Gravitational Wave Detection with Novel Machine Learning Models

Ian Whitehouse & Dr. Roberto Corizzo

American University, Department of Computer Science

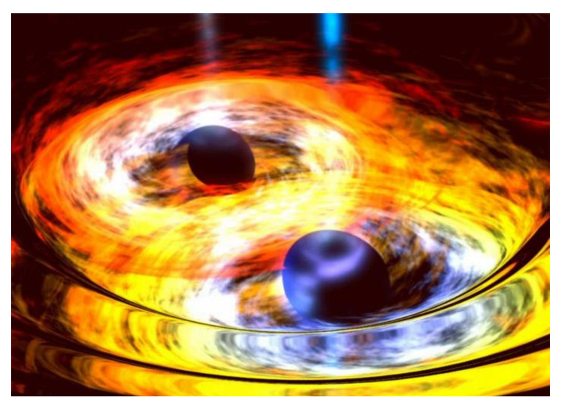
Introduction

Gravitational waves (GW), first observed by the Advanced Laser Interferometer Gravitational-Wave Observatory (LIGO) in 2015, are an important subject of study in astrophysics, paving the way for a better understanding of the universe.

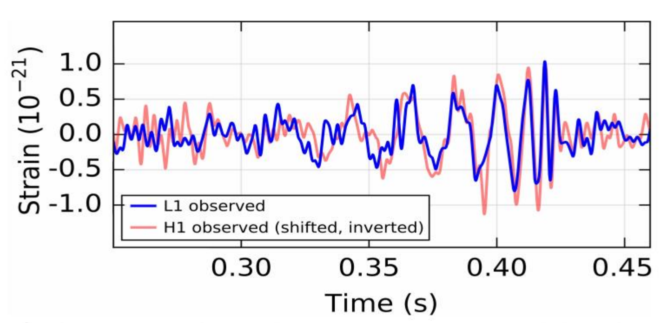
Gravitational waves are rarely observed and do not fit into predictable patterns, whereas noisy time series are abundant. As a result, machine learning-based anomaly detection techniques can significantly support researchers in the analysis of LIGO and Virgo telescope waveforms.

This research applies a novel model, the adversarial autoencoder (AAE), to gravitational wave detection, and compares the AAEs' results to a traditional autoencoder (AE)s' results. We hypothesize that, because of the adversarial nature of AAEs, adversarial autoencoders will be better anomaly detectors than traditional autoencoders.

Hanford (WA) GW interferometer



Example of GW phenomenon (colliding black holes)



Example of time series data generated by interferometer

Experiment Setup

We performed experiments using 3 datasets, all of which included four gravitational wave discoveries (GW150914, LVT151012, GW151226, and GW170104) (Abbott et al).

Data was collected at a sampling rate of 4096hz and included time series with lengths of both 1 second and 3 seconds.

- **GW1**: Anomalous timeseries from these events, and simulated noise from the PyCBC library (Nitz et al).
- **GW2**: Anomalous samples blended with synthetic noise. The noise was added with a weight of 0.1, 0.25 and 0.5.
- **GW3**: Same anomalous samples as GW1, and noise sampled in the proximity of the events, which provides a more realistic test

To test the models, approximately 180 trials were run, with hyperparameters randomly sampled. The distribution of hyperparameters were selected to match the previous research and are described in our methodology.

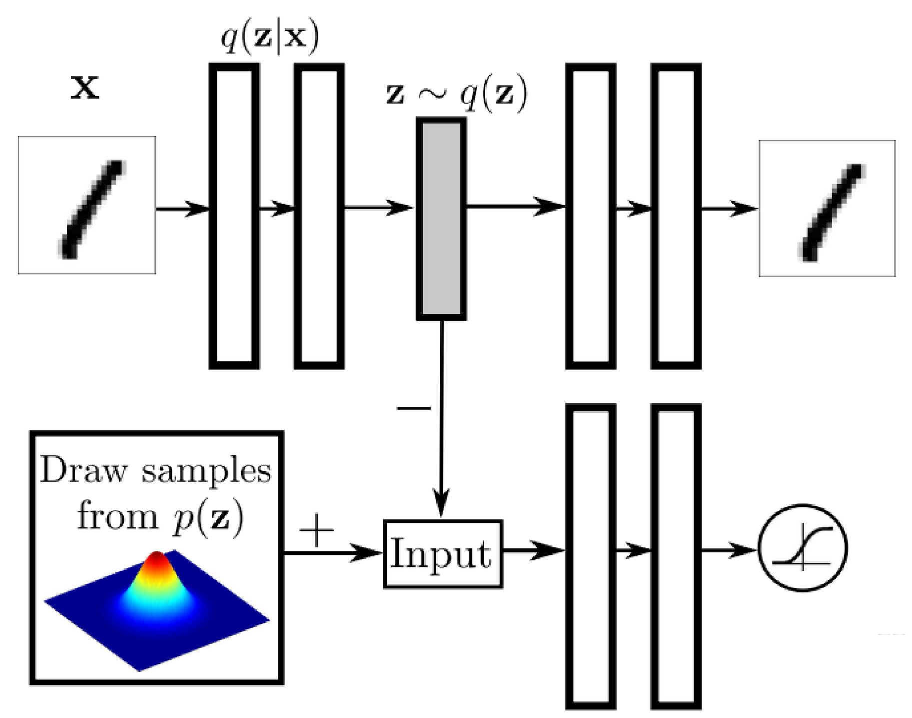
The experiments were run using TensorFlow, on a computer with a RTX 3090 and 4090.

Methodology

Autoencoders are machine learning models that consist of an encoding stage and a decoding stage. During training, the model learns how to compress the data into a meaningful representation in a lower-dimensional space (latent layer).

The model's loss, therefore, is the difference between the decoding of the lower-dimensional space and the original input. Autoencoders are often used for anomaly detection because, when trained on only non-anomalous samples, the autoencoder is unable to decompress anomalous samples without incurring high reconstruction error.

Adversarial autoencoders (Makhzani et al) are an extension of autoencoders that include an additional discriminator. In a traditional autoencoder, the latent layer is not continuous, however, by including the discriminator and using the autoencoder's compression stage as a generator, the latent layer is made continuous and the model can be used to generate synthetic data, like in a general adversarial network.



Inclusion of normal distribution in an AAE's training by adding a discriminator to the model

During the experiments, pairs of autoencoders and adversarial autoencoders were created by randomly sampling hyperparameters from the table below

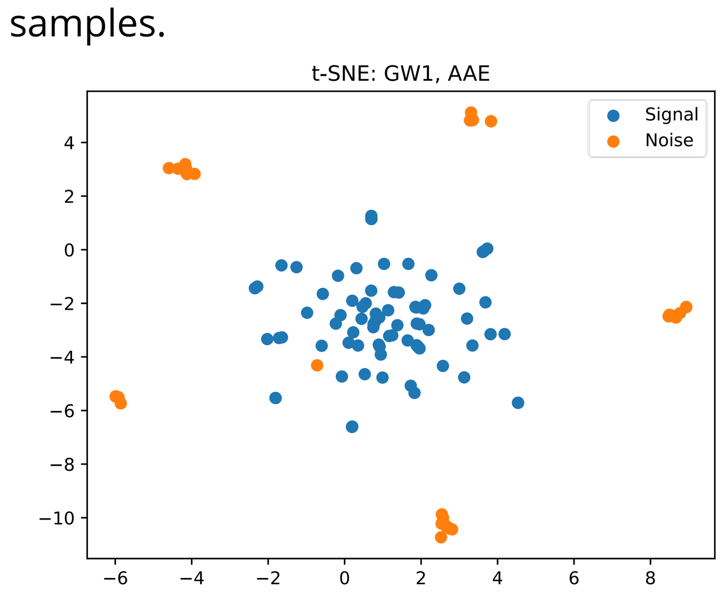
Hyper Parameters	Possible Values
Time Sequence Length	1, 3
Input Layer Neurons	4096 * time sequence length
Hidden Layers	1, 2
Hidden Layer Neurons	1/2, 1/4 or 1/8 of the input layer This term is squared for models with 2 hidden layers
Latent Layer Neurons	1/4, 1/8, 1/16 or 1/32 of the input layer
Dropout	10%, 25% or 50%
Learning Rate	10^-7, 10^-6 or 10^-5
Batch Size	4, 8 or 16

Each model was trained solely on noise (one-class learning), before being fed testing data that included both noise and anomalous signals (GW). The model attempted to classify these anomalous signals using its latent representation and the model's reconstruction accuracy.

To classify the signals using the model's latent representation, we employed simple ML classifiers optimized via grid search, shown in the table below

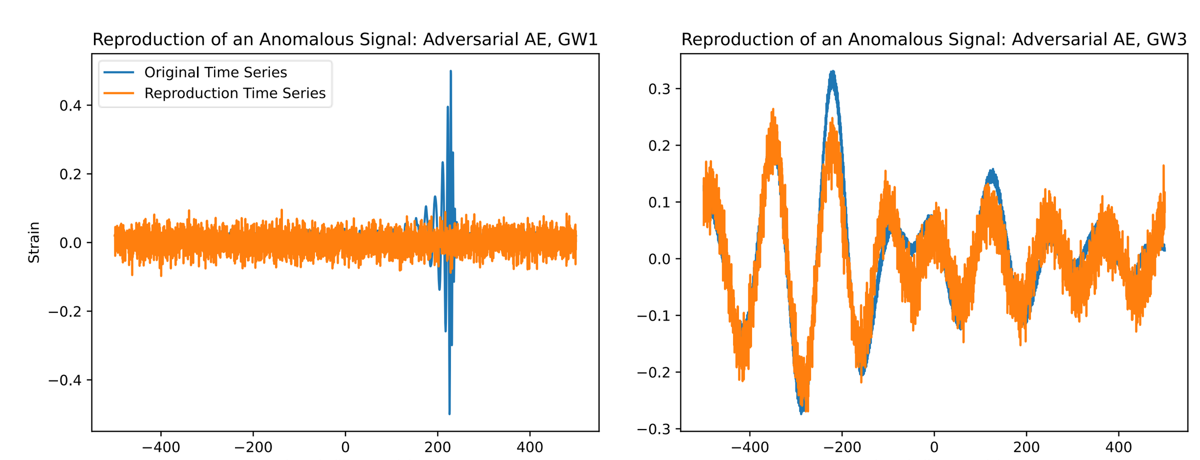
Latent Layer Classifiers	Hyper Parameters	Possible Values
Logistic Regression	Max Iterations	100
Extra Trees	Number of Estimators	10, 100 or 1000
Random Forest	Number of Estimators	10, 100 or 1000
Gradient Boosted Trees	Max Depth	5
	Minimum Samples per Leaf	1
XGBoost	Number of Estimators	1000
Support Vector Machine	Kernel	RBF
	C	1, 10 or 100
	Gamma	1^-4, 1^-3, 1^-2 or 0.1
K Nearest Neighbors	Algorithm	KD Tree
	Leaf Size	1, 6, 11, 16, 21, 26, 31, 36, 41 or 46
	Number of Neighbors	1, 6, 11, 16, 21, 26 or 31

The classifiers were trained on latent layers of both noise and anomalous samples, making them supervised learners. In line with our hypothesis, we believed that, since the AE and AAE were only trained on noise, it would be trivial to separate out anomalous samples.



Latent representations of GW1 signals after compression by an AAE, highlighting clusters formed by anomalous data

The model's reconstruction accuracy is an unsupervised anomaly detection method that was the focus of Dr. Corizzo's earlier work. In these experiments, we also tested whether reconstructions of the anomalous signals would be less accurate when applying an adversarial autoencoder.

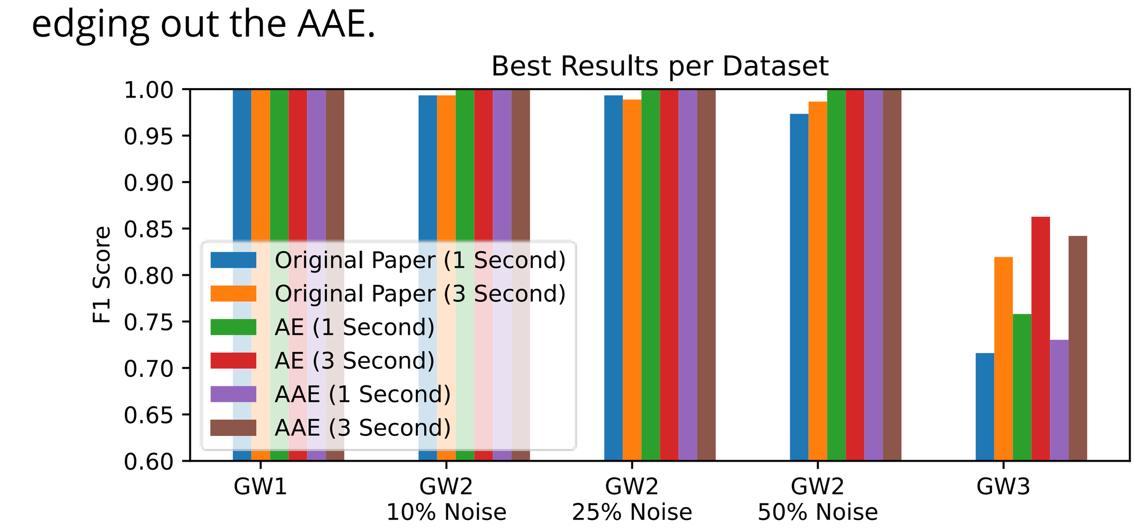


GW1: AE and AAE were unable to replicated anomalous signals, despite easily replicating noise (Left).

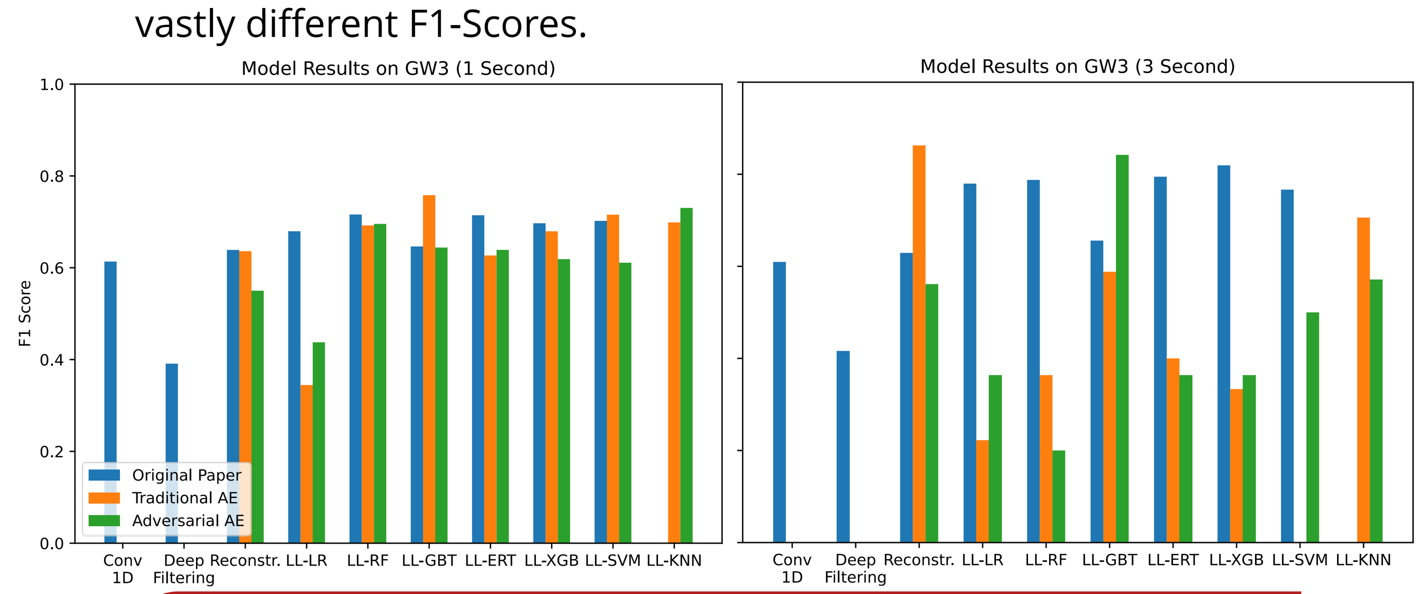
GW3: AAE easily replicates the anomalous signal, despite being trained only on noise (Right)

Results

In our experiments, the both the AE and AAE models moderately improved on the original paper's F1-Scores, with the AE model edging out the AAE.



On the most difficult problem, GW3, the AE and AAE improved on the original paper, but different latent layer (LL) classifiers had vastly different F1-Scores.



Conclusion

These results represents an improvement on prior results. However, additional research is necessary to further improve the accuracy of the models and assess them under challenging conditions and experimental settings.

One hypothesis for why AAEs did not perform better as anomaly detectors is that the models tested were so advanced that they could easily reconstruct the anomalies, leading to them having higher reconstruction accuracies, as shown in the results for GW3. This is supported by the paper "Do Deep Generative Models Know What They Don't Know?" (Nalisnick et al), which posits that generative models, like AAE's, can confidently parse inputs that they were not trained on, which means that may not be usable as anomaly detectors. In our own models, the AAE appears to easily reconstruct the anomalies, as shown in the figure to the left

We plan to further analyze these results as we begin to compile our findings

Citations and Acknowledgements

Abbott, B. P., et al. "GWTC-1: a gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs." Physical Review X 9.3 (2019): 031040.

Corizzo, R, et al. "Scalable auto-encoders for gravitational waves detection from time series data." Expert Systems with Applications 151 (2020): 113378.

Makhzani, Alireza, et al. "Adversarial autoencoders." arXiv preprint arXiv:1511.05644 (2015).

Nalisnick, Eric, et al. "Do deep generative models know what they don't know?." arXiv preprint arXiv:1810.09136 (2018).

Nitz, Alex, et al. Gwastro/Pycbc: V2.1.1 Release of PyCBC. v.2.1.1, Zenodo, 2023, doi:10.5281/ZENODO.7746324.