per-timestep labels switch to unity.

Figure 5.2: Representative 100-second BNS pre-merger training examples after whitening. The blue trace shows the strain data (signal plus noise), the orange trace shows the signal-only contribution, and the red curve indicates the normalised inspiral SNR, $\rho_{\mathrm{insp}}(t)$ (equivalently, the CSNR), used to assign per-timestep labels (shown in purple). These examples illustrate the CSNR-based labelling strategy used for supervised training of the LSTM model.

5.2.5 Dataset Composition and Splits

The training and validation datasets are generated using the same procedure, with the validation set corresponding to 10% of the total generated data. The training set comprises 100 files, each containing 10,000 time series of 100 s duration, yielding a total of 10^6 training examples. The validation set comprises 10 files of equal size, totaling 10^5 examples. The test set is generated separately and consists of 20,000 time series of 100 s duration, distributed across 2 files, each containing 10,000 examples. In contrast to the training data, which are restricted to signals capable of exceeding the CSNR threshold used for labelling, the test set includes signals extending to lower inspiral SNR, $\rho_{\mathrm{insp}}(t_{\mathrm{end}})$. This enables a systematic evaluation of model performance in the weak-signal regime.

All datasets preserve a 25% noise-only fraction at the segment level, meaning that 25% of the time series contain no injected signal. The CSNR threshold, corresponding to $\rho_{\mathrm{insp}}(t) = 5$, is selected to define a meaningful level of accumulated signal power above the noise background, ensuring reliable and stable label assignment. Signals with lower SNR would fail to exceed this threshold and would therefore be indistinguishable from noise, making them unsuitable for supervised training.

5.3 LSTM Model Architecture and Training Adaptations

5.3.1 Base Architecture and Temporal Output

The LSTM architecture introduced in Chapter 4 is retained in this analysis. This choice is motivated by the use of a lower sampling rate in the BNS setup, which results in longer input sequences ($128 \times 100 =$ 12,800 samples) compared to the BBH analysis in Chapter 4 ($1024 \times 8 =$ 8,192 samples), without increasing the intrinsic complexity of the data representation. In addition, the signal space for BNS systems is expected to be of lower complexity, due to the narrower mass range and the restriction to the inspiral phase, which lacks the highly non-linear dynamics present near merger in BBH systems. Consequently, the existing LSTM

architecture is sufficient to capture the relevant temporal features of the signal without increasing model complexity.

The model consists of stacked recurrent layers followed by a fully connected output layer that produces a probability estimate at each time step. For BNS inspirals, the network is trained on long-duration input sequences corresponding to a fixed 100-second time window. During training (and validation), each input time series is discretised into 200 temporal steps, corresponding to a labelling cadence of 0.5 seconds. This discretisation is specific to the supervised training setup. At inference time, the model can be applied to input sequences of arbitrary length, rather than being restricted to fixed-duration segments. This temporal resolution is determined by the 128 Hz sampling rate and the LSTM step size, such that each step corresponds to 64 input time series samples. As a result, the network produces a sequence of 200 probability outputs for each input example, enabling time-resolved detection decisions throughout the inspiral phase. This design allows the model to issue an early-warning detection at any time step prior to merger once the detection threshold is exceeded. The 0.5-second cadence defines the fundamental temporal resolution of the analysis and therefore sets a minimum detection latency. However, since the primary objective is to provide early-warnings on timescales of tens of seconds prior to merger, this temporal resolution is more than sufficient and does not impose a practical limitation on the achievable warning time.

5.3.2 Learning Rate Scheduling Strategies

Training deep recurrent networks on long-duration, low-SNR data presents optimisation challenges, particularly with respect to convergence stability and false-alarm control. To address this, several learning rate strategies were evaluated during model development, including a fixed learning rate, a step-based learning rate schedule, and a cosine annealing learning rate schedule. For the fixed learning rate configuration, a constant learning rate of 1×10^{-4} was used throughout training. In the

step-based schedule, the learning rate was periodically reduced by a fixed decay factor after a specified number of epochs.

For the cosine annealing schedule, training began with an initial learning rate of 1×10^{-4}, which was gradually reduced according to a cosine decay schedule to a minimum learning rate of $\eta_{\min} = 1 \times 10^{-6}$ over the course of training. Each strategy was assessed based on training stability, validation loss behaviour, and detection performance at low FARs. While fixed and step-based learning rates achieved reasonable convergence, they exhibited less stable loss evolution and reduced sensitivity in the low false-alarm regime. In contrast, the cosine annealing learning rate schedule produced smoother convergence and consistently improved performance at low FARs. This behaviour is particularly important for early-warning detection, where maintaining a low FAR is critical for practical deployment. Based on these observations, cosine annealing was selected as the learning rate schedule for all subsequent training runs, as it consistently exhibited more stable training behaviour and improved sensitivity at low FARs compared to the alternative schedules.

5.3.3 Training Duration and Convergence

The BNS inspiral signals considered here are represented as long input sequences in terms of the number of samples, over which the signal amplitude evolves gradually from low values. As a result, the informative features are distributed across extended regions of the sequence rather than concentrated near a single point. This requires the network to capture dependencies across multiple parts of the input sequence, particularly in the low-SNR regime where the signal remains weak relative to the noise. To ensure adequate convergence, the model was trained for more epochs than in the BBH configuration. Several training durations were explored, ranging from 150 to 350 epochs. Training stability and validation loss behaviour indicated that extending the training duration, measured in terms of the number of epochs, improved convergence without signs of overfitting. Based on these tests, a duration of 350 epochs was adopted for the final BNS model. Early stopping based on the validation loss was

applied to prevent unnecessary overtraining while allowing the network to capture long-duration inspiral features effectively.

Fig. 5.3 shows the evolution of the training and validation loss for the selected configuration using cosine annealing. The loss decreases steadily across epochs, and the validation curve closely follows the training curve throughout optimisation. A small but noticeable generalisation gap between the training and validation loss is observed towards the end of training. While this does not indicate strong overfitting, it suggests that additional training data could help reduce this gap and improve generalisation performance.

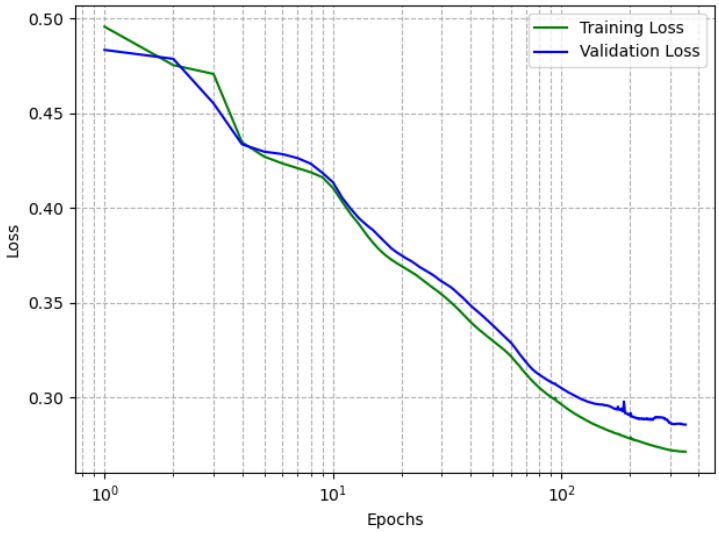


Figure 5.3: Training and validation loss as a function of epoch for the BNS LSTM model trained using cosine annealing learning rate scheduling. Both curves decrease steadily during training, indicating that the model is learning progressively. The validation loss closely tracks the training loss throughout training, suggesting consistent learning behaviour and no clear evidence of significant overfitting.

It is also important to note that cosine annealing affects late-stage training dynamics. As the learning rate approaches its minimum, the optimisation process slows, leading to a plateau in the loss curves. Consequently, it is not possible to conclude from these results alone that further training would not yield additional improvements in performance. These

observations highlight a trade-off between training duration, dataset size, and computational cost. Increasing the amount of training data or extending the training duration may improve performance, but at the expense of longer training times. In comparison to the BBH case presented in Chapter 4, the training behaviour is broadly similar, although the smoother evolution of the BNS signals leads to more gradual convergence and a less pronounced separation between training and validation losses.

5.3.4 Summary of Model Adaptations

In summary, the core LSTM architecture is retained, while targeted adaptations are introduced to accommodate long-duration BNS inspirals. Crucially, using a reduced sampling rate enables efficient processing of these extended signals by focusing on the low-frequency inspiral regime most relevant for early-warning detection. These adaptations include an increased number of temporal steps to enable time-resolved early-warning alerts, the adoption of a cosine annealing learning rate schedule to improve optimisation stability and performance at low FARs, and extended training durations to support learning across long inspiral sequences.

5.4 Training Procedure and Evaluation Methodology

This section describes the LSTM model's training configuration and the methodology used to evaluate its early-warning detection performance on BNS inspiral signals.

5.4.1 Training Setup

The LSTM model is trained in a supervised manner using the simulated BNS dataset described in Sec. 5.2. The dataset is divided into training, validation, and test subsets, with the test set reserved entirely for final performance evaluation.

Model optimisation is performed using the Adam optimiser, which is well-suited to training deep recurrent networks on noisy time series data. The loss function is BCE, computed between the predicted per-timestep probabilities and the ground-truth labels derived from the CSNR criterion

described in Subsec. 5.2.4. Each training batch contains a mixture of signal-plus-noise and noise-only examples, preserving the 25% noise-only fraction defined at the segment level during data generation. At the label level, however, the dataset is inherently imbalanced, as only a subset of timesteps within signal-containing segments exceed the CSNR threshold and are assigned positive labels, while the majority of timesteps remain labelled as background.

Training is carried out for up to 350 epochs, with early stopping based on the validation loss to safeguard against overtraining. With cosine annealing, the learning rate is gradually reduced to a small value, which can artificially slow the loss decrease and lead to a plateau. Consequently, early stopping is applied with caution, as it may reflect a reduced learning rate rather than true model convergence.

5.4.2 Detection Statistic and Thresholding

For each input example, the trained network produces a sequence of output values $Z(t)$ spanning the 200 temporal steps of the 100-second pre-merger window. These values represent the raw LSTM outputs (logits), which are transformed into probabilities via a sigmoid activation when required for the computation of the BCE loss and for visualisation purposes.

An event-level detection statistic is constructed by applying a fixed threshold to the time-resolved model output, calibrated to achieve a predefined false-alarm probability. Candidate detection is declared at the earliest timestep for which the corresponding predicted probability exceeds the threshold. This approach enables time-resolved detection and decision-making, and supports early-warning alerts prior to a merger.

5.4.3 False Alarm Rate Definition

False alarms are estimated from the model's time-resolved outputs across all examples in the test set. At each timestep, a false alarm is registered when the predicted probability exceeds the chosen threshold,

even though the corresponding ground-truth label indicates the absence of a signal.

The FAR is then defined as the number of such false positives per unit time, accounting for the total duration of analysed data. This label-based definition is consistent with the time-resolved nature of the early-warning problem, in which detection decisions are made continuously throughout the inspiral. Model performance is evaluated as a function of FAR, with particular emphasis placed on the low false-alarm regime relevant for early-warning applications. A reference operating point corresponding to approximately one false alarm per week is used to characterise detection sensitivity under realistic alert conditions.

5.4.4 Definition of Early-Warning Time

For detected signal-plus-noise examples, the detection time, t_{detect}, is defined as the first discrete timestep at which the predicted probability exceeds a predefined threshold. The threshold is chosen to correspond to a fixed FAR, set to 1 false alarm per week in this work.

To quantify the model's early-warning capability, we consider two complementary time references. First, we define a frequency-based reference time, t_{64}, corresponding to the time at which the signal reaches a frequency of 64 Hz. The quantity $t_{64} - t_{\mathrm{detect}}$ therefore represents the remaining time until the signal reaches this reference frequency at the moment of detection, providing a practical and consistent measure of early detection relative to the inspiral evolution. In addition, we evaluate the detection in terms of the true time remaining to the merger. In this case, the early-warning time is computed as

$$t_{\mathrm{warn}} = t_c - t_{\mathrm{detect}}, \tag{5.5}$$

where t_c is the coalescence time of the signal. This quantity is obtained theoretically from the injected signal parameters during waveform generation, and provides a physically meaningful measure of how long before merger the signal is identified.

It is important to note that the use of t_c is only possible in the context of simulated data, where the true coalescence time is known a priori. In a realistic observational scenario, the merger time is not available at the moment of detection. Instead, the system would issue an alert indicating that a merger is expected to occur within a certain time window, which can be estimated from the distribution of early-warning times obtained from a representative testing set, for example, by adopting a conservative value such as the maximum or a high-percentile early-warning time achieved under realistic conditions. Since all input time series contain only pre-merger data and the coalescence time is placed beyond the analysed window, all detections occur strictly prior to merger. This ensures that the reported warning times reflect genuine early-warning capability rather than post-merger identification.

5.5 Detection Sensitivity and Early-Warning Performance

This section presents the quantitative detection sensitivity and early-warning performance of the LSTM-based framework when applied to BNS inspiral signals. Model performance is evaluated on an independent test set using the methodology described in Sec. 5.4. All results are reported at a fixed operating point corresponding to approximately one false alarm per week, reflecting realistic low-latency alert requirements for multi-messenger follow-up observations.

5.5.1 Learning Rate Strategy Comparison

To determine a stable optimisation strategy for early-warning detection in the low-false-alarm regime, three learning rate schedules were evaluated: a fixed learning rate, a stepwise reduction schedule, and cosine annealing. Fig. 5.4 shows the ROC curves for the three configurations on the BNS test set. Performance differences are most apparent at low FARs, which are of primary relevance for practical early-warning applications. At the reference operating point corresponding to approximately

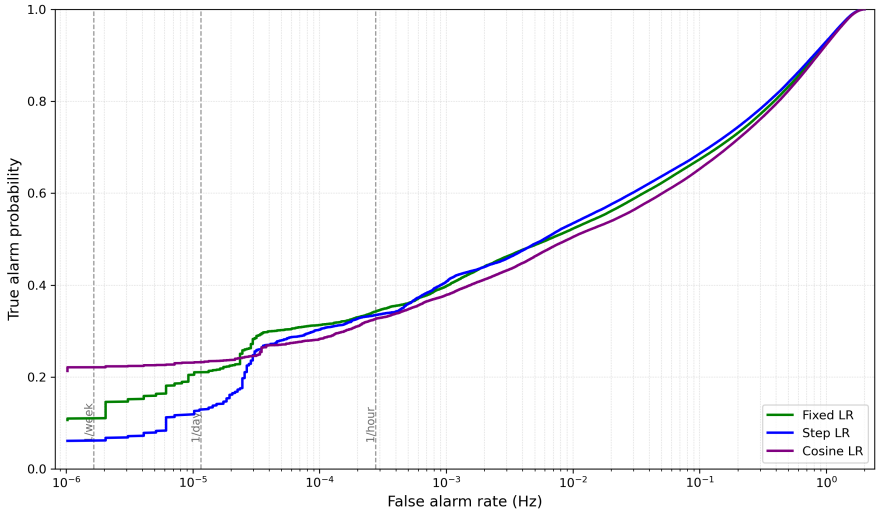


Figure 5.4: ROC curves for three learning rate strategies applied to the BNS LSTM model: fixed learning rate, step-based reduction, and cosine annealing. Vertical reference lines indicate operating points corresponding to fixed FARs of 1 per hour, 1 per day, and 1 per week. Cosine annealing achieves the strongest performance in the low-FAR regime and is therefore adopted for subsequent analysis.

one false alarm per week, the cosine annealing schedule achieves the highest true alarm probability. On this basis, cosine annealing is adopted for all subsequent analyses presented in this chapter.

5.5.2 Distribution of Model Outputs

The distribution of the LSTM model output statistic Z, corresponding to the raw logits prior to the sigmoid transformation, is presented for the test data in Fig. 5.5. These logits are used directly as input to the BCEWithLogitsLoss function, which applies the sigmoid operation internally during training. Displaying the raw outputs provides a clearer visualisation of the distribution's tails. The signal distribution exhibits a rightward shift, indicating higher values of the output statistic. In contrast, the noise distribution is shifted to the left, reflecting the model's effectiveness at distinguishing between signal and noise.

The overlap in the central region suggests the presence of signals with low SNR, where the model's distinction between signal and noise

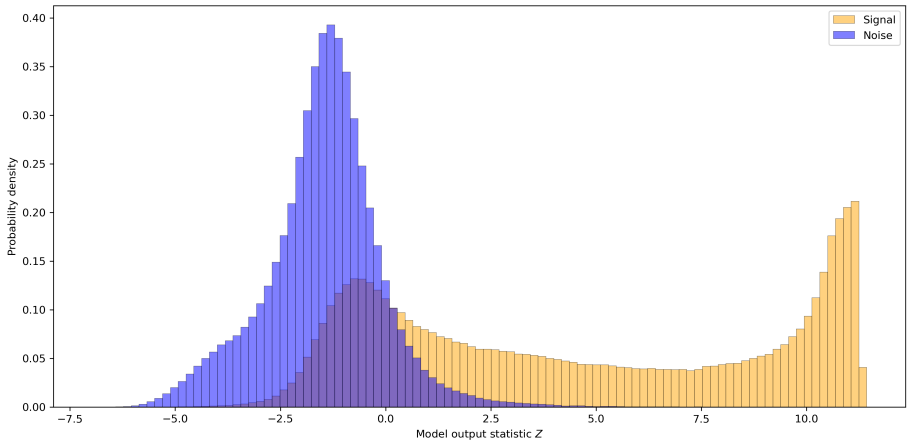


Figure 5.5: Distribution of the model output statistic Z for samples labelled as signal (signal-present, orange) and noise (noise-only, blue) in the independent BNS test set. The two distributions exhibit noticeable overlap in the central region, while the signal distribution extends towards higher positive values of Z, while the noise distribution is concentrated at negative values.

becomes less pronounced, and also indicates a subset of signals that are challenging to classify. These are instances where the signals are relatively weak and hence more susceptible to misclassification as noise. Conversely, the tail on the far right signifies signals with high SNR, which the LSTM model has identified with a high degree of confidence. These central cases may represent signals partially obscured by noise at the beginning of the signal time series, whereas events in the far-right tail yield robust detections where the signal strength is sufficiently pronounced to be clearly distinguishable from the noise. This behaviour is further analysed in Fig. 5.6.

For the BNS case considered here, the noise distribution appears smooth and does not show any additional secondary peak. This differs from the BBH case discussed in Chapter 4, as shown in Fig. 4.13, where an extra peak is present at strongly negative values of Z. The overall structure of the noise distribution is otherwise similar in both cases, but in the BBH case, this additional feature arises because the merger occurs within the analysis window. After the merger, the signal terminates

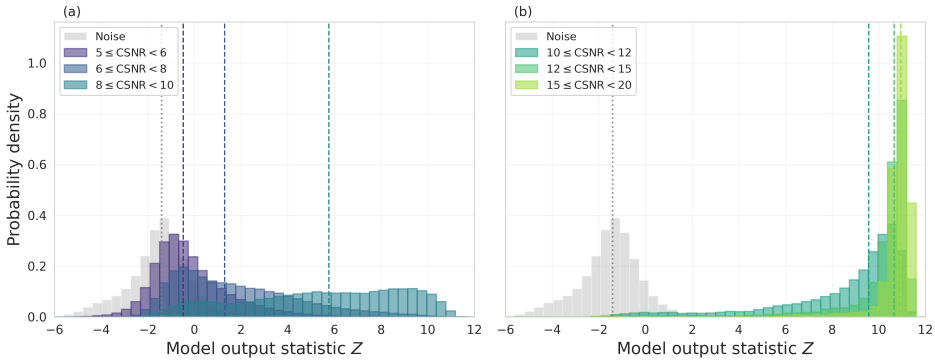


Figure 5.6: Distribution of the model output statistic Z for signal-present samples in the BNS test set, grouped by accumulated inspiral SNR. Panel (a) shows lower-CSNR signals, while panel (b) shows higher-CSNR signals. The background noise distribution is overlaid in grey for reference. As the CSNR increases, the signal distributions shift towards larger values of Z and become more concentrated, indicating that the network assigns higher confidence to stronger signals. The dashed vertical lines indicate the median of each signal distribution, while the dotted grey line marks the median of the noise distribution.

while the time series continues, and the corresponding labels revert to the noise class. As a result, late-time segments are labelled as noise despite following the presence of a strong signal, leading to an accumulation of low-Z values. In contrast, for the BNS dataset, the merger lies outside the analysed window, and no such signal-to-noise transition occurs within individual samples.

To further investigate how the model output depends on signal strength, the distribution of the output statistic Z is examined for different ranges of CSNR, as shown in Fig. 5.6. The distributions are grouped into bins of increasing CSNR, allowing a direct comparison of the network response for weak and strong signals. The background noise distribution is overlaid for reference, providing a baseline against which signal separability can be assessed. Panel (a) shows the lower-CSNR regime ($5 \leq \rho_{\mathrm{C}} < 10$), where the signal distributions are broad and exhibit significant overlap with the noise-dominated region. In this regime, the network output spans a wide range, indicating that the accumulated signal is often insufficient to yield

a confident detection. As a result, these signals are more difficult to distinguish from noise and are typically detected later in the inspiral, after additional signal power has been integrated. As the CSNR increases, a clear shift of the distributions towards higher values of Z is observed. In the intermediate regime ($10 \leq \rho_C < 12$ and $12 \leq \rho_C < 15$), the distributions become progressively narrower and move further away from the noise background, reflecting improved separability between signal and noise. This trend continues in panel (b), which shows the higher-CSNR regime ($\rho_C \geq 10$). Here, the distributions are strongly concentrated at large positive values of Z, indicating that the network assigns high confidence to these signals. For the highest CSNR values ($\rho_C \gtrsim 15$), the output values become tightly clustered near the upper range of the statistic.

This behaviour suggests a form of saturation, where the model no longer strongly differentiates between high and very high CSNR signals, instead grouping them into a common high-confidence regime. This indicates that sufficient signal evidence has been accumulated well before the merger, enabling robust, early detection. Overall, these results demonstrate that the LSTM output statistic provides a meaningful ordering of events by accumulated signal strength. While it is not directly equivalent to a matched-filter SNR, it captures the progressive build-up of signal power during the inspiral and translates this into increasing detection confidence.

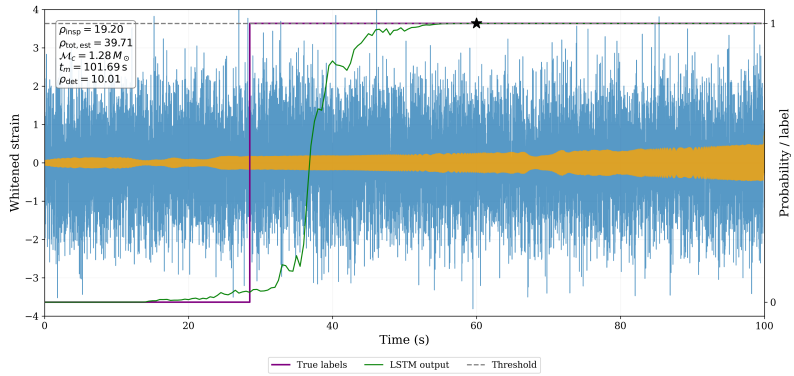
5.5.3 Representative Detection Behaviour

To illustrate the behaviour of the trained LSTM model, representative examples from the BNS test set are shown in Fig. 5.7. These examples demonstrate the model's response to both signal-plus-noise and noise-only inputs. The selected operating threshold, corresponding to one false alarm per week, lies close to unity when expressed in terms of probability and is therefore not clearly visible in the plotted output. Instead, detections are indicated by the star markers. Examining individual time series responses provides qualitative insight into how the network accumulates evidence over time and how the detection decision emerges during

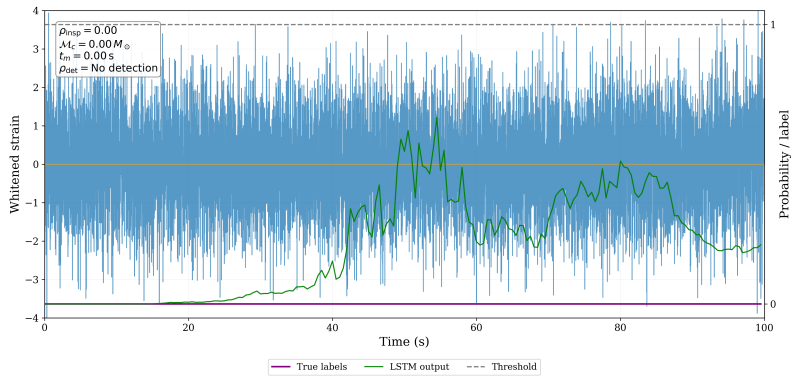
the inspiral phase. In panel (a), a high-SNR signal ($\rho_{\mathrm{insp}} = 19.20$) with chirp mass $\mathcal{M}_c = 1.28\,M_\odot$ is shown. The inspiral waveform is embedded within the noisy background and gradually increases in amplitude as the binary evolves toward merger. As the signal power grows, the LSTM output probability (green curve) increases progressively, reflecting the network's integration of signal evidence across successive time steps. The output eventually exceeds the detection threshold, producing an early-warning trigger at $t_{\mathrm{warn}} \approx 42.59\,\mathrm{s}$ before the merger. At the moment of threshold crossing, the accumulated SNR is $\rho_{\mathrm{det}} = 10.01$, while the inspiral SNR is $\rho_{\mathrm{insp}} = 19.20$, and the total estimated SNR is $\rho_{\mathrm{tot,est}} = 39.71$. This demonstrates that the signal is identified well before the full SNR has accumulated. Detection occurs when the accumulated SNR exceeds the fixed detection threshold, corresponding to one false alarm per week. This defines a characteristic SNR scale associated with the chosen FAR. It therefore demonstrates that the model can identify strong inspiral signals at an early stage of their evolution, providing advanced warning prior to merger.

Importantly, the overall probability trend increases progressively rather than abruptly, indicating that the network is responding to the cumulative temporal structure of the inspiral rather than to isolated noise fluctuations. Unlike the CSNR, which increases smoothly as signal power accumulates during the inspiral, the LSTM output is not constrained to evolve monotonically. Instead, the network output can increase or decrease between successive timesteps as the model responds to the local structure of the input data together with information retained in its recurrent memory state. This behaviour is qualitatively different from the cumulative nature of matched-filter-based statistics and suggests that the LSTM is learning a time-dependent representation of the signal rather than simply reproducing the accumulated SNR. The origin of individual fluctuations in the network output cannot be determined uniquely from the present analysis, and any detailed interpretation of these variations therefore remains speculative. In panel (b), a noise-only example is presented for comparison. In the absence of an astrophysical signal,

the LSTM output exhibits only small stochastic fluctuations associated with the background noise. However, these variations remain consistently below the detection threshold throughout the full 100 s analysis window, and no detection is produced. This stable behaviour illustrates the network's ability to suppress spurious responses and maintain a low FAR under pure-noise conditions.



(a) Detected signal-plus-noise example ($\rho_{\mathrm{insp}} = 19.20$). The LSTM output exceeds the detection threshold prior to merger; however, as the threshold lies close to unity in probability space, the crossing is not clearly visible in the plot. The detection time is indicated by the star marker.



(b) Noise-only example. The LSTM output remains below threshold, and no detection is declared.

Figure 5.7: Representative LSTM detection behaviour on the BNS test set at the selected operating threshold. The blue trace shows the whitened strain (signal plus noise), while the orange trace indicates the signal-only contribution. The green curve represents the LSTM output probability as a function of time, and the horizontal dashed line marks the fixed detection threshold. In panel (a), the probability rises above the threshold prior to merger, resulting in an early-warning detection. In panel (b), corresponding to a noise-only example, the output remains below threshold, and no trigger is produced.

Taken together, these examples highlight the time-resolved detection mechanism of the LSTM. Rather than relying on instantaneous features, the network gradually accumulates evidence during the inspiral phase

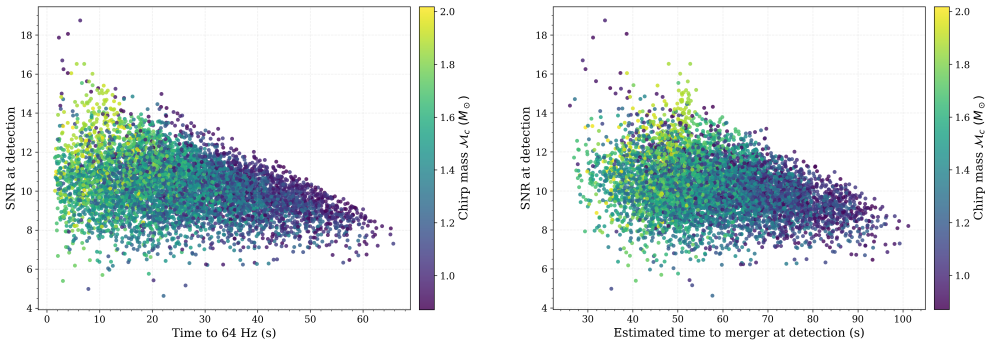
until the detection threshold is reached. This behaviour is qualitatively analogous to the gradual build-up of the matched-filter SNR in traditional GW searches, but here it is achieved directly in the time domain via the network's recurrent memory states. The examples, therefore, provide an intuitive illustration of how early-warning detection emerges from the learned temporal representation of the inspiral signal.

5.5.4 Early-Warning Performance

Fig. 5.8 shows the accumulated SNR at detection, ρ_{det}, for detected BNS events at the selected operating point corresponding to one false alarm per week, using two different time references. Panel (a) shows ρ_{det} as a function of the time remaining to 64 Hz, while panel (b) shows ρ_{det} as a function of the estimated time remaining to merger at detection. In both cases, each point represents a detected event and is coloured by chirp mass $\mathcal{M}_c$. The overall structure is similar in the two panels, despite the different definitions of the time coordinate. In panel (a), the time is measured relative to the reference frequency of 64 Hz, whereas in panel (b) it is defined using the estimated time to merger for each individual signal. The consistency between the two panels indicates that the observed behaviour is not sensitive to the specific choice of time reference. In both representations, larger values of ρ_{det} occur predominantly at shorter times before merger, whereas detections occurring earlier are associated with smaller accumulated SNR. This is expected because ρ_{det} corresponds to the accumulated SNR at the moment the detection threshold is crossed. Events detected closer to the merger have therefore had more time to accumulate signal power in the band.

A broad spread in the time coordinate is visible at lower values of ρ_{det}, indicating that the accumulated SNR at detection does not uniquely determine the achieved warning time. The absence of points in the upper-right region of the distributions reflects the fact that signals far from merger have not yet accumulated sufficient SNR to exceed the threshold, together with the finite pre-merger analysis window used in this study. The colour scale indicates a dependence on chirp mass, with lower-mass

systems tending to populate the longer-warning-time region. This is consistent with the slower inspiral evolution of lower-mass binaries, which remain in band for longer durations. Higher-mass systems evolve more rapidly and are therefore more concentrated at shorter warning times. These figures are intended primarily as qualitative summaries of the relationships between detection time, the accumulated SNR at threshold crossing, and intrinsic source properties. More quantitative statements about early-warning performance are better made at fixed total or inspiral SNR.



(a) Accumulated SNR at detection, ρ_{det}, as a function of time to 64 Hz.

(b) Accumulated SNR at detection, ρ_{det}, as a function of the time remaining to the true merger at the moment of detection.

Figure 5.8: Accumulated SNR at detection, ρ_{det}, for detected BNS events at the selected operating point corresponding to one false alarm per week. In both panels, each point represents a detected event and is coloured by chirp mass $\mathcal{M}_c$. The overall structure is consistent between the two representations, indicating that the observed behaviour is not sensitive to the specific choice of time reference. Larger values of ρ_{det} occur predominantly at shorter times before merger, reflecting the accumulation of signal power prior to threshold crossing. The absence of points in the upper-right region is due to the limited SNR available at early times in the inspiral, together with the finite pre-merger analysis window.

The dependence on chirp mass can be understood in terms of the intrinsic evolution of the inspiral signal. In Fig. 5.9, the detected population is divided into three chirp-mass ranges: low ($\mathcal{M}_c < 1.2\,M_\odot$), intermediate

($1.2 \leq \mathcal{M}_c < 1.6\,M_\odot$), and high ($\mathcal{M}_c \geq 1.6\,M_\odot$). These ranges are chosen to provide a simple qualitative separation of the detected population across the chirp-mass spectrum, rather than to define strict or physically distinct categories. Lower-mass systems sweep more slowly through the detector's sensitive frequency band, resulting in longer-duration signals that remain in band for extended periods prior to merger. This allows the network to accumulate signal evidence over a longer time interval, leading to earlier detections and consequently longer advance-warning times. In contrast, higher-mass systems evolve more rapidly and therefore spend less time in the observed frequency range. As a result, the accumulated signal power grows more quickly, but over a shorter duration, and these systems are typically detected closer to the merger. This explains the separation between chirp-mass groups observed in Fig. 5.9. These trends indicate that early-warning performance depends not only on the accumulated SNR at detection, but also on the temporal structure of the signal, which is governed by the binary's intrinsic parameters. In particular, lower chirp-mass systems are inherently more favourable for early-warning detection, as their longer inspiral phase provides a greater opportunity for the signal to be identified well before coalescence.

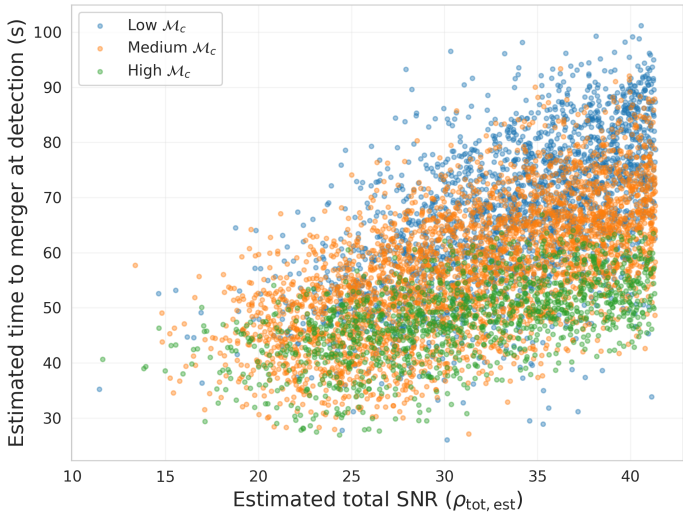


Figure 5.9: Distribution of detected BNS signals in the plane of estimated total SNR and estimated time to merger at detection, grouped by chirp mass. The population is divided into three ranges: low ($\mathcal{M}_c < 1.2\,M_\odot$), intermediate ($1.2 \leq \mathcal{M}_c < 1.6\,M_\odot$), and high ($\mathcal{M}_c \geq 1.6\,M_\odot$). Lower chirp-mass systems are associated with longer advance-warning times, while higher chirp-mass systems are detected closer to the merger.

5.6 Detection Efficiency

In order to quantify the detection power of the LSTM model, the detection efficiency is evaluated as a function of the inspiral SNR at the end of the analysed time series, $\rho_{\mathrm{insp}}(t_{\mathrm{end}})$. The efficiency is defined as the fraction of detected signals relative to the total number of injected signals within each inspiral SNR bin. All results are evaluated at a fixed operating point corresponding to a FAR of 1 per week. Fig. 5.10 shows the detection efficiency across a range of inspiral SNR values. At low SNR ($\lesssim 5$), the efficiency is effectively zero. This behaviour is expected, as signals in this regime are treated as noise within the labelling scheme, consistent with their indistinguishability from the background noise. As the SNR increases, the efficiency rises gradually, with a noticeable improvement in the range from 6 to 10, indicating the onset of the model's sensitivity.

A rapid transition in performance is observed between $\rho_{\mathrm{insp}} \approx 10$ and 13, where the model becomes increasingly reliable. The efficiency

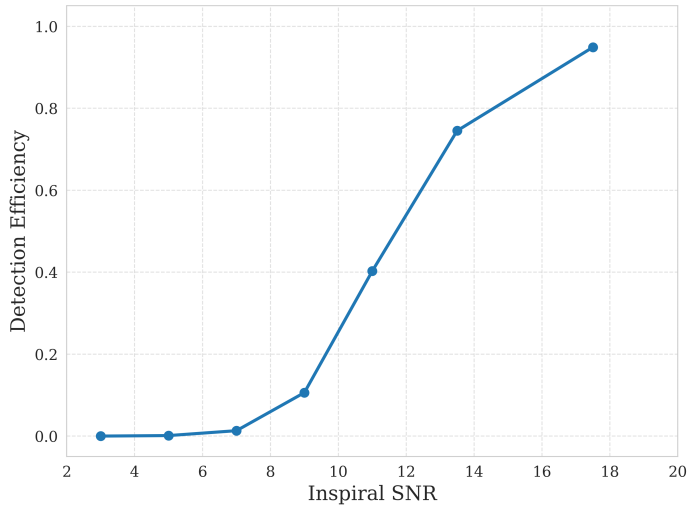


Figure 5.10: Detection efficiency of the LSTM model as a function of the inspiral SNR. The efficiency is calculated in SNR bins as the ratio of detected signals to the total number of injected signals in each bin. The results are evaluated at a fixed operating point corresponding to a FAR of 1 per week.

reaches approximately 40% in the 10–12 inspiral SNR bin and increases further to around 75% for SNR between 12 and 15. This transition region corresponds to the effective detection threshold of the model, located at $\rho_{\mathrm{insp}}(t_{\mathrm{end}}) \approx 11$–$12$. For higher SNR values ($\gtrsim 15$), the efficiency approaches unity, reaching approximately 95% in the highest inspiral SNR bin (15–20). This reflects averaging over a range of SNR values within the bin, including both moderately strong and very strong signals. This behaviour is consistent with expectations for a robust detection method, demonstrating that the LSTM is highly effective at identifying strong GW signals while remaining relatively insensitive to weak signals. However, it is notable that the transition to very high detection efficiency occurs at higher SNR values than would be expected for matched-filter-based approaches, which are known to achieve near-unity detection efficiency at lower SNR. This indicates that, while the LSTM performs well for strong signals, it remains less sensitive than optimal matched filtering in the moderate-SNR regime.

These detection-efficiency results can also be related to the ROC

analysis presented in Fig. 5.4. In that analysis, the cosine annealing strategy was selected because it provided the strongest performance in the low-FAR regime. The detection efficiency shown in Fig. 5.10 is evaluated using this selected model at the same reference operating point of one false alarm per week. While the ROC curves characterise the overall trade-off between detection performance and false alarms, Fig. 5.10 shows how the resulting detection efficiency depends on the inspiral SNR at this fixed FAR. The two figures therefore provide complementary views of the model's performance under the same false-alarm constraint.

5.7 Discussion and Conclusion

This chapter has presented a systematic investigation of early-warning detection for BNS inspirals using the LSTM-based framework introduced in Chapter 4. The model was trained and evaluated on long-duration pre-merger time series embedded in stationary Gaussian noise, with performance characterised at a fixed operating point corresponding to one false alarm per week. The results demonstrate that RNNs can provide stable, physically interpretable early-warning capability for inspiral signals. The detection efficiency results further support this behaviour, showing a clear transition from low to high detection probability with increasing inspiral SNR, and indicating an effective detection threshold around $\rho_{\mathrm{insp}} \sim 11$–$12$.

The distribution of network output scores shows a clear separation between signal-containing and noise-only examples, indicating that the network learns a meaningful detection statistic rather than responding to isolated noise fluctuations. This separation enables the definition of a detection threshold consistent with realistic low-latency alert requirements. A key objective of this work is to quantify early-warning capability. A meaningful interpretation of early-warning performance is obtained by considering the physical properties of systems that can be detected with a given advance-warning time, rather than by summarising the full detected population using a single marginal statistic. In particular, systems

yielding warning times of order tens of seconds (e.g. $\sim$ 20–40 s) are of greatest relevance for low-latency multi-messenger follow-up.

The results indicate that the achievable warning time depends on both the intrinsic binary parameters and the signal evolution within the detector band. Lower-mass systems are generally more favourable for early-warning detection, as they evolve more slowly through the sensitive frequency band, allowing the network to accumulate signal evidence over a longer duration prior to merger. In contrast, higher-mass systems evolve more rapidly and are therefore typically detected closer to merger. It is important to note that the SNR at detection, ρ_{det}, corresponds to the accumulated SNR at the moment the detection threshold is crossed, and does not directly represent the final signal strength of the event. Consequently, ρ_{det} alone does not uniquely determine the achievable early-warning time. A broad range of warning times can be obtained for similar values of ρ_{det}, particularly at lower SNR. This reflects the fact that early-warning performance depends on the detailed time evolution of the signal, rather than solely on the accumulated SNR at detection.

A more informative characterisation is therefore obtained by examining the subset of systems that can be detected with practically useful warning times. For example, systems identified $\sim$ 20–40 s before merger typically correspond to lower chirp masses and moderate-to-high SNR, reflecting the balance between signal duration and accumulated signal strength required for early detection. Such systems are expected to lie within a relatively nearby volume of the Universe, as only sufficiently strong signals can be identified well before coalescence. The maximum achievable warning time is limited by both the finite 100 s pre-merger analysis window and the gradual build-up of SNR during the inspiral. While a small subset of favourable systems can be detected up to $\sim$ 100 s before merger, such cases are rare and correspond to the most optimal signal configurations. In practice, warning times of order tens of seconds provide a more representative measure of early-warning capability.

The current LSTM pipeline is designed primarily as a detection model. Its direct output is a time-dependent detection statistic, from

which the earliest threshold-crossing time can be identified. It does not directly estimate source parameters such as the chirp mass, coalescence time, or sky position. These quantities would require an additional parameter-estimation stage or a dedicated model trained for regression or probabilistic inference. Consequently, the present analysis demonstrates early-warning detection capability rather than full source characterisation.

Time series examples further illustrate that the network output probability increases progressively during the inspiral, crossing the detection threshold only after sufficient cumulative signal evidence has been integrated. In contrast, noise-only examples remain below threshold throughout the analysis window. The optimisation strategy also plays an important role in achieving stable early-warning performance. In particular, cosine annealing learning rate scheduling was found to produce the most reliable convergence and the strongest performance in the low false-alarm regime. This behaviour is especially important for long-duration inspirals, where the network must integrate weak signal evidence over extended temporal contexts. Training for up to 350 epochs allowed the model to converge without significant overfitting, as indicated by the close agreement between training and validation loss curves.

For example, Gravitational-wave Optical Transient Observer (GOTO) has demonstrated rapid automated follow-up, with a minimum recorded response time of 28 s after receiving a GW alert. During its commissioning observations under favourable conditions, GOTO began observations within 60 s of receiving the alert for seven out of eight events [58]. These results illustrate the potential benefit of providing advance warning of a merger, although the time available for electromagnetic observations also depends on alert-generation latency, sky visibility, and the accuracy of source localisation.

Despite these encouraging results, several limitations should be noted. The analysis presented in this chapter is performed using simulated stationary Gaussian noise generated from an average H1 PSD. Real interferometer data exhibit non-stationary noise, spectral variability, and non-

Gaussian transient artefacts that may affect both false-alarm behaviour and detection sensitivity. In addition, the use of a fixed 100 s pre-merger window limits the maximum achievable warning time and restricts the earliest inspiral phase accessible to the model. The CSNR-based labelling strategy, while physically motivated, represents only one possible approach to defining detection targets and may not be optimal for maximising sensitivity to extremely weak early inspiral signals. Finally, the present study considers a single-detector configuration, whereas operational GW searches rely on coherent multi-detector analyses to improve robustness and reduce false alarms.

An important consideration for the practical application of early-warning detection is how a candidate identified during the low-frequency inspiral could be used to generate an alert for electromagnetic follow-up. The LSTM model developed in this chapter is designed for signal detection using data from a single detector and does not directly estimate the coalescence time or sky position. In a realistic observing scenario, these quantities would need to be inferred through additional analysis. In particular, the remaining time to coalescence could be estimated from the observed inspiral evolution, provided that the binary parameters, especially the chirp mass, can be constrained. Astrophysically motivated priors on neutron-star masses could help reduce the uncertainty in this estimate, although the resulting prediction would remain sensitive to parameter-estimation errors and detector noise.

Sky localisation would require information from multiple detectors, including the relative arrival times and the detectors' responses to the signal. Importantly, observing the merger itself is not essential for measuring inter-detector time delays, since the phase evolution of the inspiral can, in principle, be used to estimate the arrival time of a common waveform feature or a reference coalescence time. However, restricting the analysis to frequencies below 64 Hz reduces the available signal bandwidth and may increase timing uncertainty, particularly for weak signals. Uncertainty in the inferred chirp mass and in the time associated with a reference frequency, such as t_{64}, would propagate into the estimated coalescence time.

Uncertainties in the relative arrival times between detectors would also affect the accuracy of sky localisation. A future multi-detector framework would therefore need to combine early detection with waveform-based timing and parameter estimation, while quantifying the resulting uncertainties before issuing an astrophysically useful alert.

Overall, the results demonstrate that RNNs provide a viable and complementary approach to low-latency GW detection. The model demonstrates promising detection sensitivity under controlled Gaussian-noise conditions while simultaneously providing meaningful early-warning capability that could support rapid EM follow-up observations. These findings highlight the potential of ML-based approaches for future multi-messenger observing campaigns. While the analysis in this chapter establishes the feasibility of early-warning detection under these simulated conditions, the ultimate test of such methods lies in their performance on real interferometer data. The following chapter, therefore, applies the trained model to real H1 O3Replay strain data to examine its behaviour in the presence of non-stationary detector noise and investigate the challenges associated with generalising the model to realistic detector conditions.

CHAPTER 6

Evaluation of the Generalisation Performance of the LSTM Early-Warning Model on O3Replay Detector Data

6.1 Introduction and Motivation

In Chapter 5, an LSTM-based early-warning detection model was developed and evaluated using simulated Gaussian noise with injected BNS signals. Although controlled simulations provide a well-defined environment for model training and benchmarking, real GW detector data exhibit significantly more complex behaviour. Interferometer strain data exhibit both non-Gaussian and non-stationary behaviour. Non-Gaussianity refers to deviations of the noise-amplitude distribution from an ideal Gaussian distribution, including transient artefacts such as glitches, whereas non-stationarity refers to changes in the statistical and spectral properties of the detector noise with time. Real detector data also contain persistent spectral lines and other instrumental features that are not fully represented in idealised simulations. The objective of this chapter is to evaluate the robustness and generalisation capability of the trained LSTM model when applied to realistic detector conditions. To achieve this, the model is applied directly to O3Replay strain data, a replay of Advanced LIGO's O3 data designed to reproduce realistic detector noise. Importantly, the network is evaluated without any retraining or fine-tuning, thereby providing a strict test of its ability to generalise to realistic detector noise conditions.

This represents a challenging domain generalisation problem, since the network was trained exclusively on simulated Gaussian noise. Consequently, strong generalisation performance is not necessarily expected due to the mismatch between the statistical properties of the training and inference data. In particular, even after preprocessing, the conditioned O3Replay strain data retain residual detector artefacts and non-

stationary behaviour that cannot be completely removed through standard conditioning procedures. To enable a meaningful evaluation, the O3Replay strain data must first be conditioned to approximate the statistical and sampling properties expected by the trained model. This requires constructing a preprocessing pipeline consistent with the procedures adopted during training, including filtering, whitening, and resampling steps. The whitening procedure used in this analysis relies on estimating the PSD from the available strain data segment prior to inference. While this approach provides a practical method for offline evaluation and detector-noise conditioning, it does not fully represent a true low-latency early-warning scenario.

In a real-time implementation, early-warning detection systems must operate causally, using only information available before the analysed time sample. Consequently, the PSD cannot be estimated using the full data segment being analysed, as future data would not yet be available in an online detection setting. Instead, practical low-latency implementations would require adaptive or continuously updated PSD estimates derived from preceding detector data or from reference noise measurements obtained prior to analysis. Therefore, the preprocessing strategy adopted in this chapter should be interpreted primarily as an offline generalisation test rather than a fully real-time early-warning pipeline. In addition, the filtering, whitening, and resampling stages introduce processing latency that is not quantified in the present analysis. A practical low-latency implementation would therefore need to account explicitly for the computational and buffering delays associated with these preprocessing steps when estimating the total alert latency. Once conditioned, the data are processed in continuous inference mode, allowing the network to generate time-dependent detection probabilities across extended strain segments. This provides a more realistic assessment of model behaviour under detector conditions compared to isolated simulation-based testing. The analysis presented in this chapter therefore focuses on three primary objectives:

1. Developing a preprocessing pipeline to condition O3Replay detector data for compatibility with the trained LSTM model.

2. Applying the trained model to realistic interferometer strain data in continuous inference mode.

3. Evaluating detection performance, false-alarm behaviour, and early-warning times under realistic detector noise conditions.

This investigation highlights the challenges associated with applying machine-learning-based early-warning detection models to realistic detector data, particularly due to the mismatch between simulated training data and real interferometer noise. More broadly, it addresses the critical problem of domain shift, which remains one of the major challenges for machine-learning-based GW detection methods. The results presented in this chapter therefore provide insight into the limitations of models trained solely on idealised Gaussian simulations, as well as the challenges involved in transitioning toward realistic low-latency GW detection pipelines.

6.2 O3Replay Data Description

The analysis presented in this chapter uses the O3Replay dataset produced as part of a large-scale MDC conducted by the LVK collaboration in preparation for the O4. The purpose of this challenge was to evaluate the performance and readiness of GW detection pipelines under realistic observing conditions. In this framework, calibrated detector strain data from the O3 are replayed in time order while simulated CBC signals are injected into the data stream. The injected signals are distributed throughout the dataset and span a range of astrophysical source classes; however, the injection rate is intentionally significantly higher than the true astrophysical event rate to ensure adequate statistical coverage for robust model evaluation. The low-latency replay framework described in [139] enables detection algorithms to be tested under conditions that closely resemble real detector operation, while maintaining precise knowledge of the injected signal parameters.

The O3Replay dataset therefore provides a realistic approximation to detector-noise conditions in which GW detection algorithms can be evaluated, while preserving the non-Gaussian and non-stationary characteristics of real interferometer noise. Using this framework, [139] evaluated the performance of the LVK low-latency alert infrastructure using simulated CBC signals injected into replayed O3 detector data. In the MDC, the injected population consisted of approximately 40.9% BNS systems distributed uniformly in comoving volume up to a redshift of $z = 0.15$. Out of the 5×10^4 simulated compact binary injections, 1489 BNS events were successfully recovered by the online search pipelines. The study showed that low-latency searches operating on replay data were capable of recovering injected signals and issuing preliminary alerts with a median latency of approximately 29.5 s after merger.

In addition, the MDC included dedicated early-warning searches targeting neutron-star mergers during the inspiral phase, which successfully identified a subset of BNS injections prior to coalescence. The reported median latency for early-warning detections was approximately -3.1 s, corresponding to detections occurring a few seconds before merger. The study also assessed rapid sky-localisation performance for significant multi-detector BNS candidates, finding median searched areas of approximately 100–200 deg^2. These results demonstrate the feasibility of near real-time detection and localization of BNS mergers under realistic detector-noise conditions, supporting rapid multimessenger follow-up observations.

6.2.1 Detector Strain Data

The detector data analysed in this work are obtained from the LIGO Hanford H1 interferometer as part of the O3Replay dataset. A single-detector configuration is adopted to ensure consistency with the training setup of the LSTM model, which was developed using single-detector strain time-series data. This choice provides a controlled framework for evaluating the model's ability to identify GW signals in realistic, non-Gaussian and non-stationary detector noise. Focusing on a single detec-

tor allows the analysis to isolate the problem of signal detection in time-series data without introducing additional complexities such as time-delay alignment and coherent combination of multiple data streams. While more complex configurations may enhance detection confidence and enable additional inference tasks, the single-detector setup is sufficient for assessing the core detection capability of the proposed method. Extension to more complex configurations is left for future work. The strain data are stored in frame files of duration 4096 s and contain calibrated interferometer strain measurements. Each frame includes the strain channel `H1:GDS_CALIB_STRAIN_INJ1`, which contains the detector output with simulated GW injections added. In addition, the MDC provides a corresponding noise-only strain channel containing detector data without injected signals. This channel was used in this work for estimating the background noise distribution and evaluating the false alarm behaviour of the detection model.

In realistic observational scenarios, a separate noise-only data stream is not available, and noise characterisation must instead be performed directly on the detector data containing astrophysical signals. However, the MDC injection campaign contains an artificially high rate of simulated signals, making the injected strain unsuitable for reliable background estimation. The availability of a dedicated noise-only channel in the MDC therefore provides a practical way to estimate detector noise properties without contamination from the large number of simulated injections. The strain data reflect realistic interferometer noise conditions derived from the O3. As such, the data include a variety of instrumental features, such as spectral lines and transient noise artefacts (glitches), which are characteristic of real detector noise and differ significantly from the stationary Gaussian noise used during the training of the LSTM detection model described in Chapter 5. These transient features can mimic short-duration signals, posing a challenge for GW detection algorithms. Consequently, applying the trained model to O3Replay data provides a stringent test of its ability to generalise to realistic detector environments. An example segment of the O3Replay strain data is shown in Fig. 6.1.

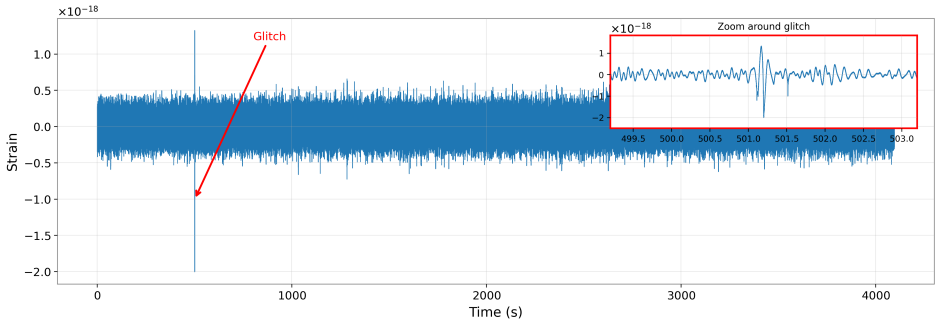


Figure 6.1: Example segment of raw O3Replay strain data from the H1 detector, shown without any additional preprocessing or filtering applied. Although the strain appears approximately stationary over the duration shown, transient noise artefacts (glitches) are present as short-duration high-amplitude excursions. The highlighted glitch is shown in greater detail in the inset panel, illustrating its transient oscillatory structure.

Over the duration displayed, the time series appears approximately stationary; however, it contains transient noise artefacts (glitches), visible as short-duration high-amplitude excursions. The data are obtained from the standard detector strain channel and are presented here for illustrative purposes.

The frequency-domain representation of the data is shown in Fig. 6.2. This figure is intended as a diagnostic representation of the raw O3Replay data prior to filtering, whitening, and resampling, rather than as the PSD used by the final 128 Hz analysis pipeline. The PSD of the raw O3Replay strain data exhibits narrow spectral features (spectral lines) as well as strong frequency-dependent structure, indicating that the detector noise is highly non-white prior to preprocessing. These features arise from instrumental and environmental sources and are not present in ideal white noise models commonly used in simulated analyses. A zoomed-in view of a representative transient noise artefact is shown in Fig. 6.3. This illustrates the short-duration, high-amplitude structure of glitches present in the detector data.

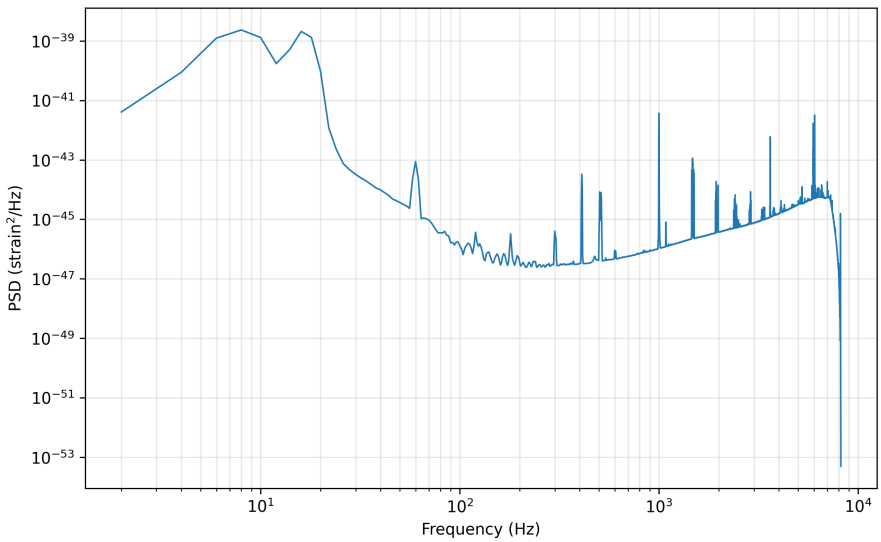


Figure 6.2: PSD of the O3Replay strain data from the H1 detector. The spectrum shows narrow spectral features characteristic of instrumental noise.

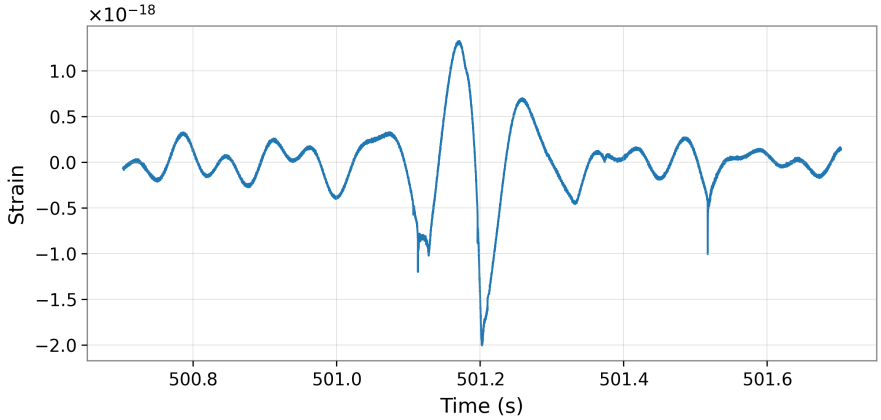


Figure 6.3: Zoomed-in view of a transient noise artefact (glitch) in the O3Replay strain data, showing a short-duration high-amplitude excursion relative to the surrounding background noise.

The detector strain data are stored in frame files, which are the standard LIGO data format used to store calibrated interferometer time-series data over fixed-duration segments. In this work, each frame file contains 4096 s of strain data. The frame files are processed sequentially

and concatenated to form a continuous strain time series that is subsequently conditioned using the preprocessing pipeline described later in this chapter. The injected signals included in the O3Replay dataset span several classes of CBC sources, including BNS, BBH, and NSBH systems. The selection of the specific injections analysed in this study is described in the following subsection.

6.2.2 Injection Catalogue

The parameters of the simulated GW signals injected into the O3Replay strain data are provided in an accompanying injection catalogue. This catalogue contains a structured dataset named `events`, which lists the physical parameters and injection times for a large set of simulated CBC signals embedded within the detector data, enabling direct comparison between injected signals and model predictions. Each entry in the catalogue includes a comprehensive set of parameters describing the injected system, including the component masses, the matched-filter SNR, and the geocentric coalescence time, along with additional source properties and the waveform model used to generate the injected signal.

These metadata allow the precise identification of injection times within the detector strain data and enable direct comparison between the model predictions and the true signal occurrence. A signal-only time series can be obtained by subtracting the noise-only strain channel from the corresponding signal-plus-noise channel, thereby providing an approximate representation of the injected waveform. Successive injections occur at intervals of approximately one minute ($\sim$60 s), as defined in the O3Replay MDC. To illustrate the characteristics of the injected signals, representative examples of the signal-only time series are shown in Fig. 6.4. The signals exhibit characteristic chirp behaviour, with increasing amplitude and frequency as the binary system approaches merger. The differences in duration and frequency evolution between BNS and BBH systems reflect their distinct physical properties and mass ranges.

In this study, the analysis focuses specifically on BNS systems. Since the injection catalogue does not explicitly label the source type of each

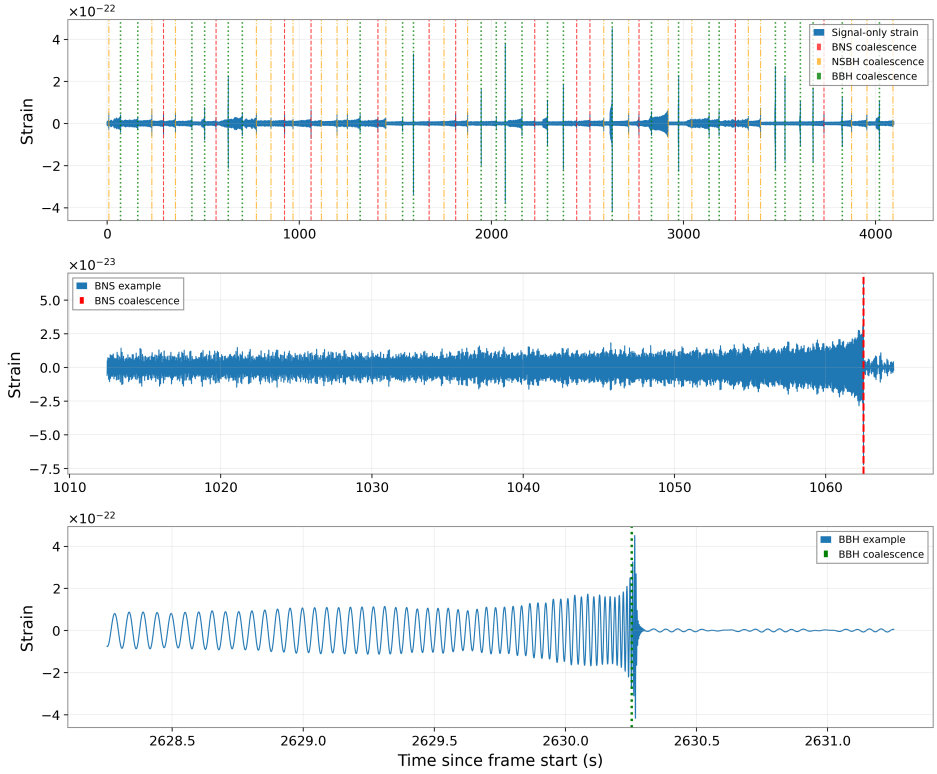


Figure 6.4: Examples of unprocessed signal-only strain data containing injected CBC signals. Top: Strain time series from a single 4096 s frame containing $N = 60$ injected CBC events. The red dashed vertical lines indicate BNS coalescence times, while the green dotted vertical lines indicate BBH coalescence times. Middle: Example of a signal-only BNS waveform over the final ~ 50 s prior to coalescence, with component masses $m_1 = 1.60\,M_\odot$ and $m_2 = 1.26\,M_\odot$. Bottom: Example of a signal-only BBH waveform over the final few seconds prior to coalescence, with component masses $m_1 = 23.55\,M_\odot$ and $m_2 = 21.50\,M_\odot$, showing a shorter-duration, lower-frequency waveform followed by a ringdown phase. Small residual oscillations visible after the merger are due to data-processing artefacts and do not represent physical post-ringdown strain.

event, BNS signals are identified by selecting events for which both component masses lie within the range $1\,M_\odot \leq m_1, m_2 \leq 2.5\,M_\odot$. This mass range is consistent with the prior used during the training of the early-warning detection model (see Tab. 5.1), ensuring that the evaluation is performed on signals drawn from the same region of parameter space. The injection parameters also span a broader set of physical properties, including variations in sky locations, distances, and binary orientations, generated according to the predefined distributions used in the O3Replay dataset. The training priors were selected to be broadly consistent with these MDC injection distributions, allowing for a meaningful assessment of model performance on realistic replayed detector data. In the O3Replay injection set used in this work, BNS waveform durations are typically of order 10^2–10^3 s. During the inspiral phase, the signal amplitude and frequency increase gradually as the binary system loses orbital energy through GW emission. This extended inspiral evolution provides an opportunity for detection algorithms to identify the signal before the final coalescence, potentially enabling advance alerts for EM follow-up observations.

The injection catalogue is used throughout the analysis to identify the timing of injected signals, evaluate detection performance, and measure the early-warning time achieved by the LSTM detection model relative to the true coalescence time. For evaluation purposes, the data can be interpreted as comprising two complementary testing sets: a foreground dataset, consisting of strain data containing injected signals, and a background dataset, consisting of noise-only strain. The background dataset is essential for estimating the FAR, particularly given the high density of injected signals in the foreground data, where signals are present almost continuously in time. The LSTM model is applied to the full dataset, irrespective of the type of injected signal present, in order to assess its ability to generalise beyond the BNS parameter space on which it was trained, including its response to BBH and NSBH signals.

6.3 Experimental Design and Evaluation Plan

The objective of this analysis is to evaluate the performance of the trained LSTM-based early-warning detection model when applied to realistic interferometer data from the O3Replay dataset. The model was trained exclusively on simulated Gaussian noise with injected BNS signals, as described in Chapter 5. In this chapter, the trained model is applied directly to O3Replay detector data without retraining or fine-tuning. This approach allows the robustness and generalisation capability of the model to be assessed under realistic detector noise conditions. The evaluation is performed by processing the conditioned strain data in a continuous inference framework. The LSTM model generates a time-dependent detection probability for each input time step, indicating the likelihood that a GW signal is present in the data. As the analysis proceeds through the strain time series, these outputs form a probability stream that reflects the evolving response of the model to the detector data.

These probabilities are subsequently analysed to examine whether the model output exceeds the detection threshold and to evaluate the timing of the network response relative to the true coalescence time of injected signals. The detection criterion is defined as the first time at which the output probability exceeds the threshold corresponding to a fixed FAR.

These times are used to align the model output with the injected signals and to examine the temporal behaviour of the network response relative to the true coalescence time. The overall evaluation therefore focuses on the behaviour of the model in realistic detector noise, the false-alarm characteristics of the detection statistic, and the response of the network around injected BNS signals. Since the model outputs do not exceed the adopted FAR-based detection threshold for the analysed BNS injections, the results are interpreted as a study of model behaviour and generalisation rather than as a quantitative measurement of detection efficiency or early-warning performance.

6.4 Preprocessing Pipeline for O3Replay Strain

To ensure that the detector data analysed in this chapter are compatible with the trained model, a preprocessing pipeline is constructed to reproduce the same conditioning steps applied during training. This pipeline transforms the raw interferometer strain data into a representation that matches the statistical structure expected by the LSTM model while preserving the underlying GW signal content. This consistency is essential, as neural networks are highly sensitive to changes in input data distribution, and even small deviations can significantly degrade model performance. The preprocessing procedure consists of several stages, including high-pass filtering, whitening, low-pass filtering, and downsampling to 128 Hz. The resulting time series are subsequently normalised prior to being input to the LSTM model, ensuring consistency with the data distribution used during training.

6.4.1 High-Pass Filtering

The calibrated interferometer strain data contain significant low-frequency noise contributions arising from seismic motion, suspension resonances, and slow instrumental drift. These components dominate the strain at very low frequencies and are not informative for CBC detection in the frequency range considered in this work. To suppress this contamination, a fourth-order Butterworth high-pass filter with a cutoff frequency of 18 Hz is applied to the strain data. The filter is implemented using second-order sections and applied in a zero-phase forward–backward configuration via `sosfiltfilt`, ensuring that phase distortions are avoided and the temporal structure of the signal is preserved.

The preprocessing pipeline is applied separately to both the noise-only strain data and the signal-plus-noise strain data. The filtered noise-only channel is subsequently used for PSD estimation, while the corresponding signal-plus-noise channel is whitened using the PSD estimated from the associated noise segment. This consistent treatment ensures compatibility between the PSD estimation procedure and the whitening

stage applied to the detector strain data. Filtering is performed independently on each 4096 s data segment prior to subsequent preprocessing stages, including automatic notch filtering, whitening, low-pass filtering, and resampling. This approach provides uniform conditioning of all data segments while avoiding inconsistencies across frame boundaries. The choice of an 18 Hz cutoff is motivated by the need to suppress low-frequency contamination dominated by seismic and instrumental noise, while retaining the inspiral signal content relevant for early-warning detection.

6.4.2 Whitening of the Strain Data

Following high-pass filtering, the calibrated interferometer strain data are whitened in order to reduce the frequency-dependent structure of the detector noise. Whitening transforms the strain time series such that the noise PSD is approximately flattened across the analysed frequency band. The data are processed in multiple 4096 s frame files. For each frame, the whitening PSD is estimated directly from the corresponding filtered noise-only channel using Welch's method. The same PSD is then applied to whiten the associated signal channel. This approach preserves consistency between the detector noise properties and the whitening transformation applied to the strain data while allowing the whitening filter to adapt to variations in the detector noise between different frame files. The whitening procedure is implemented in the frequency domain. The Fourier transform of the strain time series is divided by the square root of the reference PSD, and the resulting spectrum is transformed back to the time domain. In this work, whitening is restricted to the 20–58 Hz frequency band through the application of a smooth frequency-domain taper. Frequencies below 20 Hz and above 58 Hz are progressively suppressed during the whitening operation in order to avoid sharp spectral discontinuities.

The LSTM model was trained on strain data sampled at 128 Hz, corresponding to a Nyquist frequency of 64 Hz. Restricting the analysed

frequency range to 20–58 Hz ensures consistency with the training configuration while maintaining a conservative margin below the Nyquist limit to prevent aliasing effects during downsampling. This produces a time series in which the noise variance is approximately uniform within the analysed band. In GW data analysis, it is standard practice to suppress high-frequency content prior to decimation in order to avoid spectral distortion and aliasing artefacts [122, 163]. Applying whitening using a PSD estimated from the corresponding detector noise ensures that both the signal and noise components undergo a physically consistent transformation. Since whitening modifies not only the detector noise but also the morphology and amplitude evolution of the GW signal, consistency between the PSD estimation and the processed strain data is important for maintaining a stable signal representation for the ML model.

It should be noted, however, that real detector noise is non-stationary and its PSD evolves over time. Consequently, whitening does not produce perfectly white noise, but instead provides an approximate flattening of the detector spectrum within the analysed frequency band. Narrow spectral lines may persist due to instrumental artefacts and non-stationary noise behaviour. To mitigate these effects, automatic notch filtering is applied prior to whitening. Spectral lines are identified from the noise-only PSD and suppressed using narrow-band notch filters, reducing the contribution of persistent line artefacts that are not fully removed by the whitening procedure alone.

6.4.3 Low-Pass Filtering

Following the whitening procedure, a fourth-order Butterworth low-pass filter with a cutoff frequency of 58 Hz is applied to suppress high-frequency components above the analysed frequency range. The filter is implemented using second-order sections and applied in a zero-phase forward–backward configuration via `sosfiltfilt`, thereby avoiding phase distortion of the signal.

The filtering step is applied prior to resampling in order to reduce potential aliasing effects during downsampling. Restricting the analysed

bandwidth also reduces the data volume and computational cost associated with processing long-duration detector time series. Since the analysis is primarily focused on the early inspiral regime of BNS systems, the retained low-frequency signal content remains sufficient for advance detection prior to merger.

6.4.4 Resampling and Downsampling

The calibrated interferometer strain data are originally recorded at a sampling rate of approximately 16384 Hz, which is substantially higher than required for the LSTM detection model used in this work. In order to match the sampling rate adopted during model training, the conditioned strain data are resampled to a final sampling rate of 128 Hz. The resampling process is carried out using polyphase filtering through the `resample_poly` implementation in SciPy [164]. This method integrates anti-aliasing filtering with resampling in a computationally efficient manner, allowing the sampling rate to be reduced while preserving the low-frequency inspiral characteristics of the GW signal.

Reducing the sampling rate significantly decreases the computational cost associated with analysing long stretches of detector data. In this work, the retained temporal structure corresponds primarily to the early inspiral phase of BNS signals, characterised by a slowly evolving frequency chirp and a gradual accumulation of SNR. This regime is particularly relevant for early-warning detection, as it enables identification of the signal well in advance of merger, rather than within the final $\sim$ 10 seconds where higher-frequency contributions dominate the waveform evolution. Following resampling, the strain amplitude is scaled by the square root of the downsampling factor in order to preserve the approximate variance of the whitened time series after resampling. Finally, edge regions affected by filtering and resampling transients are removed before the data are used for subsequent analysis. The resulting 128 Hz strain time series is consistent with the data representation used during model training, as described in Chapter 5. Maintaining this sampling rate ensures that the input data during evaluation match the conditions

under which the LSTM model was trained, thereby avoiding discrepancies in the temporal features learned by the model.

6.4.5 Construction of the Model Input Time Series

Following the preprocessing steps described above, the strain data are segmented into 4096 s chunks to construct the input time series for the LSTM model. Each segment is processed independently and transformed into a representation consistent with the data used during model training. The preprocessing pipeline includes high-pass filtering, whitening using a PSD estimated from the corresponding noise-only channel, low-pass filtering, resampling to 128 Hz, and edge cropping. These steps condition the strain data while preserving the low-frequency inspiral structure most relevant for early-warning detection. Each segment is processed independently rather than assuming continuity across the full dataset, which may contain gaps, non-stationary noise behaviour, and variations in detector data quality. This segmented approach allows the preprocessing to remain stable across long-duration stretches of detector data while maintaining consistency between different frame files.

While it does not fully capture the time-varying nature of the detector noise, the preprocessing steps described above help mitigate some of the resulting variability in the strain data. The resulting preprocessed time series preserve the temporal evolution of the detector strain within the selected frequency band while ensuring consistency with the input representation used during model training. These conditioned strain data are subsequently used in the continuous inference framework described in the following sections to evaluate the performance of the early-warning detection model on realistic detector noise.

6.5 Verification of Preprocessing

6.5.1 Kolmogorov–Smirnov Test

In addition to visual inspection of the processed strain time series, a statistical consistency check was performed to assess whether the conditioned O3Replay noise data exhibit distributional properties broadly

consistent with the Gaussian noise used during model training. This was evaluated using the Kolmogorov-Smirnov (KS) test applied to preprocessed noise-only strain files. The KS statistic D measures the maximum difference between the empirical cumulative distribution function of the processed strain data and that of a standard normal distribution [165, 166]. It should be noted that the KS test probes only the distributional properties of the data and does not capture temporal correlations or non-stationary behaviour present in realistic detector noise. Consequently, consistency with the KS criterion does not imply that the detector noise is fully Gaussian or statistically equivalent to the simulated training data used during model development.

For the KS analysis, each preprocessed noise-only strain file, with a duration of 4096 s, is first normalised using statistics computed over the full file duration. The resulting time series is then divided into overlapping windows of duration 16 s with a stride of 8 s, corresponding to a 50% overlap between adjacent windows. A one-sample KS test is subsequently performed for each window by comparison with a standard normal distribution. In this work, the Gaussian consistency criterion is based on the KS statistic D rather than the associated p-value, since the p-value depends strongly on the sample size and window duration. Windows satisfying $D \leq 0.075$ are considered approximately consistent with the Gaussian assumptions adopted during model training. To evaluate the behaviour of an entire 4096 s file, the fraction of windows satisfying this criterion is computed. A file is classified as Gaussian-like if at least 80% of its windows pass the KS threshold.

In addition, the presence of large-amplitude outliers is monitored by counting samples whose absolute normalised amplitude,

$$z = \frac{x - \mathrm{median}(x)}{\sigma}, \tag{6.1}$$

exceeds 5, where $\mathrm{median}(x)$ and σ denote the median and standard deviation of the strain samples within each 4096 s file. This provides a simple diagnostic for identifying residual glitch-like transients in the processed strain data. Applying this procedure to the dataset shows that

processed noise-only files ($\sim 75\%$) are classified as Gaussian-like according to the acceptance criterion described above. This indicates that a substantial fraction of the conditioned dataset exhibits amplitude distributions broadly consistent with the statistical assumptions used during model training.

The distribution of the normalised strain for multiple noise-only files classified as passing or failing the KS consistency criterion is shown in Fig. 6.5. Each curve corresponds to an individual preprocessed 4096 s file. The KS-passed files show distributions broadly consistent with a standard normal distribution, whereas the KS-failed files exhibit visible deviations from Gaussian behaviour, particularly in the shape of the central peak and the distribution tails. These results indicate that, although the preprocessing pipeline substantially reduces non-Gaussian structure in many files, residual non-Gaussian features remain in a subset of the dataset.

Excluding files that fail the KS consistency criterion also removes the corresponding time intervals from the subsequent analysis. Consequently, any gravitational-wave signals occurring entirely within these excluded intervals would not be evaluated by the model, while signals extending across the boundaries of excluded files may be only partially represented. This represents a limitation of the adopted data-selection procedure and reduces the temporal coverage of the analysis.

The trained LSTM model was developed using approximately stationary Gaussian noise, and its performance may therefore be affected by deviations from these statistical properties. Although the majority of the processed files satisfy the Gaussian consistency criterion, passing this test does not guarantee the complete absence of non-Gaussian features. Such residual deviations may influence the background behaviour of the network and contribute to increased false-alarm activity within the selected and preprocessed detector data.

At the same time, the preprocessing pipeline reduces transient detector artefacts but does not eliminate all non-Gaussian features. However, the exclusion of files that failed the Gaussian consistency criterion limits

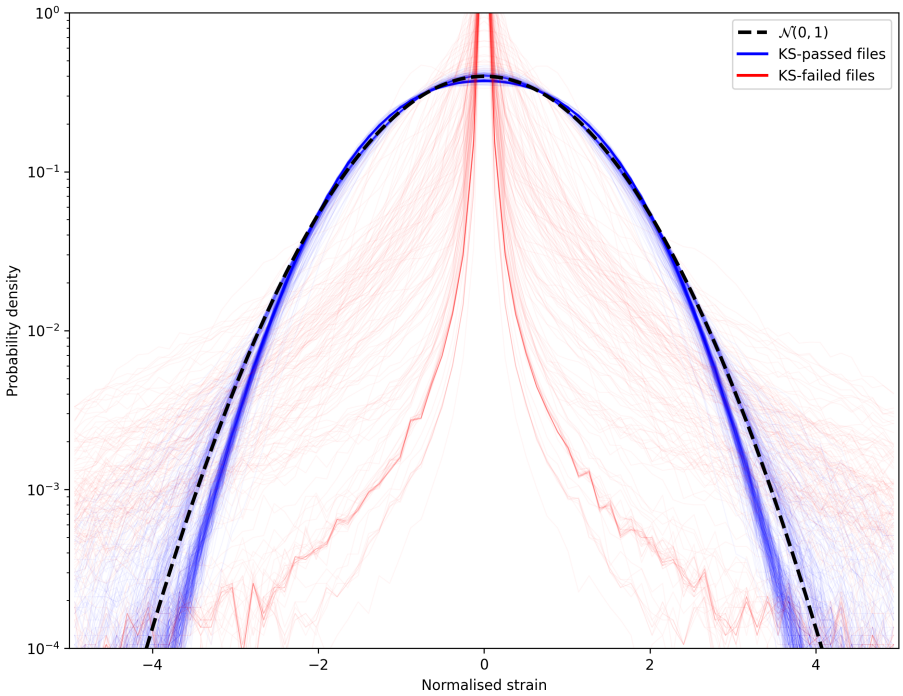


Figure 6.5: Distribution of the normalised strain for noise-only data classified as passing or failing the KS consistency test. Each blue or red curve corresponds to the strain distribution of an individual preprocessed 4096 s file in the KS-passed or KS-failed set, respectively. The dashed black curve represents the standard normal distribution, $\mathcal{N}(0,1)$. The KS-passed files closely follow the Gaussian distribution, whereas the KS-failed files exhibit visible deviations in both the central peak and distribution tails.

the range of detector noise conditions represented in the analysis. Consequently, the evaluation is based on a selected subset of preprocessed O3Replay strain data that exhibits approximately Gaussian-like statistical properties, while retaining some residual non-Gaussian features. Following the construction and verification of the preprocessing pipeline, the trained LSTM model is applied to the conditioned O3Replay strain data using long-duration sequential inference performed independently on each preprocessed 4096 s file. Within each file, the internal LSTM state is propagated sequentially between consecutive time steps over durations significantly longer than the 100 s sequences used during training.

However, the inference is not continuous across file boundaries, since the internal network state is reset at the start of each new file. The resulting probability time series are subsequently analysed to evaluate detection performance, false-alarm behaviour, and early-warning capability under preprocessed real detector noise conditions.

6.6 Model Performance on Preprocessed O3Replay Data

6.6.1 Statistical Diagnostics of the Preprocessed Detector Noise

Although the preprocessing pipeline and validation tests presented earlier indicated approximate agreement with Gaussian amplitude statistics, additional diagnostics show that residual temporal and spectral structure remains present after preprocessing and whitening. The diagnostics discussed below were computed using a representative continuous 4096 s segment from a single preprocessed O3Replay detector-noise file sampled at 128 Hz. To further examine the statistical properties of the preprocessed detector data, several time-domain and spectral diagnostics were compared with a reference Gaussian noise distribution. The first set of diagnostics was computed after normalising the data using statistics estimated over the full 4096 s segment.

Fig. 6.6 compares the time evolution of the band-limited power for the reference Gaussian noise and the preprocessed O3Replay detector data across multiple representative frequency bands. The reference Gaussian noise exhibits relatively stable behaviour across all frequency bands, consistent with approximately stationary white noise. In contrast, the O3Replay detector data shows substantial temporal variability and intermittent excursions throughout the analysed duration. The similar temporal behaviour observed across the three frequency bands suggests that these fluctuations are primarily associated with broadband variations in the overall detector noise level, rather than with isolated features confined to a specific frequency band.

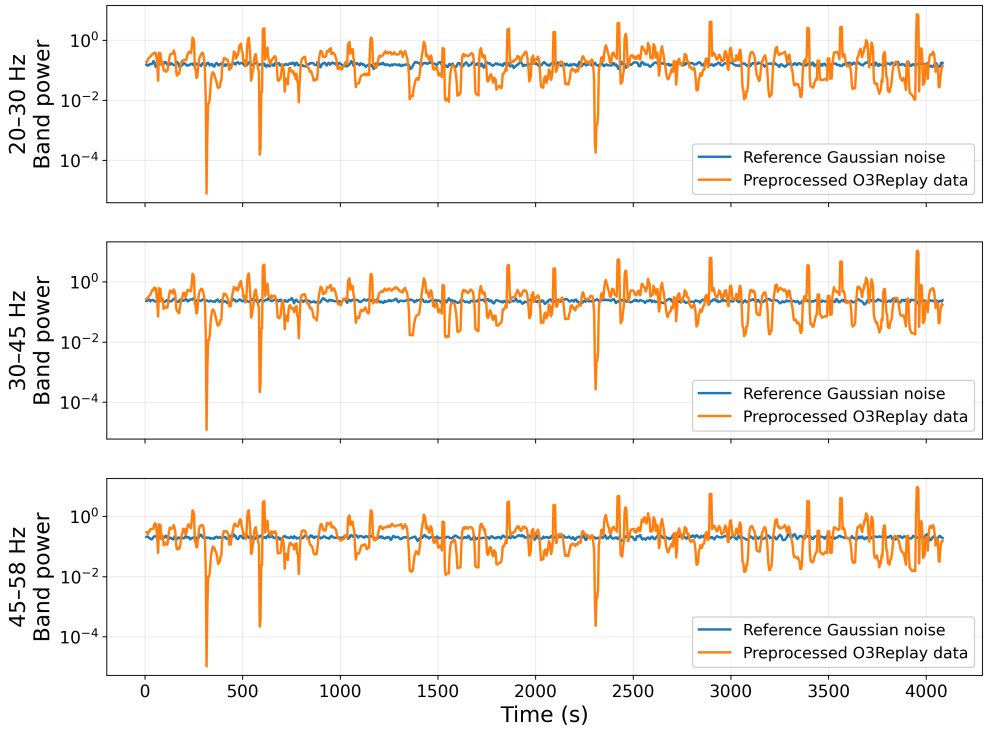


Figure 6.6: Time evolution of the band-limited power for the reference Gaussian noise and a representative 4096 s segment of the preprocessed O3Replay detector data in three representative frequency bands: 20–30 Hz, 30–45 Hz, and 45–58 Hz. The band-limited power was estimated using 16 s windows advanced in steps of 4 s, corresponding to 75% overlap between adjacent segments.

To further examine the temporal statistical behaviour of the data under the same long-timescale normalisation, the rolling variance was evaluated as a function of time. Fig. 6.7 shows that the reference Gaussian noise maintains a relatively stable variance throughout the analysed duration, whereas the O3Replay detector data exhibits clear temporal variability and intermittent excursions. Together, the band-power and rolling-variance diagnostics indicate that normalisation using statistics computed over the full 4096 s segment is insufficient to fully track the non-stationary behaviour of the detector noise. These results therefore motivate the use of a shorter-timescale normalisation strategy.

To reduce the impact of slow temporal variations in the detector

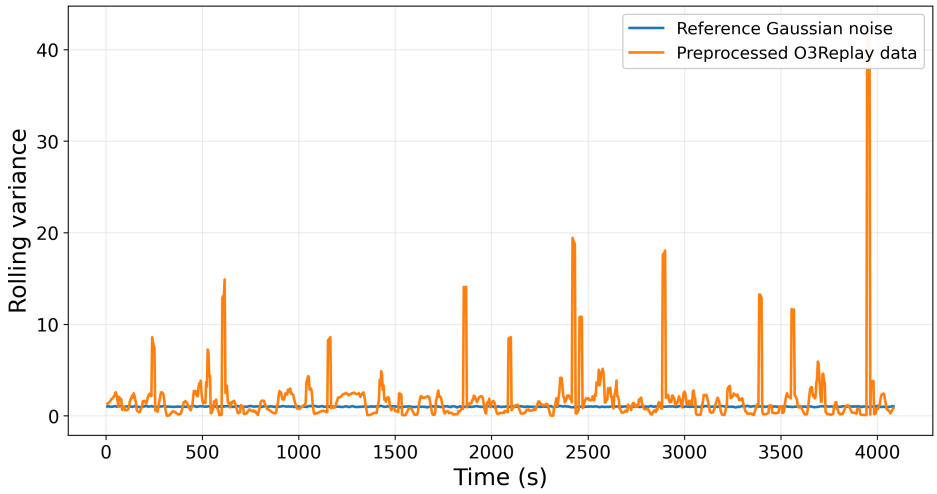


Figure 6.7: Rolling variance as a function of time for the reference Gaussian noise and the preprocessed O3Replay detector data after normalisation using statistics computed over the full 4096 s segment.

noise level and improve the statistical stability of the strain data, the normalisation procedure was modified prior to LSTM inference. Instead of using statistics computed over the full 4096 s segment, the strain data were normalised independently within fixed 100 s windows. This normalisation strategy was consistent with the preprocessing applied during model training and was intended to stabilise the mean and variance over time while mitigating broadband fluctuations in the detector noise level. Fig. 6.8 shows the rolling-variance diagnostic after applying the 100 s local normalisation. The large-scale variance fluctuations in the O3Replay detector data are substantially reduced relative to the corresponding results obtained using normalisation over the full 4096 s segment. This demonstrates that the 100 s local normalisation improves the statistical stability of the detector-noise data.

However, intermittent excursions and residual temporal variability remain present after local normalisation. This indicates that, although the shorter-timescale normalisation reduces long-timescale non-stationary behaviour, the preprocessed detector noise does not fully exhibit the approximately stationary statistical behaviour expected for ideal white

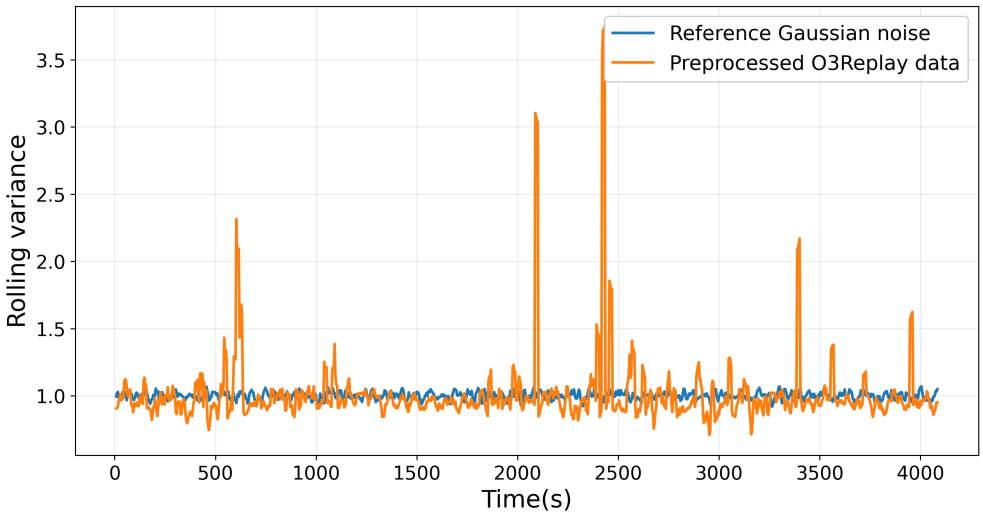


Figure 6.8: Rolling variance as a function of time for the reference Gaussian noise and the preprocessed O3Replay detector data after applying the 100 s local normalisation prior to LSTM inference.

Gaussian noise. Since the LSTM model processes the strain data sequentially, such residual temporal correlations and statistical variations may influence the hidden-state evolution and consequently affect the network response. These effects are consistent with the broader and more asymmetric distribution of the raw output statistic $z(t)$ observed for the O3Replay detector noise relative to the reference Gaussian distribution.

Overall, these results suggest that approximate agreement with Gaussian amplitude statistics alone is insufficient to guarantee that the preprocessed detector data fully satisfies the stationary white-noise assumptions underlying the idealised training conditions. Nevertheless, the preprocessing, whitening, and 100 s local normalisation substantially reduce broadband variance fluctuations and long-timescale non-stationary behaviour in the detector noise. Remaining local variations may still contribute to differences in behaviour when the model is applied to real detector noise, and further investigation would be required to determine whether these effects could be reduced through alternative preprocessing approaches or more adaptive local normalisation strategies.

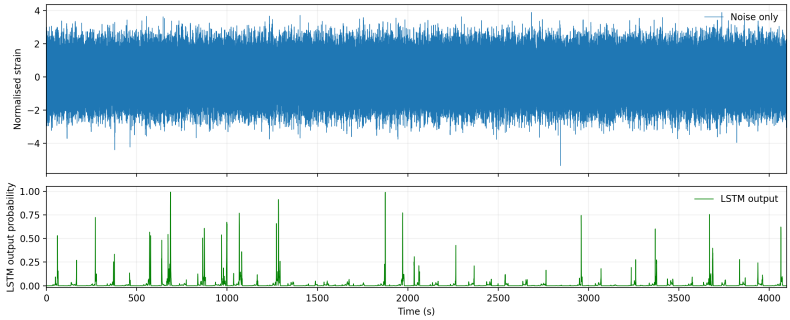
6.6.2 Model Behaviour on Noise-Only Data

The network produces a raw output $z(t) \in \mathbb{R}$ at each time step. For interpretability, this output can be mapped to a probability via the sigmoid function

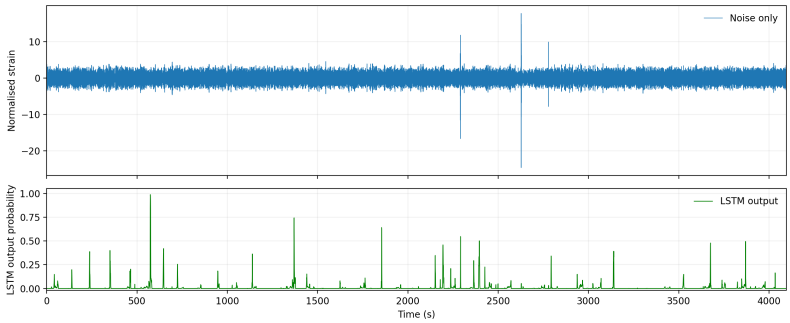
$$p(t) = \sigma(z(t)) = \frac{1}{1 + e^{-z(t)}}. \tag{6.2}$$

In this work, analyses involving the statistical characterisation of the network response in noise are primarily performed using the raw output $z(t)$, since it preserves the full dynamic range of the model response. However, for visualisation purposes and qualitative interpretation in selected figures, the corresponding sigmoid-transformed probability $p(t)$ is also shown. The trained LSTM model was applied to noise-only segments of the O3Replay data in order to evaluate its behaviour under realistic detector noise conditions. In this setting, no injected signals are present, and excursions of $z(t)$ above a chosen threshold are interpreted as potential false alarms.

Fig. 6.9 shows two representative examples of the model behaviour over full time segments. The first example corresponds to a relatively stationary, Gaussian-like noise segment, while the second contains a transient noise feature (glitch). These examples illustrate the variability of detector noise within segments that satisfy the Gaussian consistency criterion, together with the corresponding LSTM output. The strain data shown are normalised to approximately zero mean and unit variance using statistics computed independently for each 100-second segment, consistent with the preprocessing procedure applied during training. In the presence of the glitch, the LSTM output does not exhibit a strong or clearly localised increase in probability. This behaviour may arise because the model is trained to identify the extended temporal structure of long-duration inspiral signals rather than short transient disturbances. Consequently, isolated glitch-like features do not strongly resemble the gradual time-evolution patterns associated with CBC signals.



(a) Gaussian-like noise segment

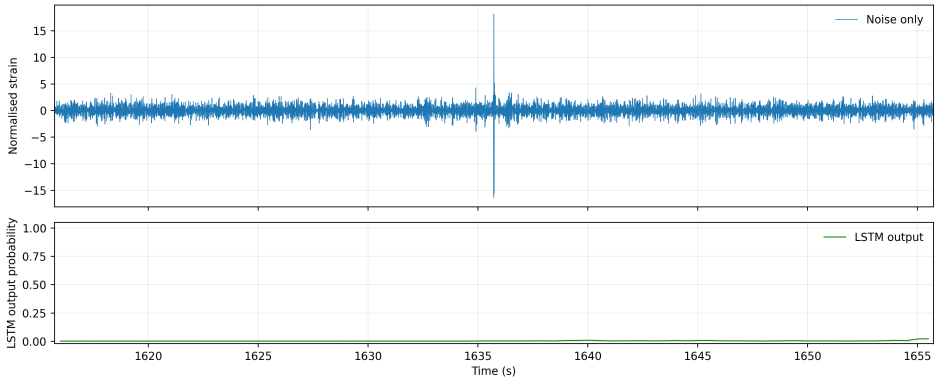


(b) Noise segment containing transient features

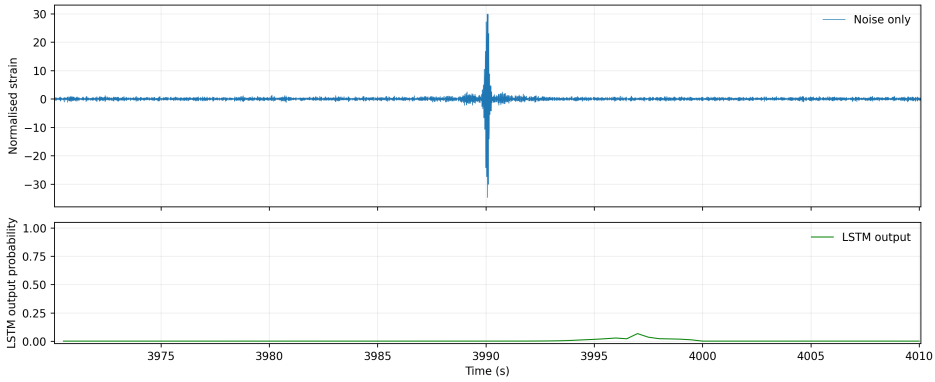
Figure 6.9: Examples of model behaviour on noise-only data from the H1 detector. (a) A segment exhibiting approximately Gaussian and stationary characteristics (GPS start time 1262309376). (b) A segment containing multiple transient noise features (glitches) (GPS start time 1262489600). In both cases, the top panel shows the normalised strain. The data are standardised to zero mean and unit variance on a per-window basis, with statistics computed independently for each 100 s input segment. The bottom panel shows the corresponding LSTM output probability.

To examine the behaviour of the trained LSTM model in greater detail, Fig. 6.10 presents zoomed-in views (± 20 s) centred on representative transient noise features identified in multiple noise-only segments. These examples are drawn from different files to illustrate the variability of non-Gaussian noise present in the dataset. In each case, the strain time series exhibits short-duration, high-amplitude deviations from the background noise, characteristic of transient noise artefacts (glitches). It is worth noting that, even away from the transient glitch, the noise does not appear

perfectly Gaussian in the time domain. Although the data have passed the standard validation and preprocessing steps, real detector noise is not strictly Gaussian or white. Residual temporal correlations and spectral structure remain after whitening, particularly when using finite-length PSD estimates.



(a) Zoomed-in view (±20 s) of a transient glitch in a noise-only segment



(b) Zoomed-in view (±20 s) of a transient glitch in a noise-only segment

Figure 6.10: Zoomed-in views (±20 s) of representative transient noise features (glitches) in noise-only segments. In each subplot, the top panel shows the normalised strain, while the bottom panel displays the corresponding LSTM output probability. These examples illustrate the model's response to short-duration, non-Gaussian noise artefacts.

To further examine the time-domain behaviour of the preprocessed detector noise, a zoomed-in segment is shown in Fig. 6.11. While the noise may appear approximately Gaussian over long durations, the zoomed-in

view reveals visible temporal structure and local amplitude variations, indicating departures from ideal Gaussian behaviour and the presence of short-timescale correlations. As a result, the strain may exhibit structured fluctuations that deviate from ideal Gaussian noise, even in apparently quiet segments. This behaviour is consistent with the PSD analysis presented later, which shows that the noise is not perfectly flat across the analysed frequency band. In these examples, the corresponding LSTM output shows only a limited response, with modest increases in the predicted probability that remain well below the detection threshold. Although small variations in the model output are observed in the vicinity of some glitches, these responses are not sustained and, in the examples shown, remain well below values associated with confident detections. These illustrative cases suggest that the model is not strongly triggered by short-duration transient noise features in such instances. However, as these examples represent only a limited subset of possible noise realisations, a systematic statistical analysis is required to assess the model response more rigorously. However, it should be noted that these examples are not necessarily representative of all transient noise features. While many glitches produce only a weak response, a subset of events can lead to significantly larger outputs, as explored in the statistical analysis that follows.

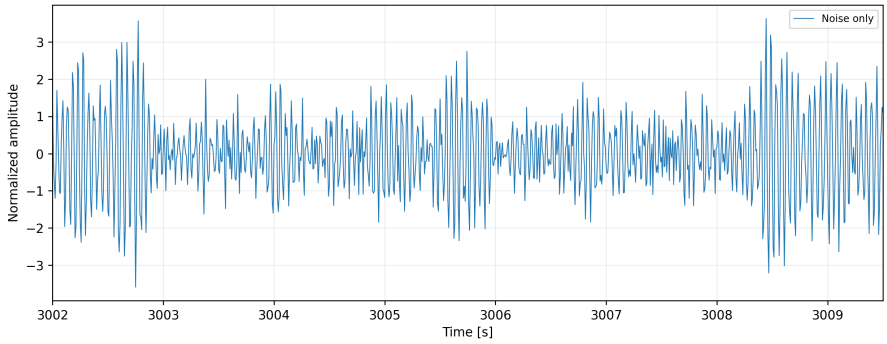


Figure 6.11: Zoomed-in view of the normalised O3Replay strain data illustrating local temporal structure and deviations from ideal Gaussian behaviour.

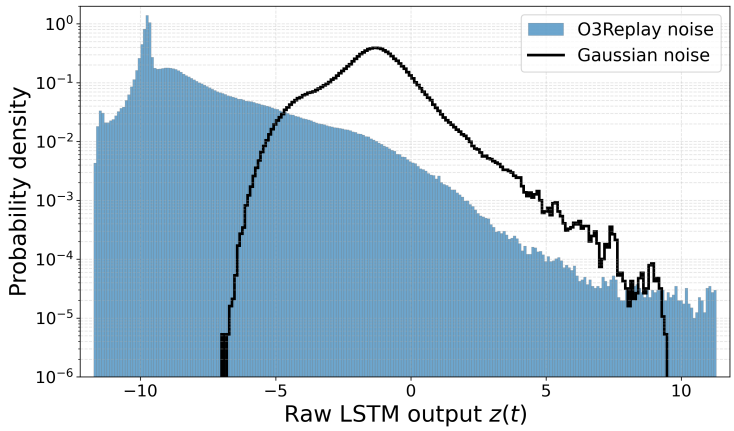


Figure 6.12: Distribution of the raw LSTM output, $z(t)$, for noise-only data in O3Replay detector noise and simulated Gaussian noise. The Gaussian-noise distribution is narrower and more regular, whereas the real detector noise exhibits broader, asymmetric heavy tails, indicating non-Gaussian structure and temporal correlations that increase the probability of large $z(t)$ excursions.

The primary statistic used to characterise the noise response of the trained LSTM model is the raw network output, denoted by $z(t)$. This statistic is preferred over the sigmoid-transformed output because it preserves the full dynamic range of the network response and provides a more informative diagnostic of differences between idealised Gaussian noise and preprocessed O3Replay detector noise. The distribution of $z(t)$ was therefore examined for noise-only O3Replay data, as shown in Fig. 6.12. For comparison, the corresponding distribution obtained from idealised Gaussian noise, constructed following the procedure described in Chapter 5, is also shown as a reference.

The comparison reveals clear differences between the two cases. The Gaussian-noise distribution is relatively smooth and concentrated over a narrower range of $(z(t))$, whereas the O3Replay distribution is broader, more asymmetric, and exhibits extended tails. In particular, the real detector noise produces a substantially wider spread of network outputs, including extended positive and negative tails that reach substantially larger $|z(t)|$ values than those populated by the Gaussian reference distri-

bution These differences indicate that the trained LSTM responds differently to the preprocessed O3Replay noise compared to idealised Gaussian noise. Although the preprocessing pipeline reduces many non-stationary and instrumental artefacts, the observed asymmetry and broadened tails are consistent with residual non-Gaussian structure in the real detector data that is not represented in the Gaussian training distribution.

Although the analysed dataset passed the KS-based selection procedure described in Sec. 6.5, the resulting $z(t)$ distribution remains significantly broader and more asymmetric than the Gaussian reference distribution. This suggests that, even after the applied selection and preprocessing steps, the retained O3Replay noise is not statistically equivalent to the idealised Gaussian noise used during training. It is also important to distinguish between Gaussianity and whiteness. Although some preprocessed noise segments may be approximately Gaussian in their amplitude distribution, the whitening procedure does not produce a perfectly flat PSD across the full analysed frequency range. Residual spectral structure remains, particularly at low frequencies and around narrow instrumental features. This indicates that the preprocessed detector noise is not strictly white and retains temporal correlations.

These observations motivate further diagnostic studies of the preprocessed detector noise, including investigations of temporal correlations, spectral structure, and normalisation consistency. Residual temporal correlations can introduce structured variations in the time-domain strain that influence the sequential response of the LSTM model. Since the network processes the strain data as a time series, such correlated structure can influence the hidden-state evolution and contribute to the asymmetric and extended tails observed in the real-noise distribution of $z(t)$.

In the present analysis, the strain data were normalised independently over fixed 100 s intervals in order to maintain consistency with the preprocessing applied during training and to reduce long-timescale variations in the local mean and variance. This local normalisation substantially improves the statistical stability of the detector noise and mitigates broadband variance fluctuations across the analysed duration. Although

residual temporal correlations and spectral structure remain present after preprocessing and normalisation, the resulting data more closely approximates the statistical conditions assumed during training.

6.6.3 Model Response to Signal Injections in Noise

The model was trained using simulated signals with inspiral SNR values in the approximate range $7 \leq \text{SNR} \leq 20$. The discussion in this section therefore concentrates primarily on injections lying within, or close to, the signal-strength regime represented during training. Injections with substantially lower SNR values are not expected to produce significant network responses, while very high-SNR injections fall outside the distribution encountered during training and may therefore exhibit behaviour that is not representative of the trained model performance. The analysis also focused on representative BNS injections that were not visibly contaminated by strong nearby glitch-like detector transients, in order to better investigate whether the observed LSTM responses were associated with the target BNS signals rather than unrelated noise artefacts.

In light of the residual non-stationary behaviour and temporal variability identified in Subsec. 6.6.1, this section does not attempt to characterise the full statistical distribution of the model outputs or establish robust false-alarm thresholds for a comprehensive sensitivity analysis. Instead, the focus is placed on examining whether the trained LSTM network produces identifiable responses to injected BNS signals embedded within realistic detector noise. For each selected injection, the corresponding model output is analysed in order to investigate the qualitative behaviour of the network response and its relationship to both the signal properties and the surrounding detector-noise environment. To better understand the visibility of the injected BNS signals within the analysed frequency band, the distribution of their SNR values is first examined.

Fig. 6.13 shows both the total SNR distribution and the accumulated SNR below 58 Hz, corresponding to the frequency range used by the low-latency analysis framework. As shown in Fig. 6.13, the majority of

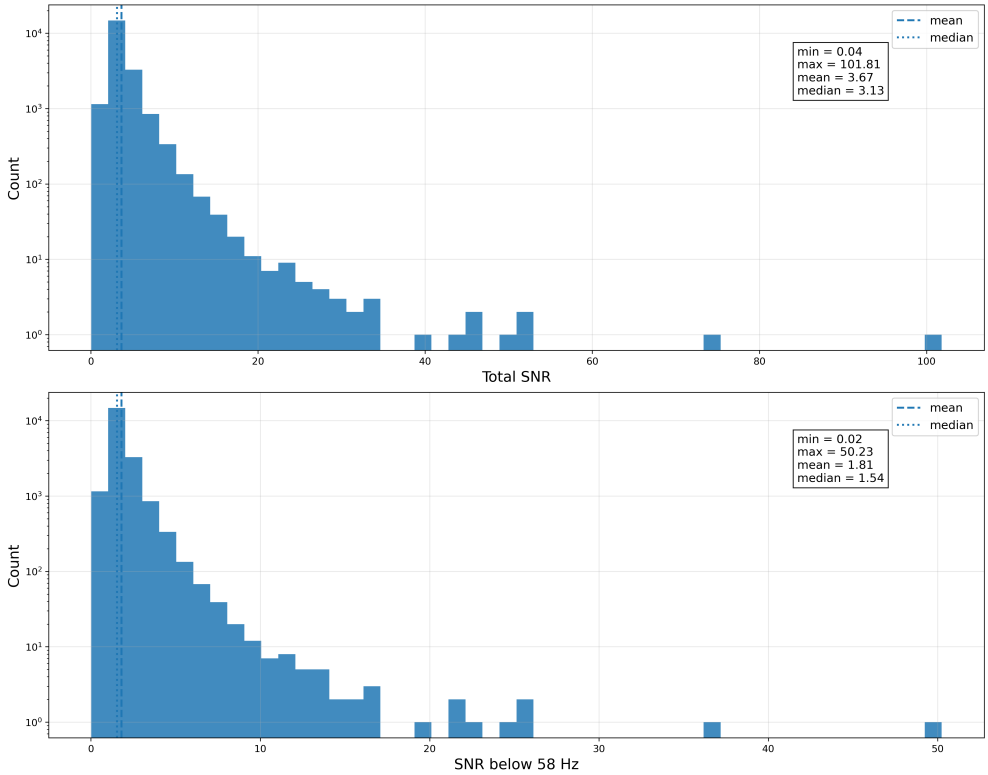


Figure 6.13: Top: Distribution of total SNR for BNS injections. Bottom: Distribution of accumulated SNR below 58 Hz, corresponding to the analysed frequency range used by the low-latency LSTM framework.

injected BNS signals occupy relatively low SNR values. The accumulated SNR below 58 Hz is reduced compared to the total SNR, indicating that only part of the full inspiral signal power contributes within the analysed frequency range used by the model. For the BNS systems considered here, the total SNR is typically approximately a factor of two larger than the accumulated SNR below 58 Hz.

The reduced visibility of some injected BNS signals in the O3Replay analysis is influenced not only by signal amplitude, but also by differences between the idealised Gaussian-noise conditions used during training and the statistical properties of real detector noise. Residual temporal and spectral structure within the detector data, together with time-dependent non-stationary behaviour, may reduce the separability between signal-related features and detector-noise fluctuations. The threshold used in

the following response plots corresponds to a FAR of one false alarm per day. Nevertheless, the trained LSTM model continues to exhibit identifiable responses to a subset of injected BNS signals within the signal regime represented during training, demonstrating that the network retains sensitivity to inspiral features embedded within realistic detector noise. Only a representative subset of the available events is shown in the following figures, since many of the remaining events exhibit qualitatively similar LSTM response behaviour.

Nevertheless, the trained LSTM model continues to exhibit identifiable responses to a subset of injected BNS signals within the signal regime represented during training. These responses indicate that the network remains responsive to inspiral-like structure in realistic detector noise, although they do not exceed the adopted FAR-based detection threshold.

To examine the model behaviour over a longer time scale, Fig. 6.14 shows the LSTM output over a representative 4000 s segment of the O3Replay data. The segment contains multiple compact-binary injections, with the times and classes of all injected BNS, NSBH, and BBH systems shown explicitly in the figure. The selected target BNS events are highlighted separately. This long-duration representation allows the timing of the complete sequence of injected events to be compared directly with the evolution of the LSTM output statistic, rather than considering only short intervals around individual target signals.

The long-duration view shows that the LSTM output exhibits substantial temporal variation throughout a segment containing many closely spaced CBC injections. However, comparison with the injection times does not reveal a simple one-to-one correspondence between individual injected events and changes in the LSTM output, nor is a consistent periodic response observed at the injection times. The long-timescale behaviour therefore cannot be explained simply as an isolated response to each individual injection.

Since the LSTM retains temporal information through its recurrent hidden state, its response at a given time may depend on the preceding temporal sequence as well as on the current input. The high temporal

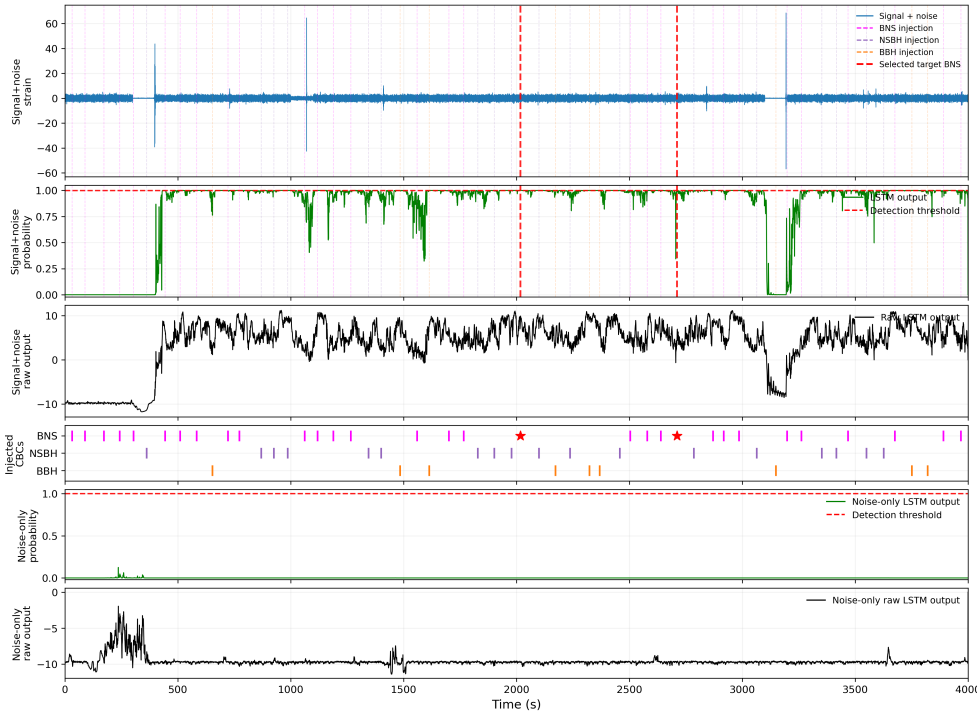


Figure 6.14: Long-duration LSTM output over a representative 4000 s segment of O3Replay data. The top panel shows the signal-plus-noise strain. The second and third panels show the corresponding LSTM probability output and raw network output for the signal-plus-noise case, respectively. The fourth panel shows the times and classes of all injected compact-binary signals in the segment, with BNS, NSBH, and BBH injections shown separately and the selected target BNS events highlighted by red stars. The fifth and sixth panels show the corresponding probability and raw output for a noise-only reference segment. The red dashed horizontal lines indicate the adopted FAR-based detection threshold.

density of injections in the O3Replay MDC therefore makes memory effects associated with preceding signals a possible contributing factor to the observed behaviour. In particular, the network may not return to an equivalent internal state before a subsequent injected signal is encountered, making it difficult to attribute a particular variation in the output uniquely to an individual target BNS.

The present figure does not establish that preceding or closely spaced injections are responsible for suppressing the response to the target BNS signals, nor does it demonstrate that such memory effects are the cause of the absence of FAR-threshold crossings. Rather, it shows that the relationship between the injection sequence and the network statistic is more complex than a direct response to individual events, and that the influence of the preceding injection history cannot be excluded from the present analysis. A controlled comparison in which otherwise identical target injections are analysed both in isolation and in the presence of preceding closely spaced signals would be required to quantify this effect. Fig. 6.15 and Fig. 6.16 show illustrative examples of the LSTM response to detector data containing injected BNS signals. The examples are selected from injections with accumulated SNR up to 58 Hz, $\mathrm{SNR}_{58\mathrm{Hz}}$, in the range approximately 10–20, corresponding to the signal regime where identifiable network responses begin to emerge. A larger number of injections were examined within this SNR range, many of which exhibited qualitatively similar LSTM response behaviour, and only representative examples are shown here for clarity. In each figure, the top subplot shows the normalised detector strain containing the injected signal, while the following subplots show the LSTM probability output, the raw LSTM output, and the corresponding response obtained from a noise-only reference segment.

The results indicate that the LSTM model can produce a positive response in the presence of injected BNS signals. In several cases, the LSTM probability increases before or near the target BNS merger time and approaches the chosen detection threshold. This suggests that the model is responding to signal-like structure in the data, rather than remaining

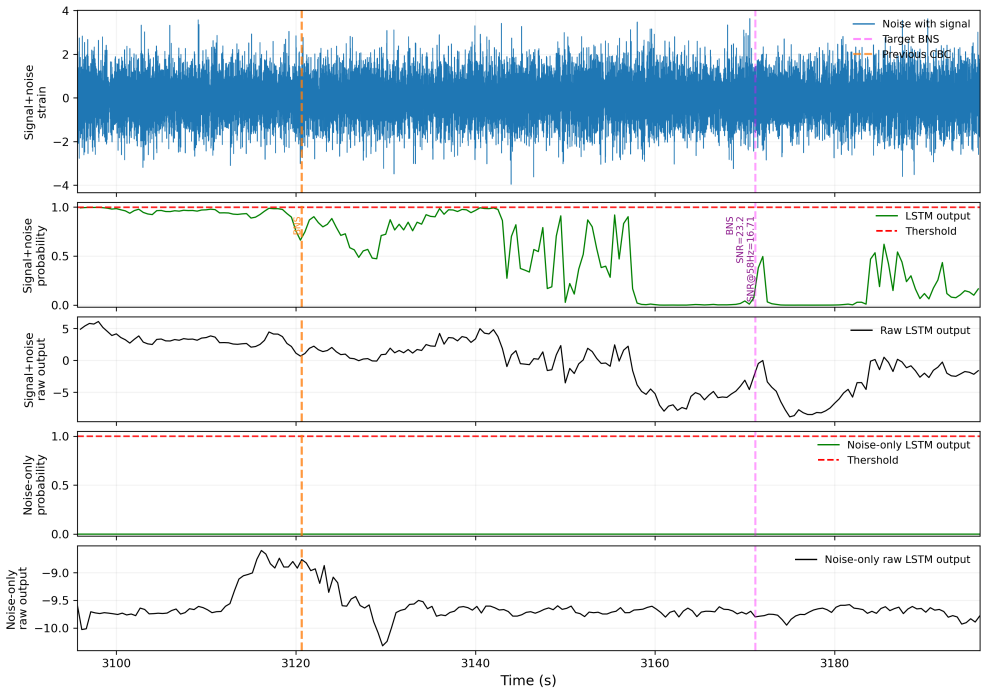


Figure 6.15: Example of a positive LSTM response to a target BNS injection embedded in detector noise. The panels show the normalised signal-plus-noise strain, the LSTM probability output with the detection threshold, the raw LSTM output, and the corresponding outputs for a reference noise-only segment. The magenta dashed line marks the target BNS merger time, while the orange dashed line marks the previous CBC injection in the same data segment.

at the background noise level. The raw LSTM output also shows corresponding changes around the signal region, consistent with a change in the network response around the injected waveform. The orange vertical dashed line marks the previous CBC injection in the same data segment, while the magenta dashed line marks the target BNS merger time. It should be noted, however, that the high rate of closely spaced injections present in the MDC represents an intentionally challenging test configuration and is not expected to reflect the typical astrophysical signal rate observed in advanced GW detectors.

These examples should therefore be interpreted as preliminary diagnostic plots. Although the model shows encouraging positive responses

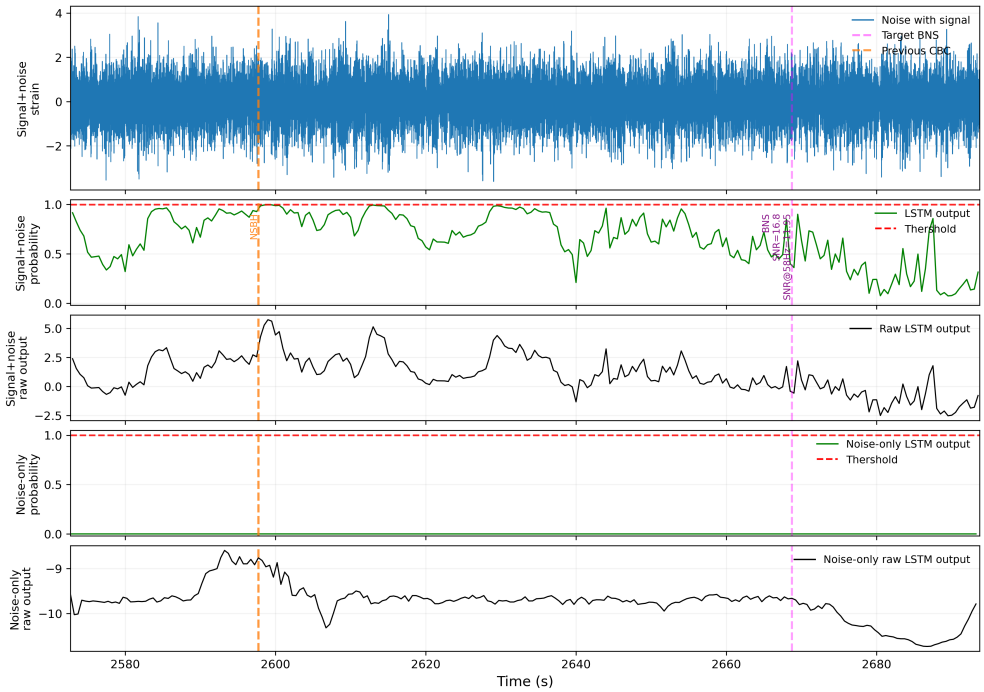


Figure 6.16: Second representative example of a positive LSTM response to a target BNS injection embedded in detector noise. The panels show the normalised signal-plus-noise strain, the LSTM probability output with the detection threshold, the raw LSTM output, and the corresponding outputs for a reference noise-only segment. The magenta dashed line marks the target BNS merger time, while the orange dashed line marks the previous CBC injection in the same data segment.

for some BNS injections with $\mathrm{SNR}_{58\mathrm{Hz}}$ between approximately 10 and 20, further investigation is required to understand the effect of previous injections, overlapping signals, glitches, and non-stationary noise on the LSTM output. While a larger number of injections within this SNR range were qualitatively examined, a more systematic population-level analysis would be required to quantify the extent to which the observed positive responses are consistently associated with the target BNS signals rather than being partly influenced by earlier CBC events or detector-noise artefacts. Although the LSTM output exhibits a distinguishable response to the signal-plus-noise data when compared with the noise-only data, the network does not yet achieve a confidence level sufficient for robust

detection. This behaviour suggests that the model has learned certain signal-sensitive temporal features; however, the output probabilities remain below the adopted detection threshold.

One possible contributing factor is the difference between the signal representations used during training and those present in the O3Replay analysis. During training, the simulated GW signals were generated directly at a sampling rate of 128 Hz. In contrast, the O3Replay signals originate from data sampled at a substantially higher rate and are subsequently resampled to 128 Hz following preprocessing procedures such as whitening and filtering. Although both datasets share the same final sampling rate, differences introduced by the preprocessing and resampling procedures may affect the detailed signal morphology presented to the LSTM model and could potentially influence the network response.

Another possible contributing factor is the temporal overlap between injected signals in the O3Replay data. In some cases, more than one CBC signal may be present within the same analysis window, whereas the model was trained primarily on examples containing either noise alone or a single injected signal. Such overlapping injections may increase the complexity of the signal-plus-noise data presented to the LSTM and could potentially influence the subsequent network response through the temporal memory carried by the hidden state. A more systematic investigation using controlled signal injections in Gaussian noise would be required to quantify the extent of this effect and determine the characteristic timescale over which the LSTM hidden state naturally resets following a preceding signal. These effects require further investigation in future work, particularly in relation to the impact of preprocessing procedures, sampling strategies, and overlapping injected signals on the generalisation capability of deep learning models when applied to realistic detector data.

6.7 Discussion and Conclusion

In this chapter, the behaviour of the trained LSTM model was evaluated on preprocessed O3Replay detector data. The analysis focused

on understanding the model response to both noise-only data and injected BNS signals under realistic detector conditions. The primary objective of this chapter is to evaluate the behaviour and limitations of the trained LSTM early-warning model under realistic detector conditions, with particular emphasis on identifying the dominant sources of domain mismatch between the idealised Gaussian-noise training environment and real detector noise. The analysis is therefore designed as a generalisation study rather than as a production-level search or sensitivity benchmark on O3Replay data.

Although only the low-frequency inspiral portion of the waveform is accessible to the model, this frequency range is expected to contain sufficient information for early-warning detection of a subset of higher-SNR BNS signals. However, the O3Replay analysis indicates that the response of the trained LSTM model is sensitive to differences between the idealised Gaussian training conditions and the statistical properties of the real detector environment. In particular, residual temporal and spectral structure in the detector noise, together with time-dependent statistical variability, may reduce the separability between signal-related features and structured detector-noise fluctuations. Consequently, although the model continues to exhibit responses to some injected signals, these responses are often less distinct and less consistently separable from the surrounding detector-noise behaviour than observed under the simulated training conditions.

However, the behaviour of the model cannot be fully explained by the signal properties alone. The additional statistical diagnostics presented in this chapter show that the preprocessed detector data retains residual temporal and spectral structure after whitening, including non-stationary behaviour and time-dependent spectral variability. In this analysis, the detector data were high-pass filtered above approximately 18 Hz prior to whitening in order to reduce the impact of strong low-frequency detector noise [10, 114]. The whitening procedure was performed using PSD estimates obtained directly from the data using Welch's

method. Although these preprocessing steps improve the statistical behaviour of the detector data, the additional diagnostics presented in this chapter demonstrate that residual temporal and spectral structure remains present after preprocessing and whitening. This behaviour is not unexpected, since whitening based on a finite-duration PSD estimate can only approximately flatten the detector spectrum and cannot fully remove time-dependent non-stationarity or residual instrumental structure from real interferometer data.

These effects highlight a mismatch between the idealised Gaussian noise used during training and the real detector noise used during evaluation. Consequently, the LSTM model is influenced not only by true signal features, but also by structured noise variations present in the detector data. These results suggest that residual temporal correlations and spectral variability in the detector noise may influence the sequential response of the model and reduce the separability between signal-related features and structured detector-noise fluctuations. Although such effects may potentially be mitigated through improved preprocessing or adaptive normalisation strategies, they nevertheless illustrate an important difference between the idealised Gaussian training conditions and the real detector environment used during evaluation.

The main limitation of this analysis lies in the simplified training conditions, where the model was trained on idealised stationary Gaussian noise that does not fully reproduce the complexity of real detector data. Although the preprocessing pipeline reduces many instrumental and low-frequency noise contributions, residual non-Gaussian and non-stationary structure remains present in the detector environment. Future work should focus on improving the consistency between training and evaluation data. This includes training or fine-tuning the model on more realistic detector noise, incorporating adaptive preprocessing and PSD estimation methods [34, 122], and investigating the effect of different frequency cut-offs on model behaviour. Further analysis is also required to better understand the temporal behaviour of RNNs and their response

to residual non-Gaussian and non-stationary noise features in realistic detector data.

Overall, this chapter demonstrates that differences between the statistical properties of the simulated training data and the real detector environment can significantly influence the behaviour of ML models applied to GW data. These results highlight the importance of realistic noise modelling, robust preprocessing, and careful treatment of detector non-stationarity in the development of reliable ML approaches for GW detection.

CHAPTER 7

Conclusions and Future Work

7.1 Conclusions

The first direct detection of GWs in 2015 marked a major milestone in modern physics, opening a new observational window onto the Universe. Since then, GW astronomy has advanced rapidly, enabling the detection of numerous compact binary mergers and providing unprecedented insight into the properties of black holes, neutron stars, and the behaviour of gravity in extreme regimes. These developments have also established the foundation of MMA, in which GW observations are combined with EM signals to provide a more complete understanding of transient astrophysical phenomena. A central challenge in this field is the development of low-latency detection methods capable of identifying GW signals prior to merger. Early-warning detection is particularly critical for systems involving neutron stars, where EM counterparts evolve on short timescales and require rapid observational response. However, detecting signals during the early inspiral phase remains inherently difficult due to their low amplitude and the presence of complex, non-Gaussian detector noise.

This thesis demonstrates that ML, and in particular LSTM neural networks, provides a viable and effective framework for early-warning detection of GW signals. The results establish that data-driven models can identify signals during the inspiral phase and provide meaningful advance warning prior to merger. This investigation was conducted through a sequence of progressively more realistic studies. In Chapter 4, an LSTM-based model was developed for BBH systems. Despite the short-duration nature of these signals, the model successfully identified GW signatures several seconds prior to merger. This result demonstrates that LSTM networks can extract meaningful temporal features even in rapidly evolving and limited pre-merger regimes.

In Chapter 5, the approach was extended to BNS systems, which provide a more favourable scenario for early-warning detection due to their longer inspiral phase and the high likelihood of associated EM emission, in contrast to BBH systems. The results show that the model achieves early-warning times of tens of seconds prior to merger, reaching up to approximately one minute depending on the signal strength. Furthermore, the detection performance exhibits a smooth and consistent dependence on the SNR, indicating that the model learns a stable and physically meaningful detection statistic. These findings establish that early-warning detection is achievable on astrophysically relevant timescales and strongly depends on the intrinsic properties of the source.

In Chapter 6, the trained LSTM model was evaluated on preprocessed O3Replay detector data in order to investigate its behaviour under realistic detector-noise conditions. The analysis showed that the model remains relatively controlled when applied to noise-only detector data, without producing excessive false triggering. However, the responses to injected BNS signals were generally less distinct and less consistently separable than those observed under the idealised Gaussian training conditions.

Additional statistical diagnostics demonstrated that the preprocessed detector data retains residual temporal and spectral structure after whitening, introducing a mismatch between the simulated Gaussian training environment and the real detector conditions used during evaluation. These results highlight the sensitivity of recurrent neural-network models to the statistical properties of realistic detector noise and emphasise the importance of realistic noise modelling and robust preprocessing in the development of reliable machine-learning approaches for GW early-warning detection.

Taken together, the results of this thesis demonstrate that LSTM-based models can identify GW inspiral signals prior to merger under idealised conditions. This establishes that early-warning detection is feasible on physically meaningful timescales relevant for low-latency observational strategies. While these models are not intended to replace es-

tablished matched-filtering pipelines, they provide a complementary approach with potential advantages in computational efficiency and rapid inference, making them attractive for future real-time applications.

Despite these advances, several limitations remain. The models developed in this thesis were primarily trained on simulated signals embedded in stationary Gaussian noise, which does not fully capture the complexity of real detector data. Real detector noise is inherently non-Gaussian, non-stationary, and contains a wide range of transient artefacts and time-dependent spectral features that can affect model behaviour and reduce generalisation performance. In addition, the current implementation focuses on single-detector time-series data, whereas practical GW detection relies on a network of detectors. Multi-detector analyses are also essential for accurate sky localisation, which is critical for enabling rapid EM follow-up observations and MMA.

Future work should therefore focus on improving the consistency between training and evaluation conditions. This includes training ML models on more realistic detector noise, incorporating adaptive or local PSD estimation methods, investigating the impact of different frequency cutoffs, and extending the analysis to multi-detector frameworks. Further investigation is also required to better understand the temporal behaviour of RNNs in realistic detector environments and their response to residual non-Gaussian and non-stationary noise structure.

Overall, this thesis demonstrates that ML can contribute directly to physically meaningful inference in GW astronomy. In particular, the ability to identify inspiral signatures prior to merger establishes early-warning detection as a promising component of future low-latency GW detection strategies. At the same time, the results highlight the importance of realistic noise modelling, robust preprocessing, and careful treatment of detector non-stationarity in the development of reliable ML approaches for GW data analysis.

7.2 Future Work

While this thesis establishes the feasibility of early-warning detection using LSTM-based models, several important directions remain for further development.

A key priority for future work is the direct training and evaluation of models on real detector data. Training on realistic, non-Gaussian and non-stationary noise would allow the model to learn the more realistic statistical properties of the detector environment, including transient noise artefacts and time-varying noise characteristics. This would likely require the incorporation of data quality information and glitch mitigation strategies, as well as careful handling of non-stationarity through techniques such as adaptive normalisation or time-dependent feature extraction. Such developments are essential for improving robustness and reducing false detections in realistic observational settings. In the present study, the model was not trained directly on real detector data, but only evaluated on O3Replay data, which may limit its ability to fully generalise to real observations.

Another important direction is the extension to multi-detector analysis. The present work considers single-detector inputs, whereas practical GW detection relies on coincident observations across a network of detectors such as LIGO, Virgo, and KAGRA. Incorporating multi-detector data would improve detection confidence and reduce FARs by exploiting coherence between detectors. This could be achieved by designing architectures that take multiple time-series inputs simultaneously, or by incorporating inter-detector time delays and correlations as explicit features. A key challenge in this context is the alignment of signals across detectors, particularly in the low-latency regime where precise timing information must be inferred in real time.

A further and particularly important extension is the development of sky localisation capabilities within the LSTM framework. While the current model focuses on detection and early-warning timing, it does not estimate the source location. Future work could explore architectures that learn spatial information from multi-detector time-series data, enabling

rapid inference of both the sky position of the source and the associated uncertainty region. This would likely require the model to learn timing differences and amplitude consistency across detectors, which encode directional information. Producing reliable uncertainty estimates represents an additional challenge, as this requires moving beyond deterministic predictions toward probabilistic modelling. Such information is essential for guiding EM follow-up observations. While localisation may be provided by external analysis pipelines, its availability in the pre-merger regime is critical for enabling early-warning detections to contribute meaningfully to MMA.

It is important to recognise the scope and limitations of the present work. The models developed in this thesis were primarily trained on simulated signals embedded in stationary Gaussian noise. While this provides a controlled and well-defined environment for model development, real detector noise is non-Gaussian, non-stationary, and contains a wide range of transient artefacts. As such, further work is required to fully characterise model performance under realistic observational conditions, including systematic evaluation on real data and comparison with established search pipelines.

More broadly, this thesis demonstrates that ML can contribute directly to physically meaningful inference in GW astronomy. In particular, the ability to identify signals during the inspiral phase and anticipate merger events shows that early-warning detection is achievable and can play a direct role in enabling low-latency, predictive detection systems for MMA. In summary, advancing robustness to realistic noise, incorporating multi-detector information, and enabling rapid sky localisation with associated uncertainty estimates will be essential for developing fully real-time, low-latency GW detection systems. These developments are necessary for enabling EM follow-up observations and maximising the scientific impact of future multi-messenger observations.

References

[1] Reem Alfaidi and Christopher Messenger. Long short-term memory for early warning detection of gravitational waves. *Classical and Quantum Gravity*, 43(8):085003, 2026.

[2] Michele Maggiore. *Gravitational Waves: Volume 1: Theory and Experiments*. (Oxford Academic, 2007). DOI: 10.1093/acprof:oso/9780198570745.001.0001.

[3] Jolien D. E. Creighton and Warren G. Anderson. *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis*. (Wiley-VCH, 2011).

[4] LIGO Caltech: LIGO's Interferometer. [Online]. Available: https://www.ligo.caltech.edu/page/ligos-ifo [Accessed: April 28, 2026].

[5] F. Acernese et al. 2015. Advanced Virgo: A Second-Generation Interferometric Gravitational Wave Detector. *Class. Quantum Grav.* **32**(2), 024001. DOI: 10.1088/0264-9381/32/2/024001.

[6] J Aasi et al. 2015. Advanced LIGO. *Class. Quantum Grav.* **32**, 074001. DOI: 10.1088/0264-9381/32/7/074001.

[7] B. P. Abbott et al. 2020. Prospects for observing and localizing gravitational-wave transients with Advanced LIGO, Advanced Virgo and KAGRA. *Living Rev. Relativ*, **23**(3). DOI: 10.1007/s41114-020-00026-9.

[8] M Cabero et al. 2019. Blip glitches in Advanced LIGO data. *Class. Quantum Grav.* **36**, 155010. DOI: 10.1088/1361-6382/ab2e14.

[9] M Zevin et al. 2017. Gravity Spy: Integrating Advanced LIGO Detector Characterization, Machine Learning, and Citizen Science. *Class. Quantum Grav.* **34**, 064003. DOI: 10.1088/1361-6382/aa5cea.

[10] B. P. Abbott et al. 2016. Observation of Gravitational Waves from a Binary Black Hole Merger. *Phys. Rev. Lett.* **116**, 061102. DOI: 10.1103/PhysRevLett.116.061102.

[11] Piotr Jaranowski and Andrzej Królak. *Analysis of Gravitational-Wave Data*. (Cambridge University Press, 2009).

[12] T. Akutsu et al. 2020. Overview of KAGRA: Detector design and construction history. arXiv preprint: arXiv:2005.05574v2.

[13] K. L. Dooley et al. 2016. GEO 600 and the GEO-HF Upgrade Program: Successes and Challenges. *Class. Quantum Grav.* **33**(7). DOI: 10.1088/0264-9381/33/7/075009.

[14] A. G. Abac et al. 2025. GWTC-4.0: Updating the Gravitational-Wave Transient Catalog with Observations from the First Part of the Fourth LIGO-Virgo-KAGRA Observing Run. arXiv preprint: arXiv:2508.18082.

[15] LIGO Caltech: Sources and Types of Gravitational Waves. [Online]. Available: https://www.ligo.caltech.edu/page/gw-sources [Accessed: April 28, 2026].

[16] Piotr Jaranowski, Andrzej Królak and Bernard F. Schutz. 1998. Data analysis of gravitational-wave signals from spinning neutron stars. I. The signal and its detection. arXiv preprint: arXiv:gr-qc/9804014.

[17] R. Abbott et al. 2022. All-sky search for continuous gravitational waves from isolated neutron stars using Advanced LIGO and Advanced Virgo O3 data. *Phys. Rev. D.* **106**. DOI: 10.1103/PhysRevD.106.102008.

[18] Tim Dietrich et al. 2019. Improving the NRTidal model for binary neutron star systems. *Phys. Rev. D.* **100**(4), 044003. DOI: 10.1103/PhysRevD.100.044003.

[19] Einstein@Home. [Online]. Available: https://einsteinathome.org/ar [Accessed: April 28, 2026].

[20] Man Leong Chan, Ik Siong Heng and Chris Messenger. 2020. Detection and classification of supernova gravitational wave signals: A deep learning approach. *Phys. Rev. D.* **102**(4), 043022. DOI: 10.1103/PhysRevD.102.043022.

[21] Curt Cutler and Éanna E. Flanagan. 1994. Gravitational waves from merging compact binaries: How accurately can one extract the binary's parameters from the inspiral waveform? *Phys. Rev. D.* **49**(6), 2658-2697. DOI: 10.1103/PhysRevD.49.2658.

[22] B. P. Abbott et al. 2018. GW170817: Measurements of Neutron Star Radii and Equation of State. *Phys. Rev. Lett.* **121**(16), 161101. DOI: 10.1103/PhysRevLett.121.161101.

[23] Alexander H. Nitz et al. 2018. Rapid detection of gravitational waves from compact binary mergers with PyCBC Live. *Phys. Rev. D.* **98**(2), 024050. DOI: 10.1103/PhysRevD.98.024050.

[24] F Aubin et al. 2021. The MBTA pipeline for detecting compact binary coalescences in the third LIGO–Virgo observing run. *Class. Quantum Grav.* **38**(9), 095004. DOI: 10.1088/1361-6382/abe913.

[25] Ethan Marx et al. 2025. Machine-learning pipeline for real-time detection of gravitational waves from compact binary coalescences. *Phys. Rev. D.* **111**(4), 042010. DOI: 10.1103/PhysRevD.111.042010.

[26] B. S. Sathyaprakash and S. V. Dhurandhar. 1991. Choice of filters for the detection of gravitational waves from coalescing binaries. *Phys. Rev. D.* **44**(12), 3819-3834. DOI: 10.1103/PhysRevD.44.3819.

[27] Shio Sakon et al. Template bank for compact binary mergers in the fourth observing run of advanced ligo, advanced virgo, and kagra. 2022.

[28] Benjamin J. Owen and B. S. Sathyaprakash. 1999. Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement. *Phys. Rev. D.* **60**(2), 022002. DOI: 10.1103/PhysRevD.60.022002.

[29] Aashish Tripathee and Keith Riles. 2024. Probing More Deeply in an All-Sky Search for Continuous Gravitational Waves in the LIGO O3 Data Set. arXiv preprint: arXiv:2311.04985.

[30] R. Abbott et al. 2021. Searches for Continuous Gravitational Waves from Young Supernova Remnants in the Early Third Observing Run of Advanced LIGO and Virgo. *ApJ.* **921**. DOI: 10.3847/1538-4357/ac17ea.

[31] J. Ming et al. 2024. Deep Einstein@Home Search for Continuous Gravitational Waves from the Central Compact Objects in the Supernova Remnants Vela Jr. and G347.3-0.5 Using LIGO Public Data. *ApJ.* **977**(2). DOI: 10.3847/1538-4357/ad8b9e.

[32] Jorge Morales et al. 2025. Results from an Einstein@Home search for continuous gravitational waves from Cassiopeia A and Vela Jr. using LIGO O2 data. arXiv preprint: arXiv:2503.09731.

[33] S. Babak et al. 2013. Searching for gravitational waves from binary coalescence. *Phys. Rev. D.* **87**. DOI: 10.1103/PhysRevD.87.024033.

[34] Samantha A. Usman. 2016. The PyCBC search for gravitational waves from compact binary coalescence. *Class. Quantum Grav.* **33**(21). DOI: 10.1088/0264-9381/33/21/215004.

[35] R. Abbott et al. 2021. GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run. arXiv preprint: arXiv:2010.14527.

[36] Péter Mészáros et al. 2019. Multi-messenger Astrophysics. *Nature Reviews Physics.* **7**, 585–599. DOI: 10.1038/s42254-019-0101-z.

[37] Ankur Sharma. 2024. Multi-Messenger Connection in High-Energy Neutrino Astronomy. *Universe*. **10**(8). DOI: 10.3390/universe10080326.

[38] Hugo A. Ayala Solares et al. 2020. The Astrophysical Multimessenger Observatory Network (AMON): Performance and science program. *Astroparticle Physics*. **114**, 68-76. DOI: 10.1016/j.astropartphys.2019.06.007.

[39] Jamee Elder. 2024. Independent evidence in multi-messenger astrophysics. *Studies in History and Philosophy of Science*. **104**, 119-129. DOI: 10.1016/j.shpsa.2024.02.006.

[40] Leo P. Singer et al. 2014. The First Two Years of Electromagnetic Follow-Up with Advanced LIGO and Virgo. *ApJ*. **795**(2), 105. DOI: 10.1088/0004-637X/795/2/105.

[41] B. D. Metzger and E. Berger. 2012. What Is the Most Promising Electromagnetic Counterpart of a Neutron Star Binary Merger? *ApJ*. **746**(1), 48. DOI: 10.1088/0004-637X/746/1/48.

[42] B. P. Abbott et al. 2016. Localization and Broadband Follow-Up of the Gravitational-Wave Transient GW150914. *ApJL*. **826**(1). DOI: 10.3847/2041-8205/826/1/L13.

[43] B. P. Abbott et al. 2017. Multi-messenger Observations of a Binary Neutron Star Merger. *ApJL*. **848**. DOI: 10.3847/2041-8213/aa91c9.

[44] A. Goldstein et al. 2017. An Ordinary Short Gamma-Ray Burst with Extraordinary Implications: Fermi-GBM Detection of GRB 170817A. *ApJL*. **848**. DOI: 10.3847/2041-8213/aa8f41.

[45] LIGO Scientific Collaboration, Virgo Collaboration, and KAGRA Collaboration. Public alerts user guide. https://emfollow.docs.ligo.org/. Accessed 2026.

[46] M. Sasada et al. J-gem optical and near-infrared follow-up of gravitational wave events during ligo's and virgo's third observing run. *Progress of Theoretical and Experimental Physics*, 2021(5):05A104, 2021.

[47] Masaomi Tanaka et al. Kilonova from post-merger ejecta as an optical and near-infrared counterpart of gw170817. *Publications of the Astronomical Society of Japan*, 69(6):102, 2017.

[48] E. Troja et al. The x-ray counterpart to the gravitational-wave event gw170817. *Nature*, 551:71–74, 2017.

[49] Surabhi Sachdev et al. 2019. The GstLAL Search Analysis Methods for Compact Binary Mergers in Advanced LIGO's Second and Advanced Virgo's First Observing Runs. arXiv preprint: arXiv:1901.08580.

[50] Tito Dal Canton et al. 2022. Realtime search for compact binary mergers in Advanced LIGO and Virgo's third observing run using PyCBC Live. arXiv preprint: arXiv:2008.07494.

[51] Qi Chu. 2021. The SPIIR online coherent pipeline to search for gravitational waves from compact binary coalescences. arXiv preprint: arXiv:2011.06787.

[52] R. Abbott et al. 2022. 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.

[53] Surabhi Sachdev et al. 2020. An Early-warning System for Electromagnetic Follow-up of Gravitational-wave Events. *ApJL.* **905**(2). DOI: 10.3847/2041-8213/abc753.

[54] Kipp Cannon et al. 2012. Toward Early-Warning Detection of Gravitational Waves from Compact Binary Coalescence. *ApJ.* **748**(2), 136. DOI: 10.1088/0004-637X/748/2/136.

[55] Konstantin Otto, Micaela Oertel and Bernd-Jochen Schaefer. 2020. Hybrid and quark star matter based on a nonperturbative equation of state. *Phys. Rev. D.* **101**. DOI: 10.1103/PhysRevD.101.103021.

[56] Leo P. Singer. 2016. Going the Distance: Mapping Host Galaxies of LIGO and Virgo Sources in Three Dimensions Using Local Cosmography and Targeted Follow-Up. *ApJL.* **829**(1). DOI: 10.3847/2041-8205/829/1/L15.

[57] Wei Wei and E. A. Huerta. 2021. Deep learning for gravitational wave forecasting of neutron star mergers. *Physics Letters B.* **816**. DOI: 10.1016/j.physletb.2021.136185.

[58] Martin J. Dyer et al. Gravitational-wave optical transient observer (goto): prototype performance and prospects for transient science. *Monthly Notices of the Royal Astronomical Society*, 511(2):2405–2421, 2022.

[59] Marc Peter Deisenroth, A. Aldo Faisal and Cheng Soon Ong. *Mathematics for Machine Learning.* (Cambridge University Press, 2020). DOI: 10.1017/9781108679930.

[60] Rahul Biswas et al. 2013. Application of machine learning algorithms to the study of noise artifacts in gravitational-wave data. *Phys. Rev. D.* **88**(6), 062003. DOI: 10.1103/PhysRevD.88.062003.

[61] Matteo Scialpi. 2023. A machine learning approach to parameter inference in gravitational-wave signal analysis. Master's. Università degli Studi di Padova.

[62] Hunter Gabbard et al. 2018. Matching Matched Filtering with Deep Networks for Gravitational-Wave Astronomy. *Phys. Rev. Lett.* **120**(14), 141103. DOI: 10.1103/PhysRevLett.120.141103.

[63] Marlin B. Schäfer, Frank Ohme and Alexander H. Nitz. 2020. Detection of gravitational-wave signals from binary neutron star mergers using machine learning. *Phys. Rev. D.* **102**(6), 063015. DOI: 10.1103/PhysRevD.102.063015.

[64] Stefano Schmidt et al. 2021. Machine Learning Gravitational Waves from Binary Black Hole Mergers. *Phys. Rev. D.* **103**(4), 043020. DOI: 10.1103/PhysRevD.103.043020.

[65] Elena Cuoco et al. 2025. Applications of machine learning in gravitational-wave research with current interferometric detectors. *Living Rev. Relativ.* **28**(2). DOI: 10.1007/s41114-024-00055-8.

[66] Ibomoiye Domor Mienye, Theo G. Swart and George Obaido. 2024. Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information.* **15**(9), 517. DOI: 10.3390/info15090517.

[67] Pierre Baldi and Gianluca Pollastri. 2003. The Principled Design of Large-Scale Recursive Neural Network Architectures-DAG-RNNs and the Protein Structure Prediction Problem. *AIP.* **4**(4), 575-602. DOI: 10.1162/153244304773936054.

[68] Alexander Jung. 2022. Machine Learning: The Basics. arXiv preprint: arXiv:1805.05052.

[69] Gireen Naidu. 2023. A Review of Evaluation Metrics in Machine Learning Algorithms. Computer Science On-line Conference. DOI: 10.1007/978-3-031-35314-7_2.

[70] Rung-Ching Chen et al. 2020. Selecting critical features for data classification based on machine learning methods. *J Big Data.* **7**. DOI: 10.1186/s40537-020-00327-4.

[71] J. E. T. Akinsola et al. 2019. Performance Evaluation of Supervised Machine Learning Algorithms Using Multi-Criteria Decision Making Techniques. ITED.

[72] Alexei Botchkarev. 2019. A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. *IJIKM*. **14**, 045-076. DOI: 10.28945/4184.

[73] Peter J. Rousseeuw. 1987. Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Journal of Computational and Applied Mathematics*. **20**. DOI: 10.1016/0377-0427(87)90125-7.

[74] David L. Davies and Donald W. Bouldin. 1979. A Cluster Separation Measure. *IEEE*. **PAMI-1**(2), 224-227. DOI: 10.1109/TPAMI.1979.4766909.

[75] T. Caliński and J. Harabasz. 1974. A Dendrite Method for Cluster Analysis. *Communications in Statistics*. **3**(1), 1-27. DOI: 10.1080/03610927408827101.

[76] L. Maaten and Geoffrey E. Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*. **9**, 2579-2605. ID: 5855042.

[77] Erica Briscoe and Jacob Feldman. 2011. Conceptual complexity and the bias/variance tradeoff. *Cognition*. **118**(1), 2-16. DOI: 10.1016/j.cognition.2010.10.004.

[78] Mikhail Belkin et al. 2019. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *PNAS*. **116**(32), 15849-15854. DOI: 10.1073/pnas.1903070116.

[79] Hardev Ranglani. 2024. Empirical Analysis Of The Bias-Variance Tradeoff Across Machine Learning Models. *MLAIJ*. **11**(11), 12. DOI: 10.2139/ssrn.5086450.

[80] Xingquan Guan and Henry Burton. 2022. Bias-variance tradeoff in machine learning: Theoretical formulation and implications to structural engineering applications. *Structures*. **46**, 17-30. DOI: 10.1016/j.istruc.2022.10.004.

[81] Xue Ying. 2019. An Overview of Overfitting and its Solutions. *J. Phys.: Conf.* **1168**(2). DOI: 10.1088/1742-6596/1168/2/022022.

[82] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. ACM Symposium on Theory of Computing. DOI: 10.1145/276698.276876.

[83] James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research.* **13**, 281-305. DOI: 10.5555/2188385.2188395.

[84] Luca Franceschi et al. 2017. Forward and Reverse Gradient-Based Hyperparameter Optimization. arXiv preprint: arXiv:1703.01785.

[85] Ishthaq Ahamed and Dr.Shaheda Akthar. 2016. A Study on Neural Network Architectures. *Computer Engineering and Intelligent Systems.* **7**, 1-7. ID: 55272667.

[86] Ajay Shrestha and Ausif Mahmood. 2019. Review of Deep Learning Algorithms and Architectures. *IEEE.* **7**, 53040-53065. DOI: 10.1109/ACCESS.2019.2912200.

[87] Andriy Burkov. *The Hundred-Page Language Models Book.* (Self-published, 2025). Available: https://thelmbook.com.

[88] David Hunter et al. 2012. Selection of Proper Neural Network Sizes and Architectures—A Comparative Study. *IEEE.* **8**(2), 228-240. DOI: 10.1109/TII.2012.2187914.

[89] F. Rosenblatt. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *APA PsycArticles.* **65**(6), 386–408. DOI: 10.1037/h0042519.

[90] Douglas Williams et al. 2021. Deep Learning and its Application for Healthcare Delivery in Low and Middle Income Countries. *Front. Artif. Intell.* **4**. DOI: 10.3389/frai.2021.553987.

[91] Thomas Epelbaum. 2017. Deep learning: Technical introduction. arXiv preprint: arXiv:1709.01412.

[92] Amirhossein Ostovar, Danial Davani Davari and Maciej Dzikuć. 2025. Determinants of Design with Multilayer Perceptron Neural Networks: A Comparison with Logistic Regression. *Sustainability*. **17**(6). DOI: 10.3390/su17062611.

[93] Jiawei Zhang. 2019. Gradient Descent-Based Optimization Algorithms for Deep Learning Models Training. arXiv preprint: arXiv:1903.03614.

[94] Rudolf Kruse et al. *Computational Intelligence: A Methodological Introduction*. (Springer, 2022). DOI: 10.1007/978-3-030-42227-1.

[95] Saahil Afaq and S. Rao. 2020. Significance Of Epochs On Training A Neural Network. *International Journal of Scientific & Technology Research*. **9**, 485-488. ID: 225647672.

[96] Haidong Li et al. 2019. Research on Overfitting of Deep Learning. CIS. DOI: 10.1109/CIS.2019.00025.

[97] B. T. Polyak. 1964. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*. **4**(5), 1-17. DOI: 10.1016/0041-5553(64)90137-5.

[98] David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*. **323**, 533–536. DOI: 10.1038/323533a0.

[99] Manju Maurya and Neha Yadav. *A Comparative Analysis of Gradient-Based Optimization Methods for Machine Learning Problems*. (Springer, 2023). DOI: 10.1007/978-981-99-3432-4$_7$.

[100] Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. arXiv preprint: arXiv:1412.6980.

[101] Paul Werbos. 1974. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. PhD. Harvard University.

[102] Andrew L. Maas et al. 2012. Recurrent Neural Networks for Noise Reduction in Robust ASR. ISCA's. DOI: 10.21437/Interspeech.2012-6.

[103] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.* **9**(8), 1735-1780. DOI: 10.1162/neco.1997.9.8.1735.

[104] M. Schuster and K. K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE.* **45**(11), 2673-2681. DOI: 10.1109/78.650093.

[105] Zachary C. Lipton, John Berkowitz and Charles Elkan. 2015. A Critical Review of Recurrent Neural Networks for Sequence Learning. arXiv preprint: arXiv:1506.00019.

[106] P.J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *IEEE.* **78**(10), 1550-1560. DOI: 10.1109/5.58337.

[107] Avrim L. Blum and Ronald L. Rivest. 1993. Machine Learning: From Theory to Applications. *Springer.* **661**, 9–28. DOI: 10.1007/3-540-56483-7.

[108] Y. Bengio, P. Simard and P. Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE.* **5**(2), 157-166. DOI: 10.1109/72.279181.

[109] John F. Kolen and Stefan C. Kremer. *Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies.* (Wiley-IEEE Press, 2001). DOI: 10.1109/9780470544037.ch14.

[110] Razvan Pascanu, Tomas Mikolov and Yoshua Bengio. 2013. On the difficulty of training Recurrent Neural Networks. arXiv preprint: arXiv:1211.5063.

[111] Ronald J. Williams and David Zipser. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1**(2), 270–280. DOI: 10.1162/neco.1989.1.2.270.

[112] Felix A. Gers, Jürgen Schmidhuber and Fred Cummins. 2000. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **12**(10), 2451–2471. DOI: 10.1162/089976600300015015.

[113] Md Zahangir Alom et al. 2019. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics.* **8**(3), 292. DOI: 10.3390/electronics8030292.

[114] R. Abbott et al. 2023. GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo during the Second Part of the Third Observing Run. *Phys. Rev. X.* **13**(4), 041039. DOI: 10.1103/PhysRevX.13.041039.

[115] J. Veitch et al. 2015. Parameter estimation for compact binaries with ground-based gravitational-wave observations using the LALInference software library. *Phys. Rev. D.* **91**(4), 042003. DOI: 10.1103/PhysRevD.91.042003.

[116] Gregory Ashton et al. 2019. Bilby: A User-friendly Bayesian Inference Library for Gravitational-wave Astronomy. *ApJS.* **241**(2). DOI: 10.3847/1538-4365/ab06fc.

[117] E. A. Huerta et al. 2019. Enabling real-time multi-messenger astrophysics discoveries with deep learning. *Nature Reviews Physics.* **1**, 600–608. DOI: 10.1038/s42254-019-0097-4.

[118] Maximilian Dax et al. 2021. Real-Time Gravitational Wave Science with Neural Posterior Estimation. *Phys. Rev. Lett.* **127**(24), 241103. DOI: 10.1103/PhysRevLett.127.241103.

[119] Cody Messick et al. 2017. Analysis framework for the prompt discovery of compact binary mergers in gravitational-wave data. *Phys. Rev. D.* **95**(4). DOI: 10.1103/PhysRevD.95.042001.

[120] T. Adams et al. 2016. Low-latency analysis pipeline for compact binary coalescences in the advanced gravitational wave detector era. *Class. Quantum Grav.* **33**(17), 175012. DOI: 10.1088/0264-9381/33/17/175012.

[121] Shaun Hooper et al. 2012. Summed parallel infinite impulse response filters for low-latency detection of chirping gravitational waves. *Phys. Rev. D.* **86**(2), 024012. DOI: 10.1103/PhysRevD.86.024012.

[122] Bruce Allen. 2005. χ^2 time-frequency discriminator for gravitational wave detection. *Phys. Rev. D.* **71**(6), 062001. DOI: 10.1103/PhysRevD.71.062001.

[123] Timothy Gebhard et al. 2019. Convolutional neural networks: A magic bullet for gravitational-wave detection? *Phys. Rev. D.* **100**(6), 063015. DOI: 10.1103/PhysRevD.100.063015.

[124] Wei Wei and E. A. Huerta. 2020. Gravitational wave denoising of binary black hole mergers with deep learning. *Physics Letters B.* **800**, 135081. DOI: 10.1016/j.physletb.2019.135081.

[125] Daniel George and E. A. Huerta. 2018. Deep neural networks to enable real-time multimessenger astrophysics. *Phys. Rev. D.* **97**, 044039. DOI: 10.1103/PhysRevD.97.044039.

[126] Y.-C. Lin and J.-H. P. Wu. Detection of gravitational waves using bayesian neural networks. *Physical Review D*, 103:063034, 2021.

[127] Marlin B. Schäfer et al. 2022. MLGWSC-1: The first Machine Learning Gravitational-Wave Search Mock Data Challenge. arXiv preprint: arXiv:2209.11146.

[128] Paraskevi Nousi et al. 2023. Deep residual networks for gravitational wave detection. *Phys. Rev. D.* **108**(2), 024022. DOI: 10.1103/PhysRevD.108.024022.

[129] Narenraju and Christopher Messenger. 2025. Identifying and mitigating machine-learning biases for the gravitational-wave detection problem. *Phys. Rev. D.* **112**(10), 103002. DOI: 10.1103/zwj9-ycyz.

[130] Michael Pürrer. 2014. Frequency-domain reduced order models for gravitational waves from aligned-spin compact binaries. *Class. Quantum Grav.* **31**, 195010. DOI: 10.1088/0264-9381/31/19/195010.

[131] Scott E. Field et al. 2011. Reduced Basis Catalogs for Gravitational Wave Templates. *Phys. Rev. Lett.* **106**(22), 221102. DOI: 10.1103/PhysRevLett.106.221102.

[132] Tyson B. Littenberg and Neil J. Cornish. 2015. Bayesian inference for spectral estimation of gravitational wave detector noise. *Phys. Rev. D.* **91**(8), 084034. DOI: 10.1103/PhysRevD.91.084034.

[133] Hunter Gabbard, Chris Messenger, Ik Siong Heng, Francesco Tonolini, and Roderick Murray-Smith. Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy. *Nature Physics*, 18:112–117, 2022.

[134] Marta Garnelo et al. 2018. Conditional Neural Processes. arXiv preprint: arXiv:1807.01613.

[135] B. P. Abbott et al. 2017. Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A. *ApJL.* **848**. DOI: 10.3847/2041-8213/aa920c.

[136] Carl W. Helstrom. *Statistical Theory of Signal Detection.* (Pergamon Press, 1968). DOI: 10.1016/C2013-0-02193-7.

[137] B. P. Abbott et al. 2016. Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914. *Class. Quantum Grav.* **33**(13), 134001. DOI: 10.1088/0264-9381/33/13/134001.

[138] Chad Hanna et al. 2020. Fast evaluation of multidetector consistency for real-time gravitational wave searches. *Phys. Rev. D.* **101**(2), 022003. DOI: 10.1103/PhysRevD.101.022003.

[139] Sushant Sharma Chaudhary et al. 2024. Low-latency gravitational wave alert products and their performance at the time of the fourth LIGO-Virgo-KAGRA observing run. arXiv preprint: arXiv:2308.04545.

[140] T. Adams. 2015. Low latency search for compact binary coalescences using MBTA. arXiv preprint: arXiv:1507.01787.

[141] Manoj Kovalam, Md Anwarul Kaium Patwary, Anala K. Sreekumar, Linqing Wen, Fiona H. Panther, and Qi Chu. Early warnings of binary neutron star coalescence using the spiir search. *The Astrophysical Journal Letters*, 927(1):L9, 2022.

[142] D. Davis et al. 2021. LIGO detector characterization in the second and third observing runs. *Class. Quantum Grav.* **38**(13), 135014. DOI: 10.1088/1361-6382/abfd85.

[143] Ian W. Harry et al. 2014. Investigating the effect of precession on searches for neutron-star–black-hole binaries with Advanced LIGO. *Phys. Rev. D.* **89**(2), 024010. DOI: 10.1103/PhysRevD.89.024010.

[144] C. Furtado et al. 2021. Strong gravitational lensing in a spacetime with topological charge within the Eddington-inspired Born-Infeld gravity. *Phys. Rev. D.* **103**(4), 044047. DOI: 10.1103/PhysRevD.103.044047.

[145] Grégory Baltus. 2021. Convolutional neural networks for the detection of the early inspiral of a gravitational-wave signal. *Phys. Rev. D.* **103**(10), 102003. DOI: 10.1103/PhysRevD.103.102003.

[146] Plamen G. Krastev. 2020. Real-Time Detection of Gravitational Waves from Binary Neutron Stars using Artificial Neural Networks. arXiv preprint: arXiv:1908.03151.

[147] Hang Yu, Rana X. Adhikari, Ryan Magee, Surabhi Sachdev, and Yanbei Chen. Early warning of coalescing neutron-star and neutron-star-black-hole binaries from the nonstationary noise background using neural networks. *Physical Review D*, 104:062004, 2021.

[148] Ana Martins et al. 2024. Improving Early Detection of Gravitational Waves from Binary Neutron Stars Using CNNs and FPGAs. arXiv preprint: arXiv:2409.05068v1.

[149] Jing Luan et al. 2011. Towards low-latency real-time detection of gravitational waves from compact binary coalescences in the era of advanced detectors. arXiv preprint: arXiv:1108.3174.

[150] Wouter van Straalen et al. 2024. Pre-Merger Detection and Characterization of Inspiraling Binary Neutron Stars Derived from Neural Posterior Estimation. arXiv preprint: arXiv:2407.10263.

[151] Tianyu Du et al. 2021. Cert-RNN: Towards Certifying the Robustness of Recurrent Neural Networks. *Association for Computing Machinery*. 516-534. DOI: 10.1145/3460120.3484538.

[152] Yong Yu et al. 2019. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *MIT Press*. **31**(7), 1235-1270. DOI: 10.1162/neco$_{a0}$1199.

[153] Alex Sherstinsky. 2023. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. arXiv preprint: arXiv:1808.03314.

[154] Kamilya Smagulova and Alex Pappachen James. 2020. Overview of Long Short-Term Memory Neural Networks. *Springer, Cham*. **14**. DOI: 10.1007/978-3-030-14524-8$_1$1.

[155] PyCBC. [Online]. Available: https://pycbc.org/ [Accessed: May 3, 2026].

[156] Bruce Allen et al. 2012. FindChirp: An Algorithm for Detection of Gravitational Waves from Inspiraling Compact Binaries. *Phys. Rev. D.* **85**(12), 122006. DOI: 10.1103/PhysRevD.85.122006.

[157] Sascha Husa et al. 2016. Frequency-domain gravitational waves from nonprecessing black-hole binaries. I. New numerical waveforms and anatomy of the signal. *Phys. Rev. D.* **93**(4), 044006. DOI: 10.1103/PhysRevD.93.044006.

[158] LIGO Document T0900288-v3. [Online]. Available: https://dcc.ligo.org/LIGO-T0900288/public [Accessed: May 3, 2026].

[159] Dalwinder Singh and Birmohan Singh. 2020. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*. **97**. DOI: 10.1016/j.asoc.2019.105524.

[160] Pytorch. [Online]. Available: https://docs.pytorch.org/docs/2.11/generated/torch.nn.BCEWithLogitsLoss.html [Accessed: May 3, 2026].

[161] Marica Branchesi, LIGO Scientific Collaboration and Virgo Collaboration. 2012. Electromagnetic follow-up of gravitational wave transient signal candidates. *J. Phys.: Conf. Ser.* **375**(1), 062004. DOI: 10.1088/1742-6596/375/1/062004.

[162] Tito Dal Canton and Ian W. Harry. 2017. Designing a template bank to observe compact binary coalescences in Advanced LIGO's second observing run. arXiv preprint: arXiv:1705.01845.

[163] B. P. Abbott et al. 2017. Calibration of the Advanced LIGO detectors for the discovery of the binary black-hole merger GW150914. *Phys. Rev. D.* **95**(6), 062003. DOI: 10.1103/PhysRevD.95.062003.

[164] Pauli Virtanen et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*. **17**, 261–272. DOI: 10.1038/s41592-019-0686-2.

[165] William H. Press et al. *Numerical Recipes: The Art of Scientific Computing*. (Cambridge University Press, 2007). DOI: 10.5555/1403886.

[166] Frank J. Massey Jr. 1951. The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*. **46**(253), 68–78. DOI: 10.2307/2280095.