University of Massachusetts Dartmouth

Department of Mathematics

An Exploration of Statistical Methods in Determining the Gaussianity of LIGO Detector Data

A Data Science project
by

Ashish Thomas Mathew
(02101455)

Project Advisors:

Dr. Sarah Caudill
University of Massachusetts Dartmouth

Dr. Melissa Lopez
National Institute for Subatomic Physics (Nikhef)

Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

May, 2025

Abstract

The Advanced Laser Interferometer Gravitational-wave Observatory (LIGO) is one of the most sophisticated instruments ever built, capable of measuring motion 10,000 times smaller than a proton. It was the first device to detect a gravitational wave (GW) event. While analyzing the GW data from LIGO, the detector noise is approximated to be stationary and Gaussian. However, due to its high sensitivity, a common issue it faces is its susceptibility to glitches: transient, non-Gaussian noise bursts that contaminate the time series data obtained from the detector. This project aims to develop a tool to test if the distribution of the amplitudes of the detector noise is Gaussian or not as a way to assess whether it is clean up to a defined threshold, hence improving the quality of the analysis performed. Taking the null hypothesis that the distribution is Gaussian, we study statistical approaches of normality testing, namely the Shapiro-Wilk test, the Kolmogorov-Smirnov test, and the Anderson-Darling test, to test this null hypothesis against the alternative of the distribution being non-Gaussian. We found that of the three tests, the Shapiro-Wilk test performed the best in detecting non-Gaussian samples with an accuracy of 73.86% and a precision of 91.67%. In comparison, the Kolmogorov-Smirnov test had an accuracy of 64.43% and a precision of 100%, while the Anderson-Darling test had an accuracy of 63.9% and a precision of 100% when detecting non-Gaussianity.

1 Introduction

Nearly a century after Einstein's prediction of the existence of gravitational waves (GWs) in 1916, the first direct gravitational wave detection was achieved by the Advanced Laser Interferometer Gravitational-Wave Observatory (aLIGO) [1] and Virgo [2] interferometers during the binary black hole merger event GW150914 on September 14, 2015. The LIGO Livingston (L1) interferometer underwent several upgrades following this run - effectively a massive redesign - enhancing its sensitivity by 15% to 25% [3]. This improvement was evident during the O2 observing run during which L1, in conjunction with its Hanford (H1) counterpart and Virgo (V1), detected eleven new gravitational wave signals [4]. Subsequently, during the O3 run, all three detectors operated at their best possible sensitivity, leading to the first single-detector GW detection, GW190425, achieved by LIGO Livingston [5]. There have been over 90 GW events recorded at a high level of confidence since the LIGO-Virgo collaboration's inception to date [6].

A consequence of these detectors' sensitivities is their susceptibility to noise. During observation runs, various noise sources, such as seismic noise, suspension thermal noise, and sensing noise, affect the data collected by the interferometers. Among these, the most problematic are **glitches**: transient events caused by non-astrophysical phenomena such as anthropogenic noise, weather conditions, or instrument malfunctions [7, 8]. Glitches manifest as localized bursts of excess power in interferometer time series data and often do not have well-defined sources. They can occur at energy levels and frequencies that overlap with GW signals, thereby mimicking them and increasing the number of false positive detections. During the first half of the third observing run (O3a) H1 and L1 recorded glitches rates of 0.29 min^{-1} and 1.10 min^{-1} respectively, which rose to 0.32 min^{-1} and 1.17 min^{-1} during the second half (O3b) [6]. A notable example of glitches posing such an issue was during the binary neutron star merger event GW170817 [9], during which instrumental noise transients were detected before the event's coalescence time, complicating its detection and subsequent analysis.

This project aims to develop a tool to test if the amplitudes of detector noise in a sample of data from the detector contains a glitch through tests of **Gaussianity**. Gaussianity is the property of a sample distribution of data being normally distributed, also known as a Gaussian distribution. A normal/Gaussian distribution is a continuous probability distribution for a random variable where most of the data points are located around the mean, with lesser amounts of data further away. The Gaussian distribution is symmetric about the mean μ, which is equal to zero, with the standard deviation σ being equal to 1. This distribution is important as it can approximate many real world phenomena, such as the distribution of heights and weights of a country's population. It is also the basis for many statistical tests and methods. Any variation from this distribution is considered non-Gaussian. Figure 1 shows an example of a Gaussian distribution.

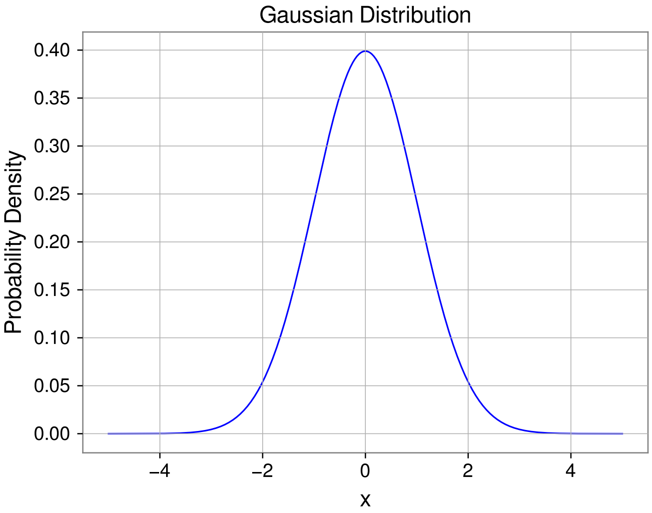
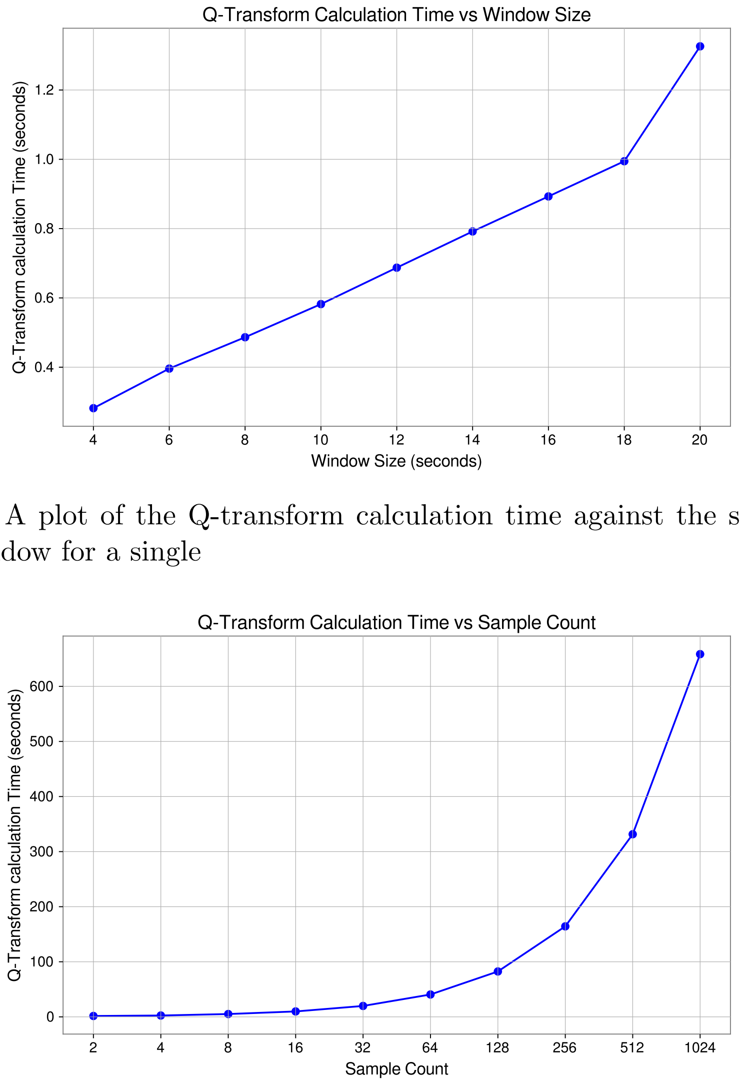


Figure 1: Example of a Gaussian distribution

All the noise sources in a detector collectively produce time series signals which can be treated as stochastic processes with their corresponding joint probability distributions and statistical properties [7]. In gravitational wave data analysis the noise floor of the detectors is approximated to be stationary and Gaussian. In the event of a glitch, the noise has some structure with a high signal-to-noise ratio (SNR), making it *non-Gaussian*.

This study uses three tests of Gaussianity, namely, (1) the Shapiro-Wilk test, (2) the Kolmogorov-Smirnov test, and (3) the Anderson-Darling test on preprocessed samples of the L1 interferometer data to determine their normality and assess how well these tests differentiate between Gaussian and non-Gaussian samples. This experiment also includes studying the distributions, durations, and frequency ranges, with particular emphasis on failure points for each statistical test across various glitch types. Furthermore, considering the frequency bands at which each glitch type occurs, this study also examines how each statistical test performs under band-pass filtering at various frequency ranges.

Another aspect of this project is the exploration of statistical hypothesis testing on time-amplitude domain data as a faster alternative to the current glitch detection methods working in the time-frequency domain. Detecting and mitigating the effects of glitches in interferometric strain data remains an active area of research within astrophysical data analysis, with several techniques proposed and implemented for the same [10–12]. However, many of these are computationally expensive, slowing down the process of assessing of the signals for glitch activity. The more popular solutions incorporate the *Q-transform* [13, 14], a modification of the standard Fourier transform where the analysis window scales inversely with frequency. The Q-transform is used as inputs to statistical tests to obtain a p-value depicting the statistical significance of excess power in the data. Assuming the background data to be Gaussian, the excess power that does not follow a Gaussian distribution suggests the presence of a glitch. While this method is effective for post-detection run analysis, it is much slower when scaled up to work on multiple samples. Figures 2a and 2b illustrate the time taken to compute the Q-transform while varying the sample window size and the number of samples respectively.



(a) A plot of the Q-transform calculation time against the sample window for a single

(b) Interference pattern

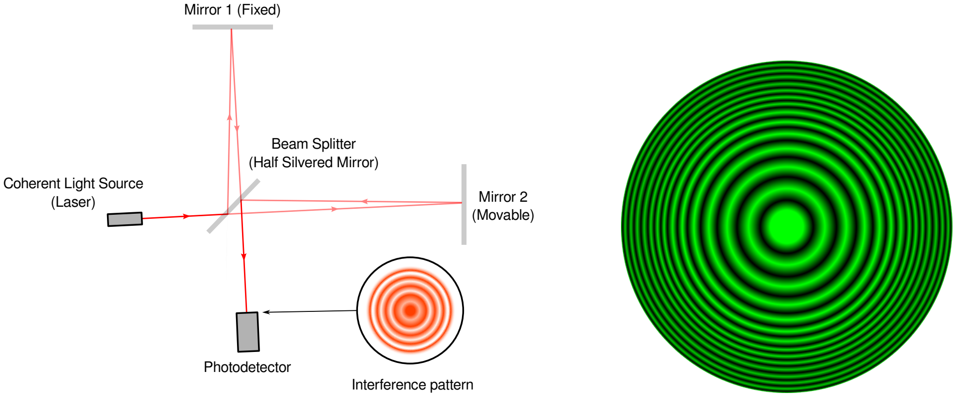
Figure 2: Plots showing how the time taken to compute the Q-transform scales with the sample window size and the number of samples.

This report is organized as follows: Section 2 discusses how interferometer data is obtained and describes the properties of the time series data used in this study. It also outlines the process of acquiring clean and glitched samples and the preprocessing applied. Section 3 describes in detail the statistical tests of Gaussianity used to assess the sample data. The experimental procedures and results are presented in Sections 4.1 and 4.2. Finally, Section 6 summarizes the findings of our experiments, discussing the strengths and limitations of the methods.

2 Data Acquisition and Conditioning

The Advanced LIGO and Virgo interferometers are large-scale, heavily modified versions of the Michelson interferometer (Figure 3a), originally invented by American physicist Albert A. Michelson in 1887 [6, 15]. LIGO operates in a vacuum, using a laser beam of light split into two orthogonal parts with the help of a half-silvered mirror mounted on horizontal seismometer suspensions. The orthogonally split beams are then sent through two arms of the interferometer, known as Fabry-Pérot cavities. Each of these arms is 4 kilometers long, consisting of mirrors at each end. The first of the mirrors is fixed in place, while the second one is movable with the help of precise micrometer drives. The laser beams, when reflected by the mirrors on either end of the arms, are combined again at the beam splitter and sent to photoelectric detectors.

The advanced LIGO interferometer is designed such that if the beams travel equal distances, the recombination would lead to a destructive interference pattern, resulting in no light coming out of the instrument [16]. When a disturbance of either astrophysical or terrestrial origin is detected, an infinitesimally small change occurs in lengths of the detector arms, in the order of $10^{-8}m$ [17]. One of the arms is stretched, and the other arm is compressed in the perpendicular direction, altering the lengths of the reflected and combined beams, misaligning them and resulting in an interference pattern (Figure 3b). This interference pattern, coupled with results from several other sensors, provides detailed information on the *strain* of the GW or glitch event being observed.



(a) Construction of an Interferometer

(b) Interference pattern

Figure 3: Illustrations of a basic Michelson interferometer [18, 19] and what an interference pattern would look like.

The GWpy Python package is widely used in this project. It provides a suite of tools to access and condition detector strain data from the Gravitational-Wave Open Science Centre (GWOSC) database [20]. This data is in the time domain, as timestamps in the GPS time system at nanosecond precision, and records the amplitude of the calibrated noise event as a differential change in lengths of the interferometer arms. GWpy handles detector data using the `TimeSeries` object, which is built upon `numpy` arrays. This allows compatibility with most of the core numpy utilities along with custom functions for signal processing, tabular data filtering, and visualization.

We use the strain data from L1 during O3a for this project due to the high rate of occurrence of glitches. Figure 4 shows the steps taken to acquire signal data from the detector and condition it for statistical testing.

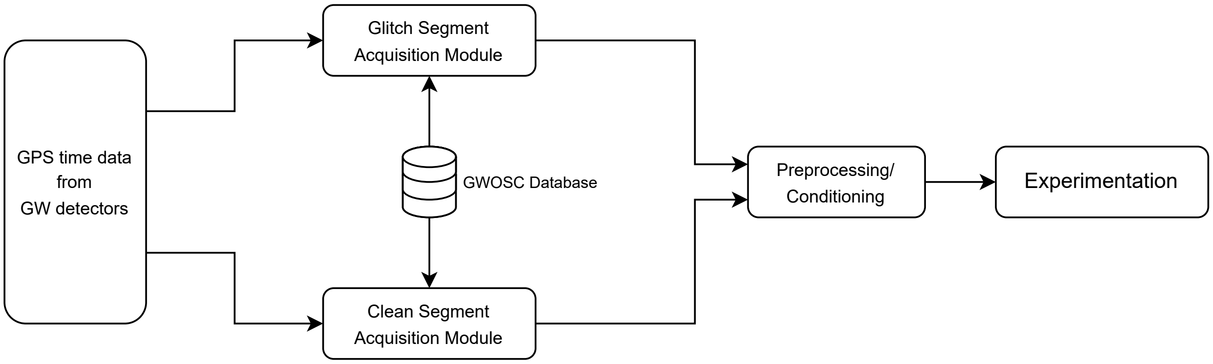


Figure 4: The Data Acquisition and Conditioning Pipeline

There are two main types of detector samples that we receive from the detectors: **Glitched detector data** and **Clean detector data** with separate modules used to obtain them.

2.1 The Glitch Segment Acquisition Module

The **Glitch Segment Acquisition Module** uses GPS times of glitch occurrences obtained from *Gravity Spy* [21] to fetch our glitch samples from the GWOSC database. GravitySpy is a large-scale citizen-science project that combines astrophysics, machine learning, and human efforts to classify glitches in GW interferometer data. The *Omicron* transient search algorithm is used by Gravity Spy to generate q-transform spectrograms and calculate the SNR of the time series samples [10]. This algorithm is crucial in identifying the most useful samples for data classification and analysis. Based on the morphological characteristics from the spectrograms, a total of 22 glitch classes were identified with an SNR above 7.5 and peak frequencies between 10 Hz and 2048 Hz.

In the O3a data for L1, Fast_Scattering and Tomte glitches are the most prevalent, while Chirp, 1080Lines and Wandering_line glitches have fewer samples, as shown in Table 1. The No_Glitch class represents glitch samples that lack significant traits or energy levels and do not fit in with the other classes morphologically. Hence, for our study, we do not consider this glitch class.

Glitch Class	Count	Glitch Class	Count
Fast_Scattering	21749	Whistle	896
Tomte	18708	Low_Frequency_Lines	788
Blip_Low_Frequency	7549	Scratchy	207
Scattered_Light	5398	Repeating_Blips	164
No_Glitch	5358	Violin_Mode	164
Extremely_Loud	4319	Paired_Doves	155
Koi_Fish	4268	Light_Modulation	72
1400Ripples	2363	Helix	21
Blip	1947	Wandering_Line	20
Power_Line	1189	1080Lines	9
Low_Frequency_Burst	1187	Chirp	6

Table 1: Glitch counts per class for LIGO Livingston (L1) during the O3a run.

The time series samples of the glitches are obtained using the `TimeSeries.fetch_open_data()` API provided by GWpy. When querying the API, a sample rate of 4096 Hz is used and a padding of 10 seconds is taken on either side of each of the glitch GPS time to obtain the beginning and end times of the window for preprocessing the glitch. This is done to ensure that the sample being used is large enough for data conditioning.

2.2 The Clean Segment Acquisition Module

The **Clean Segment Acquisition Module** is used to both, find the start and end times of the gaps of clean detector noise as well as obtain the corresponding time series data. This is done using `gwtrigfind`, a package developed to search for event triggers files from GW detectors, in conjunction with `EventTable` and `TimeSeries.fetch_open_data()`, provided by GWpy.

Taking the start and end GPS times of the O3a run, gwtrigfind is used to find the file path containing Omicron triggers from the *L1:GDS-CALIB_STRAIN* channel of the L1 detector. EventTable is then used to load all the trigger data, containing information on the start and end GPS times of the events. Taking the time frames between successive events, we obtain the start and end times of the gaps between glitches/GW triggers, which do not have a significant amount of noise activity. The sizes of the gaps are clamped between 7 and 30 seconds because too short of a gap could lead to the inclusion of glitches in the sample, as the time series data may not have enough time to stabilize after a glitch. Additionally, a short time segment would affect the PSD calculation and subsequent preprocessing. Too long of a time gap could be indicative of detector malfunction or a period of time when it is not operational.

The start and end GPS times of the clean segments are then used to calculate their Q-transform and corresponding p-values. The GPS time intervals with a p-value greater than 0.05 are considered to be clean segments of data.

To obtain the time series samples, similar to the glitch data, the GWOSC database is queried by taking the beginning and ending GPS times as the limits and a sample rate of 4096 Hz.

The clean GPS times from the clean segment acquisition module are saved in a CSV file, and the queried time series data is also stored locally. This allows them to be used for procuring the time series samples without having to query the GWOSC database during multiple stages of analysis.

2.3 Data Preprocessing/Conditioning

The strain readings obtained from a GW detector are usually a combination of the GW signal and detector noise [22–24]. Given our focus on the noise, it is important to bring out its characteristics such that it can be visually observed and studied. Considering the strain output to be $s(t)$ with $n(t)$ as the noise and $h(t)$ being the possible signal received, the GW detector output is given by:

$$s(t) = n(t) + h(t) \tag{1}$$

The preprocessing in this project, which was mentioned in previous section and will be discussed in sections 2.3.1 and 2.3.2, involves the use of Fourier transforms to divide the signal into its frequency components. Given the signal $x(t)$ in the time domain we get its corresponding Fourier transform in the frequency domain $\tilde{x}(f)$ as

$$\tilde{x}(f) = \mathcal{F}\{x(t)\}(f) = \int_{-\infty}^{\infty} x(t)e^{-2\pi ift}dt \tag{2}$$

and the inverse Fourier Transform given by

$$x(t) = \mathcal{F}^{-1}\{\tilde{x}(f)\}(t) = \int_{-\infty}^{\infty} \tilde{x}(f)e^{2\pi ift}df \tag{3}$$

2.3.1 Power Spectral Density and Amplitude Spectral Density

One of the methods to study the glitch signal is the **Power Spectral Density (PSD)**. The PSD, provides a representation of how power is distributed across different frequency bands in a noise signal[25], allowing for the identification of dominant frequency components and their corresponding amplitudes. Assuming the noise sources to be non-stationary and taking the frequency domain representation of the noise signal $n(f)$, the ensemble average of the Fourier components is given by [24]

$$\langle \tilde{n}(f)\tilde{n}^*(f') \rangle = \frac{1}{2}\delta(f - f')S_n(f). \tag{4}$$

Where, $\langle \cdot \rangle$ represents the ensemble average, $\tilde{\cdot}$ represents the Fourier transform from 2, $\cdot^*$ represents the complex conjugate, and $S_n(f)$ is the PSD. Aside from PSD, another way to characterize the noise of a detector is with the **Amplitude Spectral Density (ASD)** values, given by

$$ASD(f) = \sqrt{S_n(f)} \tag{5}$$

The ASD measures the amplitude of the signal at each frequency, and it is often used to compare the noise levels of different detectors or to assess a detector's sensitivity to specific frequencies. It is normally plotted on a logarithmic scale, with frequency on the x-axis and the ASD on the y-axis, allowing for easy identification of noise peaks and their corresponding frequencies.

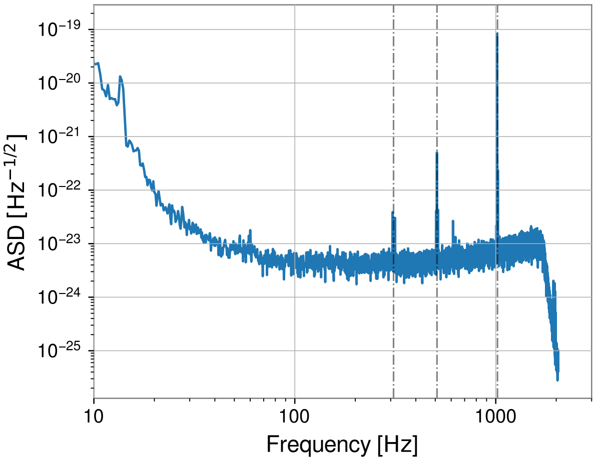


Figure 5: ASD plot for a Tomte Glitch. The gray dotted lines coincide with strong spectral lines at around the 300, 512, and 1024 Hz ranges, marked by dotted lines. There is a big chance that these are of instrumental origin.

From figure 5, which shows an example Tomte glitch ASD plot, to obtain the root-mean-square strain noise at a given frequency band, we integrate over the squares of the ASD readings over the frequency band of interest and take its square root. The problem with ASD plots, however, is that it does not visually capture the glitch signal well enough because they are relatively weak, highly transient and easily overpowered by the instrument noise. To better visualize the glitch, we would instead choose to **whiten** the time series data and experiment with it in the time-amplitude domain.

2.3.2 Whitening

Whitening is the process of suppressing low-frequency and spectral line noise from the data to allow for a clearer view of the underlying signals at sensitive frequency ranges. Whitening is one of the first steps in GW data analysis. By whitening the data, we effectively reduce the impact of noise on our analysis, improving the sensitivity of our detection methods.

Whitening is done by taking the inverse of the estimated ASD and multiplying it with the Fourier coefficients of the original time series data [7].

$$\tilde{n}_{\text{whitened}}(f) = \frac{\tilde{n}(f)}{ASD} = \frac{\tilde{n}(f)}{\sqrt{S_n(f)}} \tag{6}$$

$$n_{\text{whitened}}(t) = \mathcal{F}^{-1}\left\{\tilde{n}_{\text{whitened}}(f)\right\} \tag{7}$$

This effectively flattens the power across all frequencies, allowing for a clearer view of the glitch signal. The resulting whitened data can then be used for further analysis, such as statistical testing or machine learning classification.

In figure 6 we see a Tomte glitch signal before and after whitening. In the first plot we do not clearly see the effects of the glitch on the time series data. However, after whitening the data, we can clearly see the glitch signal in the time-amplitude plot. The Q-transform allows a view of the non-stationary portion of the glitch signal, which shows a clear peak in the normalized energy coinciding with the glitch event. For a clean signal we would not see any significant peaks in the whitened time series or non-stationary parts in the Q-transform.

In this project, the data from the clean and glitch segment acquisition modules are whitened using the method. After whitening, the glitch samples are cropped to a 0.5 second time window around the GPS time of the glitch event, effectively giving us a 1 second glitch sample to work with. The clean data, in turn, is divided into 1 second segments, each being treated as a separate clean sample for statistical testing.

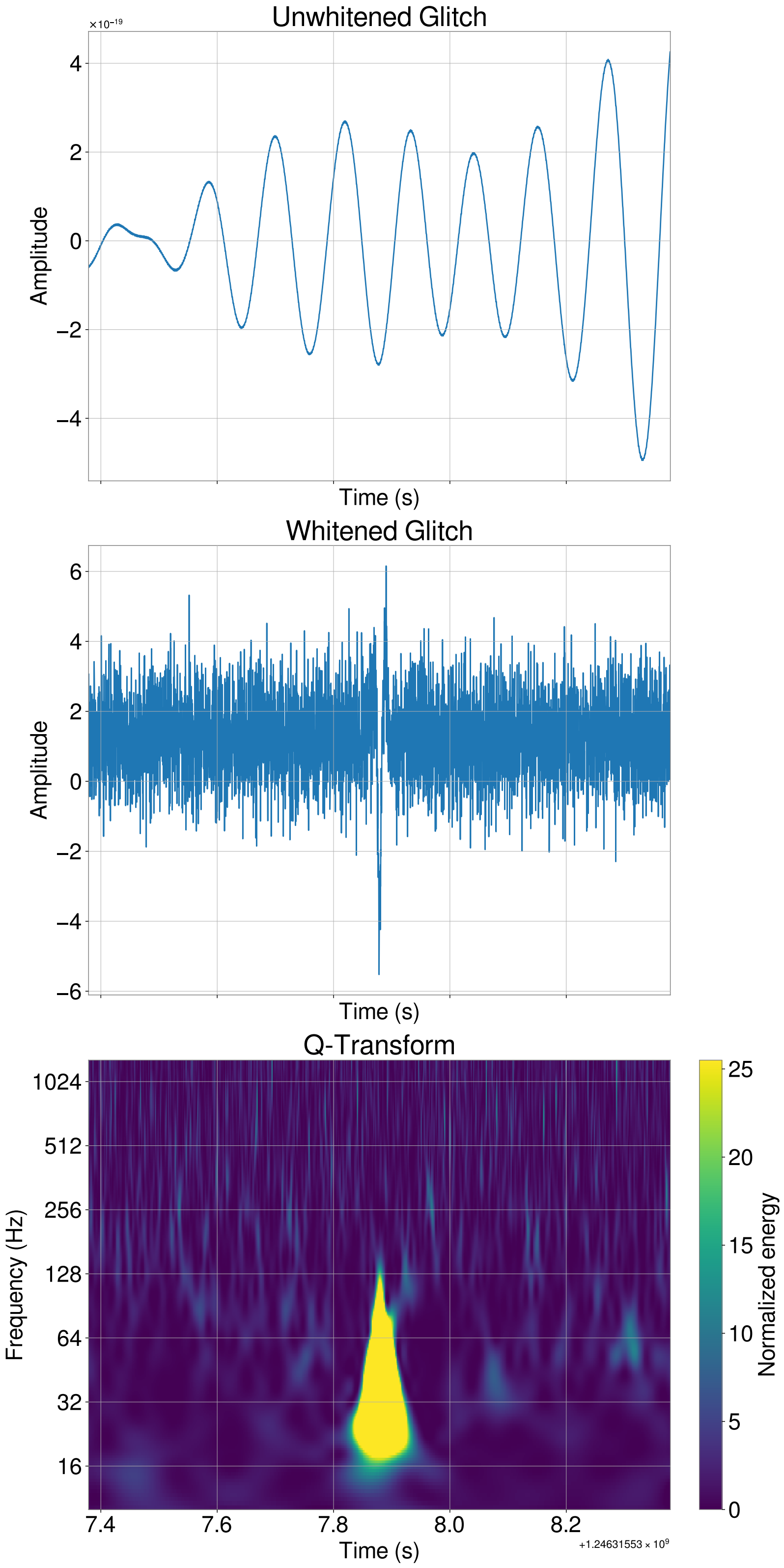


Figure 6: Example of a Tomte glitch signal before and after whitening.

3 Methods

Our main objective is to determine the Gaussianity of time series data obtained from L1 and use this as a criterion to determine the presence of a glitch in it.

If a sample is found to be non-Gaussian, this indicates an excess of power at certain frequencies due to which there is a high chance of a glitch being present. In the previous section, we have seen how whitening the time series data brings out the glitch characteristics. Taking the same example as before, if we were to treat this whitened data as a population of points and plot the distribution, we notice that it contains heavier tails, hence making it non-Gaussian. Similarly, when observing a sample of clean data, the signal is closer to that of a Gaussian distribution. Hence, we propose a pair of hypotheses to determine the presence of a glitch in a sample of data from the detector.

Null Hypothesis (H_0):
Considering a sample of whitened time series detector data, if the distribution of the amplitudes is Gaussian, then the sample does not contain a glitch, i.e. the sample is clean.

Alternative Hypothesis (H_1):
Considering a sample of whitened time series detector data, if the distribution of the amplitudes is non-Gaussian, then the sample contains a glitch.

We employ three statistical tests of normality to test these hypotheses: the Shapiro-Wilk test, the Kolmogorov-Smirnov test, and the Anderson-Darling test. Each of these have their own strengths and weaknesses, and are used to assess the normality of the data in different ways. These tests give us a statistic along with a p-value, the latter of which will be used to determine the significance of the result. The p-value tells us how strongly the statistic rejects the null hypothesis. For our tests, a low p-value indicates that the sample data is unlikely to be Gaussian, hence indicating the presence of a glitch. The significance level for our tests, α, is set to 0.05, which tells us that if the p-value is less than it, we reject the null hypothesis and accept the alternative hypothesis.

3.1 The Shapiro-Wilk Test

The Shapiro-Wilk test [26] is a parametric test, meaning that it assumes the data follows a specific distribution. This parametric test follows the idea of calculating a test statistic W based on the ratio of the best estimate of the sample data's variance to its actual variance. This was the first test that could detect departures from normality due to either the skewness or the kurtosis of the data.

Skewness is a measure of asymmetry of a distribution. It indicates how much the data deviates either to the left or right from a symmetric distribution. A skewness of 0 indicates a perfectly symmetric distribution, while positive and negative values indicate a right and left skew respectively. **Kurtosis**, on the other hand, measures the tails of the distribution, indicating how much of the data is contained in the tails compared to a normal distribution. A kurtosis of 3 indicates a normal distribution, while values greater than 3 indicate heavier tails and values less than 3 indicate lighter tails.

If we were to consider $x_1, x_2, \ldots, x_n$, to be a collection of ordered sample points from a population, ranked smallest to largest, with the sample mean $\bar{x}$ and $a_1, a_2, \ldots, a_n$ being constants computed from the expected values of the order statistics of a normal distribution, then the test statistic W is given by:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \tag{8}$$

Taking $m = (m_1, m_2, \ldots, m_n)$ to be the expected values of the order statistics of a normal distribution such that $m_i = \mathbb{E}[z_i]$ where z_i is the i-th order statistic of the normal distribution, and V the covariance matrix of the ordered statistics, the constants a_i, which make up the **optimal weight vector** $a \in \mathbb{R}^n$, can be computed as

$$\begin{aligned} a_i &= \frac{m^T V^{-1}}{||V^{-1}m||} \quad (9) \\ &= \frac{m^T V^{-1}}{(m^T V^{-1} V^{-1} m)^{1/2}} \quad (10) \end{aligned}$$

The test statistic W lies in the range $[0, 1]$, with values closer to 1 indicating that the sample data is more likely to be Gaussian. The null hypothesis H_0 is rejected if W is significantly less than 1, indicating that the sample data is non-Gaussian.

The p-value of the statistic is usually used to determine the significance of the result. If the p-value is less than the significance level defined, the null hypothesis is rejected, indicating that the sample data is non-Gaussian and likely contains a glitch.

The Shapiro-Wilk test, though powerful, has some limitations. It is sensitive to the sample size and may not perform well with small or large samples. It works ideally for sample sizes $n \leq 2000$. The `shapiro()` function from `scipy.stats` works well for $n \leq 5000$ after which the p-value calculations' accuracy drops. Additionally, for this test to work the data must be continuous and univariate.

3.2 The Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov (KS) test is a non-parametric test, originally proposed in the 1930s, used to decide whether a sample comes from a particular type of distribution [27, 28]. The idea of this test is to check the difference in shape between the distributions being studied, which is done by measuring the maximum vertical distance between the **empirical cumulative distribution function (ECDF)** of the sample and the **cumulative distribution function (CDF)** of the reference distribution, which in this case is a Gaussian distribution. This test has two versions: the **one-sample KS test** which is used to compare an empirical sample distribution with a reference theoretical distribution, and the **two-sample KS test** which compares the empirical distributions of two samples. We use the one-sample KS test in this project.

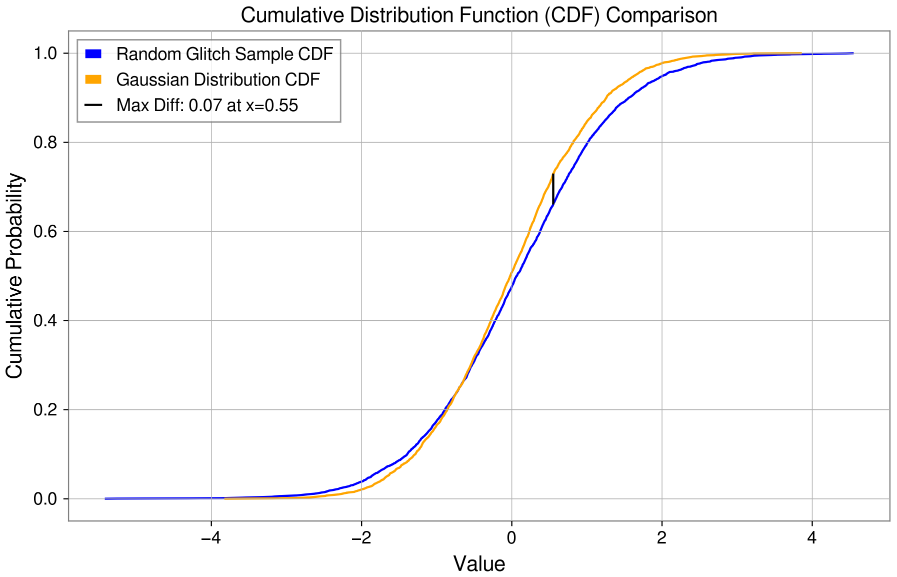


Figure 7: The KS statistic calculation for a sample glitch against a Gaussian distribution. The black line is the maximum absolute difference between the sample glitch CDF and the Gaussian CDF, which is the KS statistic D_n.

Taking a sample of ordered points $X_1, X_2, \ldots, X_N$ as N ordered points from a population, the Empirical CDF of the sample, is given by

$$F_N = \frac{n(i)}{N} \tag{11}$$

Where $n(i)$ is the number of points less than or equal to X_i in the sample. This is in the form of a step function increasing at steps of $1/N$ at each ordered point [29]. Taking the theoretical CDF of the distribution being tested as Fs, a Gaussian distribution in our case, the KS test statistic D_n is defined as

$$D_n = \sup_{1 \leq i \leq N} \left(F(X_i) - \frac{i-1}{N}, \frac{i}{N} - F(X_i) \right) \tag{12}$$

Here, sup refers to the supremum or maximum of the differences between the theoretical CDF and the ECDF. If D_n is significantly larger than the critical value, it indicates that the sample data significantly differs from a Gaussian distribution, hence rejecting the null Hypothesis.

3.2.1 Note: Scaling the data

The Kolmogorov-Smirnov test, is highly affected by the values of the data. We need to ensure that the comparisons being done in the test is not affected due to the absolute values of the data points. Hence, this requires for the data to be scaled down to the same range of a standard normal distribution. This is achieved by scaling the data to have a mean of 0 and a standard deviation of 1. This is also known as **standardization** or **z-score normalization**.

To achieve this the `StandardScaler` class from the `sklearn.preprocessing` module is used. This module works by transposing the data to have a zero mean, calculates the standard deviation, and scales the data points down such that the standard deviation is 1. The **standard score** or **z-score** of a sample point, which tells us how many standard deviations it is from the mean, is given by

$$z = \frac{x - \mu}{\sigma}, \tag{13}$$

where x is the original data point, μ is the mean, and σ is the standard deviation of the data. The z value obtained can be treated as the new data point after scaling and is particularly useful in this case as it does not change the position of the data point in its calculation.

3.3 The Anderson Darling test

The Anderson-Darling (AD) test is a goodness-of-fit test used to determine whether a sample of data is drawn from a specific distribution, most commonly a normal distribution. It is a modification of the Kolmogorov-Smirnov and Cramer-von Mises tests with more sensitivity to differences in the tails of the distribution [29, 30].

Similar to the Kolmogorov-Smirnov test, this test compares the ECDF of the sample data with the CDF of a theoretical reference distribution by taking a distance function between them. The main difference here is that the full range of the data is used to calculate the distance function rather than just the largest distance value [30]. Here, taking n to be the number of elements in the sample and $w(x)$ to be the weighting function, the Anderson-Darling statistic distance A^2, given by the squared area between the sample ECDF $F_n(x)$ and the theoretical reference CDF $F(x)$, is calculated as

$$A^2 = n \int_{-\infty}^{\infty} \left(F_n(x) - F(x)\right)^2 w(x) dF(x) \tag{14}$$

Taking the weighting function $w(x)$ is defined as

$$w(x) = \frac{1}{F(x)(1 - F(x))} \tag{15}$$

We get A^2 [31] as

$$A^2 = n \int_{-\infty}^{\infty} \frac{\left(F_n(x) - F(x)\right)^2}{F(x)(1 - F(x))} dF(x) \tag{16}$$

To obtain random samples from a distribution function, the CDF is to be calculated and random samples x are to be taken from a uniform distribution $x \in [0, 1]$. Hence, the Anderson-Darling test statistic can be rewritten as

$$A^2 = -N - S \tag{17}$$

where S is given by

$$S = \frac{1}{N} \sum_{i=1}^{N} (2i - 1) \left(\ln F(X_i) + \ln(1 - F(X_{N-i+1}))\right) \tag{18}$$

hence, giving us the final form of the Anderson-Darling statistic as

$$A^2 = -N - \sum_{i=1}^{N} \left(\frac{2i-1}{N} \right) \left(\ln F(X_i) + \ln(1 - F(X_{N-i+1})) \right) \tag{19}$$

This is effectively a weighted cross-product of the samples, allowing for a comparison of the sample data with the reference distribution.

In the case of the mean and standard deviation of the sample distribution being known with the distribution size, N being greater than 5, the significance levels are 1%, 2.5%, 5%, 10% and 15%, with the critical values for each significant level being pre-computed through Monte-Carlo simulations. The null hypothesis is rejected if the calculated statistic is greater than the critical value at a selected significance level, indicating that the sample data is not normally distributed.

3.4 Using the tests for glitch detection

In our experiments, we treat the statistical tests as classifiers, where a positive class indicates the presence of a glitch in the sample, irrespective of the type, and a negative class represents a clean signal. This is an approach usually taken with machine learning models, which predict an outcome by learning patterns from the data. In our case, the statistical tests are based on fixed sets of criteria, hence forgoing the learning aspect and directly providing a result based on the input data. This provides a straightforward way of evaluating the performance of each test through the evaluation metrics of the results.

The evaluation metrics used in this project are based on the following definitions:

- **True Positives (TP)**: The number of samples correctly identified as glitches by the test.
- **False Negatives (FN)**: The number of samples incorrectly identified to be clean/Gaussian when they contained glitches.
- **False Positives (FP)**: The number of samples incorrectly identified as glitches when they were actually clean.
- **True Negatives (TN)**: The number of samples correctly identified as clean by the test.

The following evaluation metrics are what we consider for our tests:

- **Accuracy**: The number of correct predictions out of the total number of samples. This is the most common metric used to determine the overall performance of the model or test being studied. However, it can be misleading when taken at face value as it does not account for dataset imbalances or model/test bias.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

- **Recall** or **True Positive Rate (TPR)**: The number of actual positive samples correctly identified as positive over the total number of actual positive samples gives us the *Recall.* This tells us how trustworthy the positive predictions made by the test are.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{21}$$

- **False Positive Rate** (FPR): This is the number of actual negative samples incorrectly identified as positive over the total number of actual negative samples. It is ideal to keep this value low to ensure that our tests do not incorrectly label clean signals as glitches.

$$\text{FPR} = \frac{FP}{FP + TN} \tag{22}$$

- **Precision**: The number of predicted positive samples that were actually positive over the total number of predicted positive samples gives us the *Precision*. The precision tells us how trustworthy the positive predictions made by the test are.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{23}$$

- **F1 Score**: This is another measure of accuracy which makes use of the Precision and Recall. As opposed to Accuracy, which computes how many times the model makes a correct prediction, it sees how well the model performs on each class. Learning models play a balancing act between the Precision and Recall, as the precision places more emphasis on getting the positive predictions right, while recall focuses on getting as many positive predictions as possible. Getting both these values high would be an indicator that the model or test is performing well. The F1 score combines precision and recall into a single metric by taking their harmonic mean to ensure that both maximized.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{24}$$

Along with studying how effective these tests are at determining the presence of a glitch in detector data, we will also be validating the claim in [32], which states that the Shapiro-Wilks test is the most powerful, followed by the Anderson darling test and the Kolmogorov-Smirnov test.

4 Experimentation

The implementation of our statistical tests on the whitened time series data is fairly straightforward, with the `scipy.stats` package providing dedicated functions for each. Once the time series data for the clean and glitch samples are obtained and conditioned, their amplitude values are directly used as inputs to the following functions to obtain the relevant statistics and p-values.

- `scipy.stats.shapiro()` for the Shapiro-Wilk test
- `scipy.stats.ks_1samp()` for the one-sample Kolmogorov-Smirnov test
- `scipy.stats.anderson()` for the Anderson-Darling test

For our experimentation we take up to 101 samples of each glitch type and 700 samples of clean signals, which after querying and preprocessing gives us.

Some glitches, such as 1400 Ripples, 1080 Lines, and Scattered Light, occur at specific frequency ranges. Due to this, we also experiment with using a band pass filter on our data between 10 Hz to 512 Hz and 512 Hz to 1024 Hz, and applying our statistical tests to this filtered data.

4.1 Experimenting on the complete frequency range

Taking the whole frequency range into account, we get the following results (Table 2).

Test	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
Shapiro	913	856	83	1741	0.74	0.52	0.05	0.92	0.66
KS	491	1278	0	1824	0.64	0.28	0.00	1.00	0.43
Anderson	472	1297	0	1824	0.64	0.27	0.00	1.00	0.42

Table 2: Glitch detection results for the full frequency range.

Here, we see that the Shapiro-Wilk test has the best accuracy and F1-score, with a recall of 0.52.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	4	2	0	0	0.67	0.67	0	1.00	0.80
1400Ripples	72	29	0	0	0.71	0.71	0	1.00	0.83
Air_Compressor	7	94	0	0	0.07	0.07	0	1.00	0.13
Blip	97	4	0	0	0.96	0.96	0	1.00	0.98
Blip_Low_Frequency	15	86	0	0	0.15	0.15	0	1.00	0.26
Chirp	2	4	0	0	0.33	0.33	0	1.00	0.50
Extremely_Loud	101	0	0	0	1.00	1.00	0	1.00	1.00
Fast_Scattering	3	98	0	0	0.03	0.03	0	1.00	0.06
Helix	19	2	0	0	0.90	0.90	0	1.00	0.95
Koi_Fish	99	0	0	0	1.00	1.00	0	1.00	1.00
Light_Modulation	67	5	0	0	0.93	0.93	0	1.00	0.96
Low_Frequency_Burst	8	90	0	0	0.08	0.08	0	1.00	0.15
Low_Frequency_Lines	2	99	0	0	0.02	0.02	0	1.00	0.04
Paired_Doves	43	58	0	0	0.43	0.43	0	1.00	0.60
Power_Line	7	93	0	0	0.07	0.07	0	1.00	0.13
Repeating_Blips	98	3	0	0	0.97	0.97	0	1.00	0.98
Scattered_Light	10	91	0	0	0.10	0.10	0	1.00	0.18
Scratchy	8	37	0	0	0.18	0.18	0	1.00	0.30
Tomte	55	46	0	0	0.54	0.54	0	1.00	0.71
Violin_Mode	86	15	0	0	0.85	0.85	0	1.00	0.92
Wandering_Line	9	0	0	0	1.00	1.00	0	1.00	1.00
Whistle	101	0	0	0	1.00	1.00	0	1.00	1.00
Clean_Signal	0	0	83	1741	0.95	0.00	0.05	0.00	0.00

Table 3: Shapiro-Wilk test results for each glitch class at the full frequency range.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	0	6	0	0	0.00	0.00	0	0	0
1400Ripples	0	101	0	0	0.00	0.00	0	0	0
Air_Compressor	0	101	0	0	0.00	0.00	0	0	0
Blip	37	64	0	0	0.37	0.37	0	1.00	0.54
Blip_Low_Frequency	2	99	0	0	0.02	0.02	0	1.00	0.04
Chirp	0	6	0	0	0.00	0.00	0	0	0
Extremely_Loud	101	0	0	0	1.00	1.00	0	1.00	1.00
Fast_Scattering	0	101	0	0	0.00	0.00	0	0	0
Helix	12	9	0	0	0.57	0.57	0	1.00	0.73
Koi_Fish	98	1	0	0	0.99	0.99	0	1.00	0.99
Light_Modulation	60	12	0	0	0.83	0.83	0	1.00	0.91
Low_Frequency_Burst	0	98	0	0	0.00	0.00	0	0	0
Low_Frequency_Lines	0	101	0	0	0.00	0.00	0	0	0
Paired_Doves	20	81	0	0	0.20	0.20	0	1.00	0.33
Power_Line	1	99	0	0	0.01	0.01	0	1.00	0.02
Repeating_Blips	62	39	0	0	0.61	0.61	0	1.00	0.76
Scattered_Light	3	98	0	0	0.03	0.03	0	1.00	0.06
Scratchy	3	42	0	0	0.07	0.07	0	1.00	0.12
Tomte	14	87	0	0	0.14	0.14	0	1.00	0.24
Violin_Mode	18	83	0	0	0.18	0.18	0	1.00	0.30
Wandering_Line	5	4	0	0	0.56	0.56	0	1.00	0.71
Whistle	55	46	0	0	0.54	0.54	0	1.00	0.71
Clean_Signal	0	0	0	1824	1.00	0.00	0.00	0	0

Table 4: Kolmogorov-Smirnov test results for each glitch class at the full frequency range.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	0	6	0	0	0.00	0.00	0	0	0
1400Ripples	0	101	0	0	0.00	0.00	0	0	0
Air_Compressor	0	101	0	0	0.00	0.00	0	0	0
Blip	35	66	0	0	0.35	0.35	0	1.00	0.51
Blip_Low_Frequency	2	99	0	0	0.02	0.02	0	1.00	0.04
Chirp	0	6	0	0	0.00	0.00	0	0	0
Extremely_Loud	101	0	0	0	1.00	1.00	0	1.00	1.00
Fast_Scattering	0	101	0	0	0.00	0.00	0	0	0
Helix	8	13	0	0	0.38	0.38	0	1.00	0.55
Koi_Fish	98	1	0	0	0.99	0.99	0	1.00	0.99
Light_Modulation	58	14	0	0	0.81	0.81	0	1.00	0.89
Low_Frequency_Burst	0	98	0	0	0.00	0.00	0	0	0
Low_Frequency_Lines	0	101	0	0	0.00	0.00	0	0	0
Paired_Doves	17	84	0	0	0.17	0.17	0	1.00	0.29
Power_Line	1	99	0	0	0.01	0.01	0	1.00	0.02
Repeating_Blips	60	41	0	0	0.59	0.59	0	1.00	0.75
Scattered_Light	2	99	0	0	0.02	0.02	0	1.00	0.04
Scratchy	3	42	0	0	0.07	0.07	0	1.00	0.12
Tomte	12	89	0	0	0.12	0.12	0	1.00	0.21
Violin_Mode	17	84	0	0	0.17	0.17	0	1.00	0.29
Wandering_Line	5	4	0	0	0.56	0.56	0	1.00	0.71
Whistle	53	48	0	0	0.52	0.52	0	1.00	0.69
Clean_Signal	0	0	0	1824	1.00	0.00	0.00	0	0

Table 5: Anderson-Darling test results for each glitch class at the full frequency range.

4.2 Experimenting with a band pass filter between 10 Hz to 512 Hz

Test	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
Shapiro	1252	517	735	1089	0.65	0.71	0.40	0.63	0.67
KS	630	1139	24	1800	0.68	0.36	0.01	0.96	0.52
Anderson	589	1180	0	1824	0.67	0.33	0.00	1.00	0.50

Table 6: Glitch detection results for a frequency range of 10 Hz to 512 Hz.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	3	3	0	0	0.50	0.50	0	1.00	0.67
1400Ripples	44	57	0	0	0.44	0.44	0	1.00	0.61
Air_Compressor	54	47	0	0	0.53	0.53	0	1.00	0.70
Blip	101	0	0	0	1.00	1.00	0	1.00	1.00
Blip_Low_Frequency	72	29	0	0	0.71	0.71	0	1.00	0.83
Chirp	5	1	0	0	0.83	0.83	0	1.00	0.91
Extremely_Loud	101	0	0	0	1.00	1.00	0	1.00	1.00
Fast_Scattering	41	60	0	0	0.41	0.41	0	1.00	0.58
Helix	21	0	0	0	1.00	1.00	0	1.00	1.00
Koi_Fish	99	0	0	0	1.00	1.00	0	1.00	1.00
Light_Modulation	71	1	0	0	0.99	0.99	0	1.00	0.99
Low_Frequency_Burst	43	55	0	0	0.44	0.44	0	1.00	0.61
Low_Frequency_Lines	51	50	0	0	0.50	0.50	0	1.00	0.67
Paired_Doves	86	15	0	0	0.85	0.85	0	1.00	0.92
Power_Line	68	32	0	0	0.68	0.68	0	1.00	0.81
Repeating_Blips	98	3	0	0	0.97	0.97	0	1.00	0.98
Scattered_Light	65	36	0	0	0.64	0.64	0	1.00	0.78
Scratchy	34	11	0	0	0.76	0.76	0	1.00	0.86
Tomte	96	5	0	0	0.95	0.95	0	1.00	0.97
Violin_Mode	50	51	0	0	0.50	0.50	0	1.00	0.66
Wandering_Line	3	6	0	0	0.33	0.33	0	1.00	0.50
Whistle	46	55	0	0	0.46	0.46	0	1.00	0.63
Clean_Signal	0	0	735	1089	0.60	0.00	0.40	0.00	0.00

Table 7: Shapiro-Wilk test results for each glitch class at a frequency range of 10 Hz to 512 Hz.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	0	6	0	0	0.00	0.00	0	0	0
1400Ripples	0	101	0	0	0.00	0.00	0	0	0
Air_Compressor	1	100	0	0	0.01	0.01	0	1.00	0.02
Blip	74	27	0	0	0.73	0.73	0	1.00	0.85
Blip_Low_Frequency	11	90	0	0	0.11	0.11	0	1.00	0.20
Chirp	2	4	0	0	0.33	0.33	0	1.00	0.50
Extremely_Loud	101	0	0	0	1.00	1.00	0	1.00	1.00
Fast_Scattering	2	99	0	0	0.02	0.02	0	1.00	0.04
Helix	18	3	0	0	0.86	0.86	0	1.00	0.92
Koi_Fish	99	0	0	0	1.00	1.00	0	1.00	1.00
Light_Modulation	66	6	0	0	0.92	0.92	0	1.00	0.96
Low_Frequency_Burst	3	95	0	0	0.03	0.03	0	1.00	0.06
Low_Frequency_Lines	6	95	0	0	0.06	0.06	0	1.00	0.11
Paired_Doves	48	53	0	0	0.48	0.48	0	1.00	0.64
Power_Line	2	98	0	0	0.02	0.02	0	1.00	0.04
Repeating_Blips	95	6	0	0	0.94	0.94	0	1.00	0.97
Scattered_Light	15	86	0	0	0.15	0.15	0	1.00	0.26
Scratchy	14	31	0	0	0.31	0.31	0	1.00	0.47
Tomte	52	49	0	0	0.51	0.51	0	1.00	0.68
Violin_Mode	12	89	0	0	0.12	0.12	0	1.00	0.21
Wandering_Line	0	9	0	0	0.00	0.00	0	0	0
Whistle	9	92	0	0	0.09	0.09	0	1.00	0.16
Clean_Signal	0	0	24	1800	0.99	0.00	0.01	0.00	0

Table 8: Kolmogorov-Smirnov test results for each glitch class at a frequency range of 10 Hz to 512 Hz.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	0	6	0	0	0.00	0.00	0	0	0
1400Ripples	0	101	0	0	0.00	0.00	0	0	0
Air_Compressor	0	101	0	0	0.00	0.00	0	0	0
Blip	73	28	0	0	0.72	0.72	0	1.00	0.84
Blip_Low_Frequency	8	93	0	0	0.08	0.08	0	1.00	0.15
Chirp	1	5	0	0	0.17	0.17	0	1.00	0.29
Extremely_Loud	101	0	0	0	1.00	1.00	0	1.00	1.00
Fast_Scattering	1	100	0	0	0.01	0.01	0	1.00	0.02
Helix	18	3	0	0	0.86	0.86	0	1.00	0.92
Koi_Fish	99	0	0	0	1.00	1.00	0	1.00	1.00
Light_Modulation	66	6	0	0	0.92	0.92	0	1.00	0.96
Low_Frequency_Burst	0	98	0	0	0.00	0.00	0	0	0
Low_Frequency_Lines	0	101	0	0	0.00	0.00	0	0	0
Paired_Doves	47	54	0	0	0.47	0.47	0	1.00	0.64
Power_Line	1	99	0	0	0.01	0.01	0	1.00	0.02
Repeating_Blips	91	10	0	0	0.90	0.90	0	1.00	0.95
Scattered_Light	10	91	0	0	0.10	0.10	0	1.00	0.18
Scratchy	8	37	0	0	0.18	0.18	0	1.00	0.30
Tomte	48	53	0	0	0.48	0.48	0	1.00	0.64
Violin_Mode	10	91	0	0	0.10	0.10	0	1.00	0.18
Wandering_Line	0	9	0	0	0.00	0.00	0	0	0
Whistle	7	94	0	0	0.07	0.07	0	1.00	0.13
Clean_Signal	0	0	0	1824	1.00	0.00	0.00	0	0

Table 9: Anderson-Darling test results for each glitch class at a frequency range of 10 Hz to 512 Hz.

4.3 Experimenting with a band pass filter between 512 Hz to 1024 Hz

Test	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
Shapiro	555	1214	67	1757	0.64	0.31	0.04	0.89	0.46
KS	279	1490	0	1824	0.59	0.16	0.00	1.00	0.27
Anderson	268	1501	0	1824	0.58	0.15	0.00	1.00	0.26

Table 10: Glitch detection results for a frequency range of 512 Hz to 1024 Hz.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	3	3	0	0	0.50	0.50	0	1.00	0.67
1400Ripples	6	95	0	0	0.06	0.06	0	1.00	0.11
Air_Compressor	2	99	0	0	0.02	0.02	0	1.00	0.04
Blip	47	54	0	0	0.47	0.47	0	1.00	0.64
Blip_Low_Frequency	7	94	0	0	0.07	0.07	0	1.00	0.13
Chirp	0	6	0	0	0.00	0.00	0	0.00	0.00
Extremely_Loud	100	1	0	0	0.99	0.99	0	1.00	1.00
Fast_Scattering	5	96	0	0	0.05	0.05	0	1.00	0.09
Helix	8	13	0	0	0.38	0.38	0	1.00	0.55
Koi_Fish	82	17	0	0	0.83	0.83	0	1.00	0.91
Light_Modulation	55	17	0	0	0.76	0.76	0	1.00	0.87
Low_Frequency_Burst	8	90	0	0	0.08	0.08	0	1.00	0.15
Low_Frequency_Lines	5	96	0	0	0.05	0.05	0	1.00	0.09
Paired_Doves	7	94	0	0	0.07	0.07	0	1.00	0.13
Power_Line	7	93	0	0	0.07	0.07	0	1.00	0.13
Repeating_Blips	58	43	0	0	0.57	0.57	0	1.00	0.73
Scattered_Light	2	99	0	0	0.02	0.02	0	1.00	0.04
Scratchy	4	41	0	0	0.09	0.09	0	1.00	0.16
Tomte	3	98	0	0	0.03	0.03	0	1.00	0.06
Violin_Mode	55	46	0	0	0.54	0.54	0	1.00	0.71
Wandering_Line	9	0	0	0	1.00	1.00	0	1.00	1.00
Whistle	82	19	0	0	0.81	0.81	0	1.00	0.90
Clean_Signal	0	0	67	1757	0.96	0.00	0.04	0.00	0.00

Table 11: Shapiro-Wilk test results for each glitch class at a frequency range of 512 Hz to 1024 Hz.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	0	6	0	0	0.00	0.00	0	0	0
1400Ripples	0	101	0	0	0.00	0.00	0	0	0
Air_Compressor	0	101	0	0	0.00	0.00	0	0	0
Blip	21	80	0	0	0.21	0.21	0	1.00	0.34
Blip_Low_Frequency	0	101	0	0	0.00	0.00	0	0	0
Chirp	0	6	0	0	0.00	0.00	0	0	0
Extremely_Loud	97	4	0	0	0.96	0.96	0	1.00	0.98
Fast_Scattering	0	101	0	0	0.00	0.00	0	0	0
Helix	0	21	0	0	0.00	0.00	0	0	0
Koi_Fish	66	33	0	0	0.67	0.67	0	1.00	0.80
Light_Modulation	33	39	0	0	0.46	0.46	0	1.00	0.63
Low_Frequency_Burst	0	98	0	0	0.00	0.00	0	0	0
Low_Frequency_Lines	0	101	0	0	0.00	0.00	0	0	0
Paired_Doves	0	101	0	0	0.00	0.00	0	0	0
Power_Line	0	100	0	0	0.00	0.00	0	0	0
Repeating_Blips	31	70	0	0	0.31	0.31	0	1.00	0.47
Scattered_Light	0	101	0	0	0.00	0.00	0	0	0
Scratchy	2	43	0	0	0.04	0.04	0	1.00	0.09
Tomte	0	101	0	0	0.00	0.00	0	0	0
Violin_Mode	7	94	0	0	0.07	0.07	0	1.00	0.13
Wandering_Line	7	2	0	0	0.78	0.78	0	1.00	0.88
Whistle	15	86	0	0	0.15	0.15	0	1.00	0.26
Clean_Signal	0	0	0	1824	1.00	0.00	0.00	0	0

Table 12: Kolmogorov-Smirnov test results for each glitch class at a frequency range of 512 Hz to 1024 Hz.

Label	TP	FN	FP	TN	Accuracy	Recall/TPR	FPR	Precision	F1 Score
1080Lines	0	6	0	0	0.00	0.00	0	0	0
1400Ripples	0	101	0	0	0.00	0.00	0	0	0
Air_Compressor	0	101	0	0	0.00	0.00	0	0	0
Blip	20	81	0	0	0.20	0.20	0	1.00	0.33
Blip_Low_Frequency	0	101	0	0	0.00	0.00	0	0	0
Chirp	0	6	0	0	0.00	0.00	0	0	0
Extremely_Loud	96	5	0	0	0.95	0.95	0	1.00	0.97
Fast_Scattering	0	101	0	0	0.00	0.00	0	0	0
Helix	0	21	0	0	0.00	0.00	0	0	0
Koi_Fish	65	34	0	0	0.66	0.66	0	1.00	0.79
Light_Modulation	33	39	0	0	0.46	0.46	0	1.00	0.63
Low_Frequency_Burst	0	98	0	0	0.00	0.00	0	0	0
Low_Frequency_Lines	0	101	0	0	0.00	0.00	0	0	0
Paired_Doves	0	101	0	0	0.00	0.00	0	0	0
Power_Line	0	100	0	0	0.00	0.00	0	0	0
Repeating_Blips	31	70	0	0	0.31	0.31	0	1.00	0.47
Scattered_Light	0	101	0	0	0.00	0.00	0	0	0
Scratchy	2	43	0	0	0.04	0.04	0	1.00	0.09
Tomte	0	101	0	0	0.00	0.00	0	0	0
Violin_Mode	6	95	0	0	0.06	0.06	0	1.00	0.11
Wandering_Line	6	3	0	0	0.67	0.67	0	1.00	0.80
Whistle	9	92	0	0	0.09	0.09	0	1.00	0.16
Clean_Signal	0	0	0	1824	1.00	0.00	0.00	0	0

Table 13: Anderson-Darling test results for each glitch class at a frequency range of 512 Hz to 1024 Hz.

5 Discussions

6 Conclusion

References

[1] B. P. Abbott, R. Abbott, et al. "Observation of Gravitational Waves from a Binary Black Hole Merger". In: *Physical Review Letters* 116.6 (Feb. 2016). ISSN: 1079-7114. DOI: 10.1103/physrevlett.116.061102. URL: http://dx.doi.org/10.1103/PhysRevLett.116.061102.

[2] Timothée Accadia, Fausto Acernese, et al. "Virgo: A laser interferometer to detect gravitational waves". In: *Journal of Instrumentation* 7 (Mar. 2012), P03012. DOI: 10.1088/1748-0221/7/03/P03012.

[3] Andrew Grant. "Advanced LIGO ramps up, with slight improvements". en. In: *Physics Today* 2016.11 (Nov. 2016), p. 12128. ISSN: 19450699. DOI: 10.1063/pt.5.9074. URL: https://pubs.aip.org/physicstoday/online/12128.

[4] The LIGO Scientific Collaboration, the Virgo Collaboration, 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”. In: *Physical Review X* 9.3 (Sept. 2019). arXiv:1811.12907 [astro-ph], p. 031040. ISSN: 2160-3308. DOI: 10.1103/PhysRevX.9.031040. URL: http://arxiv.org/abs/1811.12907.

[5] R. Abbott et al. *GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo During the First Half of the Third Observing Run*. arXiv:2010.14527 [gr-qc]. Mar. 2021. DOI: 10.1103/PhysRevX.11.021053. URL: http://arxiv.org/abs/2010.14527.

[6] The LIGO Scientific Collaboration, the Virgo Collaboration, et al. *GWTC-3: Compact Binary Coalescences Observed by LIGO and Virgo During the Second Part of the Third Observing Run*. arXiv:2111.03606 [gr-qc]. Oct. 2023. DOI: 10.1103/PhysRevX.13.041039. URL: http://arxiv.org/abs/2111.03606.

[7] The LIGO Scientific Collaboration, the Virgo Collaboration, et al. “A guide to LIGO-Virgo detector noise and extraction of transient gravitational-wave signals”. In: *Classical and Quantum Gravity* 37.5 (Mar. 2020). arXiv:1908.11170 [gr-qc], p. 055002. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/1361-6382/ab685e. URL: http://arxiv.org/abs/1908.11170.

[8] The LIGO Scientific Collaboration and the Virgo Collaboration. “Characterization of transient noise in Advanced LIGO relevant to gravitational wave signal GW150914”. In: *Classical and Quantum Gravity* 33.13 (July 2016). arXiv:1602.03844 [gr-qc], p. 134001. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/33/13/134001. URL: http://arxiv.org/abs/1602.03844.

[9] The LIGO Scientific Collaboration and The Virgo Collaboration. “GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral”. In: *Physical Review Letters* 119.16 (Oct. 2017). arXiv:1710.05832 [gr-qc], p. 161101. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.119.161101. URL: http://arxiv.org/abs/1710.05832.

[10] Florent Robinet et al. “Omicron: A tool to characterize transient noise in gravitational-wave detectors”. In: *SoftwareX* 12 (2020), p. 100620. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2020.100620. URL: https://www.sciencedirect.com/science/article/pii/S2352711020303332.

[11] Duncan M. Macleod et al. “GWpy: A Python package for gravitational-wave astrophysics”. In: *SoftwareX* 13 (2021), p. 100657. ISSN: 2352-7110. DOI: https://doi.org/10.1016/j.softx.2021.100657. URL: https://www.sciencedirect.com/science/article/pii/S2352711021000029.

[12] D. Davis et al. “Subtracting glitches from gravitational-wave detector data during the third observing run”. In: *Classical and Quantum Gravity* 39.24 (Dec. 2022). arXiv:2207.03429 [astro-ph], p. 245013. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/1361-6382/aca238. URL: http://arxiv.org/abs/2207.03429.

[13] S. Chatterji et al. “Multiresolution techniques for the detection of gravitational-wave bursts”. In: *Classical and Quantum Gravity* 21.20 (Oct. 2004). arXiv:gr-qc/0412119, S1809–S1818. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/21/20/024. URL: http://arxiv.org/abs/gr-qc/0412119.

[14] Leah Vazsonyi and Derek Davis. “Identifying glitches near gravitational-wave signals from compact binary coalescences using the Q-transform”. In: *Classical and Quantum Gravity* 40.3 (Feb. 2023). arXiv:2208.12338 [astro-ph], p. 035008. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/1361-6382/acafd2. URL: http://arxiv.org/abs/2208.12338.

[15] R. Weiss et al. *Quarterly Progress Report No. 105*. Apr. 1972. URL: https://dspace.mit.edu/handle/1721.1/56271.

[16] The LIGO Scientific Collaboration. *What is an Interferometer?* URL: https://www.ligo.caltech.edu/page/what-is-interferometer.

[17] Sudarshan Ghonge et al. *Assessing and Mitigating the Impact of Glitches on Gravitational-Wave Parameter Estimation: a Model Agnostic Approach*. arXiv:2311.09159 [gr-qc]. Oct. 2024. DOI: 10.48550/arXiv.2311.09159. URL: http://arxiv.org/abs/2311.09159.

[18] Stannered. *Simple Michelson interferometer diagram*. 2007. URL: https://commons.wikimedia.org/wiki/File:Interferometer.svg.

[19] WiredSense. *The Michelson Interferometer - A Laser Lab Alignment Guide*. WiredSense Tutorials. URL: https://www.wiredsense.com/tutorial/the-michelson-interferometer-a-laser-lab-alignment-guide.

[20] D. M. Macleod et al. “GWpy: A Python package for gravitational-wave astrophysics”. In: *SoftwareX* 13 (2021), p. 100657. ISSN: 2352-7110. DOI: 10.1016/j.softx.2021.100657. URL: https://www.sciencedirect.com/science/article/pii/S2352711021000029.

[21] M Zevin et al. “Gravity Spy: integrating advanced LIGO detector characterization, machine learning, and citizen science”. In: *Classical and Quantum Gravity* 34.6 (Feb. 2017), p. 064003. ISSN: 1361-6382. DOI: 10.1088/1361-6382/aa5cea. URL: http://dx.doi.org/10.1088/1361-6382/aa5cea.

[22] Curt Cutler and Eanna Flanagan. “Gravitational Waves from Mergin Compact Binaries: How Accurately Can One Extract the Binary’s Parameters from the Inspiral Waveform?” In: *Physical Review D* 49.6 (Mar. 1994). arXiv:gr-qc/9402014, pp. 2658–2697. ISSN: 0556-2821. DOI: 10.1103/PhysRevD.49.2658. URL: http://arxiv.org/abs/gr-qc/9402014.

[23] Christopher J. Moore, Robert H. Cole, and Christopher P. L. Berry. “Gravitational-wave sensitivity curves”. In: *Classical and Quantum Gravity* 32.1 (Jan. 2015). arXiv:1408.0740 [gr-qc], p. 015014. ISSN: 0264-9381, 1361-6382. DOI: 10.1088/0264-9381/32/1/015014. URL: http://arxiv.org/abs/1408.0740.

[24] M. Lopez. “Exploring the Frontier of Transient Gravitational Wave Detection - Unleashing the Power of Machine Learning”. PhD thesis. Utrecht University, 2025.

[25] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. 3rd. USA: Prentice Hall Press, 2009. ISBN: 0131988425.

[26] S. S. Shapiro and M. B. Wilk. “An Analysis of Variance Test for Normality (Complete Samples)”. In: *Biometrika* 52.3/4 (1965), pp. 591–611. ISSN: 00063444, 14643510. URL: http://www.jstor.org/stable/2333709.

[27] Frank J. Massey. “The Kolmogorov-Smirnov Test for Goodness of Fit”. In: *Journal of the American Statistical Association* 46.253 (1951), pp. 68–78. ISSN: 01621459, 1537274X. URL: http://www.jstor.org/stable/2280095.

[28] I. M. Chakravarti, R. G. Laha, and J. Roy. *Handbook of Methods of Applied Statistics, Volume I*. John Wiley and Sons, 1967, pp. 392–394.

[29] William F. Guthrie. *NIST/SEMATECH e-Handbook of Statistical Methods (NIST Handbook 151)*. en. 2020. DOI: 10.18434/M32189. URL: https://www.itl.nist.gov/div898/handbook/.

[30] Michael J De Smith. "StatsRef.com — The sourcebook for statistics". In: (2025). URL: https://www.statsref.com/.

[31] T. W. Anderson and D. A. Darling. "A Test of Goodness of Fit". In: *Journal of the American Statistical Association* 49.268 (1954), pp. 765–769. DOI: 10.2307/2281537.

[32] Nornadiah Razali and Yap Bee Wah. "Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests". In: *Journal of Statistical Modeling and Analytics* 2.1 (2011), pp. 21–33.