Transient Classification in low SNR Gravitational Wave data using Deep Learning

Rahul Nigam, Amit Mishra, and Pranath Reddy
Department of Physics, Birla Institute of Technology & Science Pilani - Hyderabad Campus, Hyderabad, India

The recent advances in Gravitational-wave astronomy have greatly accelerated the study of Multi-messenger astrophysics. There is a need for the development of fast and efficient algorithms to detect non-astrophysical transients and noises due to the rate and scale at which the data is being provided by LIGO and other gravitational wave observatories. These transients and noises can interfere with the study of gravitational waves and binary mergers and induce false positives. Here, we propose the use of deep learning algorithms to detect and classify these transient signals. Traditional statistical methods are not well designed for dealing with temporal signals but supervised deep learning techniques such as RNN-LSTM and deep CNN have proven to be effective for solving problems such as time-series forecasting and time-series classification. We also use unsupervised models such as Total variation, Principal Component Analysis, Support Vector Machine, Wavelet decomposition or Random Forests for feature extraction and noise reduction and then study the results obtained by RNN-LSTM and deep CNN for classifying the transients in low-SNR signals. We compare the results obtained by the combination of various unsupervised models and supervised models. This method can be extended to real-time detection of transients and merger signals using deep-learning optimized GPU’s for early prediction and study of various astronomical events. We will also explore and compare other machine learning models such as MLP, Stacked Autoencoder, Random forests, extreme learning machine, Support Vector machine and logistic regression classifier.

I. INTRODUCTION

Detecting gravitational waves is a huge challenge due to the scales at which these events are experienced by us. LIGO detectors detect space compression as small as 10-21m and to do this the instruments need to be very accurate. However, noise and signals due to non-astrophysical sources can creep into the readings giving false positives. Hence, classification of different signals is crucial in case of detecting gravitational waves.

In recent times, machine learning and neural networks has proven to be an extremely helpful tool for analyzing large amounts of raw data and infer some practical results from it. Here we use neural networks to classify a transient dataset with low SNR (signal-to-noise ratio) of gravitational waves to better detect them.

II. METHODOLOGY

The dataset used for training our transient detection model is created using eight different classes of transients, namely, Sine Gaussian, Ring Down, Gaussian, Supernova, Cusp, Black Hole Merger, Chirping Sine Gaussian and Blip. The transients are then whitened to reflect the actual LIGO data and also to reduce the effect of low frequency noise in the dataset. We will be working with low SNR signals with the value ranging from 5 to 25. Below we list the different signals we would be working with.

A. Gaussian(GN)

Few non-astrophysical signals are simply modelled as a Gaussian with \( \tau \) values of 0.0005, 0.001, 0.0025, 0.005, 0.0075, 0.01, 0.02 and 0.05.

\[
s(t) = e^{-\frac{(t-t_0)^2}{\tau^2}}
\]

B. Sine-Gaussian(SG)

These model non-astrophysical glitches that are significant in the analysis of coalescing compact binaries. \( \tau \) is set to \( 2/f_0 \) where \( f_0 \) is the central frequency being varied logarithmically from 100Hz to 2000Hz.

\[
s(t) = e^{-\frac{(t-t_0)^2}{\tau^2}} \sin(2\pi f_0(t - t_0))
\]

C. Ringdown(RG)

These signals can be characterized by their shorter bandwidth but longer duration and are basically sinusoids which are damped. They originate from the quasi-normal modes of a final black hole that has been formed due to coalescing compact binaries. In this case, we set \( \tau = 4/f_0 \) with \( f_0 \) being similar to the value used for Sine-Gaussian data set.

\[
s(t) = \begin{cases} 
  e^{-\frac{(t-t_0)^2}{\tau^2}} \cos(2\pi f_0(t - t_0)) & \text{if } t \geq t_0 \\
  0 & \text{if } t < t_0
\end{cases}
\]

D. Chirping Sine Gaussian (CSG)

This signal is similar to sine gaussian, CSG has an additional factor for chirping. These accurately model the white glitches that are frequently observed in LIGO.
data. The parameter are varied as follows: \( f_o \): \{5,100\}, \( \alpha \): \{10,100\} and \( \tau \): \{0.001,0.025\}.

\[
s(t) = \frac{\exp\left(-\frac{\left(1-i\alpha\right)(t-t_o)^2}{4\tau^2} + 2\pi i(t-t_o)f_o\right)}{(2\pi\tau^2)^{1/4}}
\] (4)

E. Supernova (SN)

One of supernova waveforms namely, the Zwerger-Mueller waveforms are generated by axi-symmetric core collapse of supernovae. These are simulated by hydrodynamical simulations of stellar core collapse by varying the initial conditions like adiabatic index, spin, and differential rotation profile. 78 models are included which consists of a simple analytic equation of state.

F. Cusp(CSP)

Cosmic strings are generated due to symmetry breaking phase transitions in the early universe, they are modelled as the following cusp like signals:

\[
h(f) = A(f)f^{-4/3}
\] (5)

These are generated with an exponential roll off after a set value of cut-off frequency \( f_o \) varied between 50 Hz and 2000Hz.

G. Black Hole Merger (LBM)

Using the Lazarus approach, these waveforms capture the coalescence radiation emitted from a merger of binary black-hole systems. To construct the time domain which replicates the merger scenarios, an analytical approximation is used. This is done considering the black hole binaries have a chirp mass in the range \{20,50\} and cos of inclination angle varied between zero and one.

H. Blip (Blip)

Although frequently observed in LIGO detectors, the origin of blip signals are not well understood. Here, we simulate these by clipping sine-gaussian at a small percentage level around mean amplitude.

Below we present some generated signals of the different types listed above which will further be used to train our model. In total, we plan to generate 48,000 (6000 in each class) transient signals for training and testing.
FIG. 1: Sample signals of the eight different types of transients. In order to find out whether there is noise present in the data, we use residual, i.e., we subtract the model data from experimental data and find out if this residual can fit the properties to a certain type of noise.

III. ANALYSIS

In this project, we will be exploring the use of deep learning in classifying the various transient signals mentioned in the above section. Later, we will be using an unsupervised model such as Total variation, Principal Component Analysis, Support Vector Machine, Wavelet decomposition or Random Forests on our generated dataset for feature extraction which will denoise the dataset and help our supervised model in better classifying the inputs.

We will be training a CNN (Convolutional Neural Network) model and a Bidirectional LSTM-RNN (An im-
plementation of Recurrent Neural Network using Long short-term memory (LSTM) modules model for time-series classification. We will be comparing the results of both models using standard metrics like Accuracy. Recurrent neural networks are widely used in various machine learning problems such as speech recognition, financial data analysis, time series forecasting, and time series classification. The main feature that makes RNN highly versatile when working with time series data is their ability to learn sequences in the data. CNN is primarily used for image and video data but we can use a deep CNN model for classifying the transients due to its ability to learn non-linear hierarchical structures and lower computational costs. We will also explore the results obtained by the Deep Filtering algorithm proposed by Daniel George and E. A. Huerta. Apart from these, we will also be looking at other machine learning and deep learning algorithms such as stacked autoencoder, multilayer perceptron, random forests, logistic regression classifier, Support Vector Machine (SVM), and extreme learning machine (ELM).

The architectures of the Deep Filtering algorithm, RNN, CNN, MLP, and stacked autoencoder have been presented in the figures below. For random forests, we have used 100 trees with the depth of each tree being 5. The ELM has been applied with 8000 hidden neurons and a hyperbolic tangent activation function. For implementing SVM, we have used LinearSVC (Linear Support Vector Classification) which uses one-vs-the-rest multiclass strategy. For training the gradient descent based algorithms we have used ADAM optimizer with a learning rate and decay of $10^{-5}$. CNN, RNN, and Deep Filtering algorithm have been trained for 50 epochs with a batch size of 128 whereas the MLP and stacked autoencoder have been trained for 100 epochs with a batch size of 128. We have summarized the accuracy of trained models and the computational time taken for training the models on both a CPU (i7 8750h) and a GPU (Tesla K80) respectively. We have used hold-out cross-validation for testing the performance of the models with a train-test split of 0.8-0.2.
IV. RESULTS

In this section, we will be presenting the predictions of each model in the form of confusion matrices and also the accuracy values and the training times of each model in tables below.

FIG. 6: Architecture of Deep Filtering Model

FIG. 7: Prediction of CNN Model

FIG. 8: Prediction of Deep Filtering Model

FIG. 9: Prediction of RNN Model

FIG. 10: Prediction of MLP Model
TABLE I: Index corresponding to each transient

| Index | Transient                  |
|-------|---------------------------|
| 0     | Ring Gaussian             |
| 1     | Sine Gaussian             |
| 2     | Gaussian                  |
| 3     | Chirping Sine Gaussian    |
| 4     | Cusp                      |
| 5     | Binary Black Hole Merger  |
| 6     | Blip                      |
| 7     | Supernova                 |
TABLE II: Accuracy of each model based on predictions from test data

| Model              | Accuracy |
|--------------------|----------|
| Deep Filtering     | 100.00000|
| CNN                | 100.00000|
| RNN                | 100.00000|
| Random Forest      | 97.75100 |
| MLP                | 96.62500 |
| SVM                | 91.07291 |
| Logistic Regression| 89.66666 |
| Stacked Autoencoder| 86.37500 |
| ELM                | 67.15625 |

TABLE III: Computational time for training each model (in seconds)

| Model              | CPU  | GPU  |
|--------------------|------|------|
| Deep Filtering     | 73700| 4800 |
| CNN                | 4050 | 300  |
| RNN                | 2200 | 200  |
| Stacked Autoencoder| 650  | 520  |
| MLP                | 600  | 300  |
| SVM                | 1312 | -    |
| ELM                | 379  | -    |
| Random Forest      | 86   | -    |
| Logistic Regression| 44   | -    |

V. CONCLUSION

We have explored various machine learning and deep learning algorithms for the classification of transient signals. The convolution and lstm based models, Deep Filtering, CNN and RNN were able to achieve an accuracy of 100 percent over the test dataset. Compared to Deep Filtering and CNN, the RNN model was able to get an accuracy of 100 percent in a much lower training time which makes it ideal for implementation on low powered devices for real-time transient detection and classification. On the other hand, models such as random forests and logistic regression were able to achieve a decent accuracy rate in an extremely low training time which makes them ideal for fast inference to get an insight over the gravitational wave data that is available. We plan to continue this study by introducing more types of transient waves and also add other types of low-frequency noises to the transients apart from white noise. We will also compare deep filtering algorithm with RNN for classifying low SNR binary merger signals and white noise, which is an important task while analysing highly noisy time series data streams. Our next objective is to compute residual and compare with different noise signatures. We will also be working on an unsupervised deep learning algorithm to learn the representations of merger signals and classify them from noise samples.
[1] LIGO Scientific Collaboration, Advanced LIGO, Classical and Quantum Gravity 32, 074001 (2015), arXiv:1411.4547 [gr-qc]
[2] LIGO Scientific Collaboration and Virgo Collaboration, GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral, Phys. Rev. Lett. 119, 161101 (2017), arXiv:1710.05832 [gr-qc]
[3] D. George and E. A. Huerta, Deep neural networks to enable real-time multimessenger astrophysics, Phys. Rev. D 97, 044039 (2018), arXiv:1701.00008 [astro-ph.IM]
[4] D. George and E. A. Huerta, Deep Learning for real-time gravitational wave detection and parameter estimation: Results with Advanced LIGO data, Physics Letters B 778, 64 (2018), arXiv:1711.03121 [gr-qc]
[5] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning (MIT Press, 2016) http://www.deeplearningbook.org
[6] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature 521, 436 (2015)
[7] H. Wang and B. Raj, On the Origin of Deep Learning, arXiv e-prints , arXiv:1702.07800 (2017), arXiv:1702.07800 [cs.LG]
[8] D. E. Rumelhart and J. L. McClelland, Learning internal representations by error propagation, in Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations (MITP, 1987) pp. 318–362
[9] H. Shen, D. George, E. A. Huerta, and Z. Zhao, Denoising Gravitational Waves using Deep Learning with Recurrent Denoising Autoencoders, arXiv e-prints , arXiv:1711.09919 (2017), arXiv:1711.09919 [gr-qc]
[10] F. Karim, S. Majumdar, H. Darabi, and S. Harford, Multivariate LSTM-FCNs for Time Series Classification, arXiv e-prints , arXiv:1801.04503 (2018), arXiv:1801.04503 [cs.LG]
[11] K. Antczak, Deep Recurrent Neural Networks for ECG Signal Denoising, arXiv e-prints , arXiv:1807.11551 (2018), arXiv:1807.11551 [cs.NE]
[12] M. Razzano and E. Cuoco, Image-based deep learning for classification of noise transients in gravitational wave detectors, Classical and Quantum Gravity 35, 095016 (2018)
[13] H. Shen, E. A. Huerta, Z. Zhao, E. Jennings, and H. Sharma, Deterministic and Bayesian Neural Networks for Low-latency Gravitational Wave Parameter Estimation of Binary Black Hole Mergers, arXiv e-prints , arXiv:1903.01998 (2019), arXiv:1903.01998 [gr-qc]
[14] B. Rasti, M. O. Ulfarsson, and J. R. Sveinsson, Hyperspectral feature extraction using total variation component analysis, IEEE Transactions on Geoscience and Remote Sensing 54, 6976 (2016)
[15] W. Wei and E. A. Huerta, Gravitational wave denoising of binary black hole mergers with deep learning, Physics Letters B 800, 135081 (2020), arXiv:1901.00869 [gr-qc]
[16] M. Makkie, H. Huang, Y. Zhao, A. V. Vasilakos, and T. Liu, Fast and scalable distributed deep convolutional autoencoder for fmi big data analytics, Neurocomputing 325, 20 (2019)
[17] S. Alhagry, A. Aly, and R. El-Khoribi, Emotion recognition based on eeg using lstm recurrent neural network, International Journal of Advanced Computer Science and Applications 8 (2017)