Time series recorded by a MERMAID float in the Pacific: automatic detection and filtering of hydro-acoustic earthquake signals.

Enzo Boulin

10/03/2024

Abstract

A group of autonomous floats equipped with hydrophones, known by the acronym MERMAID, monitors global seismic activity from within the oceans. The instruments are programmed to detect and transmit acoustic pressure conversions from seismic P-wave arrivals, used in mantle tomography. The instruments report seismograms via satellite within hours or days of their recording. However, the floats are generally not recovered. This happened once, and the entire memory was successfully read. We thus inherited a unique dataset: one year of sound recorded at frequencies between 0.1 and 20 Hz in the South Pacific around French Polynesia by a MERMAID float. Additionally, 213 earthquakes were manually identified in this data and linked to listed earthquakes. The onboard detection algorithm only identified 10 of these earthquakes; the others were deemed low priority. We have developed various signal processing and machine learning techniques to improve the instrument's performance.

Acknowledgments

With the valuable help of:
Karin Sigloch, Sébastien Bonnieux, Philippe Blanc, Sébastien Travadel.

1 Introduction

Global seismic tomography—imaging the three-dimensional structure of wave velocity inside the Earth—is limited by the scarcity of oceanic stations.

Approaches to mitigate this problem include installing moored hydrophones and ocean-bottom seismometers. The logistical difficulties and high costs of installation and data recovery make these methods unviable for filling vast oceanic gaps with a station density sufficient for seismic tomography. Repurposing ocean-bottom telecommunication optical fibers for distributed acoustic sensing could extend the reach of existing seismic networks. MERMAID (Mobile Earthquake Recording in Marine Areas by Independent Divers) is a recent, better-established alternative.

This low-cost, easy-to-deploy, and generally unrecovered robotic instrument is capable of maintaining a constant depth in the ocean, where it continuously records the acoustic pressure field and autonomously reports sismo-acoustic waveform arrivals in near real-time. A combination of time-domain triggering algorithms and wavelet-domain probabilistic identification, executed onboard, determines detections of probable earthquake P-wave arrivals. This triggers the MERMAID to rise to the surface to report the recorded waveforms via satellite before resuming its mission.

Over the last decade, several generations of MERMAID instruments have collected thousands of earthquakes usable for seismic studies. However, the majority of these recordings are never transmitted and remain in the instrument's memory, which, in the third generation of MERMAID, holds 1 year of data. The buffer may contain undetected earthquakes and noise from various terrestrial, oceanic, and biological sources.

Exceptionally, during a cruise of the SPPIM (South Pacific Plume Imaging and Modeling) experiment conducted in August 2019, Princeton University's instrument P0023 was recovered and redeployed, allowing the retrieval of a one-year time series.

A PhD student at Princeton University, Sirawich Pipatprathanporn, undertook the task of manually detecting all earthquakes present in this time series (see [1] for details). He first identified all possible earthquake arrivals in the buffer, then matched them, where possible, with known earthquakes from the Preliminary Determination of Epicenters (PDE) database of the National Earthquake Information Center (NEIC) of the United States Geological Survey (USGS), accessible via the Data Management Center (DMC) of the Incorporated Research Institutions for Seismology (IRIS). In total, 213 wave arrivals were matched in this way. Only 10 of them had previously been transmitted by MERMAID.

Our study aims to determine how to efficiently process the signal and develop different earthquake detection algorithms, more effective than the one currently in place. More information on MERMAID and the onboard algorithm can be found here.

2 Data

2.1 Recordings

Our data consists of time-domain recordings of acoustic pressure acquired by MERMAID P0023 at a parking depth of 1500m below the ocean surface in French Polynesia, Pacific, between its first deployment on September 13, 2018, and its fortuitous recovery on August 15, 2019.
Their sampling frequency is 40 Hz, corresponding to a Nyquist frequency of 20 Hz. Between surfacings, which take about 22 hours round-trip and during which recording is interrupted, the time series is continuous except for sporadic intervals of depth adjustments, which interrupt data acquisition for a few minutes each.
MERMAID rises to the surface as soon as it estimates that a detected P-wave arrival is likely to be used for seismic tomography, which occurs on average every 6-7 days. At this data return rate, the lifespan of a MERMAID instrument operating on a single set of lithium batteries is approximately 5 years. At the time of data transmission, Global Positioning System (GPS) position and time are obtained and bundled as metadata. The GPS timestamp is used to correct the instrument clock drift, typically a fraction of a second. During the 11-month period (approximately 336 days or 8064 hours) discussed here, MERMAID surfaced 44 times. In total, we have 7029 hours of data, equivalent to 87% of the deployment time. An example of a recording is given in Figure 1.

2.2 Listed Earthquakes

We also have a database of 213 earthquakes (thanks to [1]). We have access to various information for each earthquake; we only used:

- The arrival date of the earthquake
- Its category (Tag)

Indeed, there are different categories summarized in this table:

DET	Earthquake detected by MERMAID
REQ	Earthquake requested by satellite
(***,**,*)	Manually picked P-wave
(S3,S2,S1)	Manually picked Surface wave

For manually detected earthquakes, levels 1, 2, 3 range from the least remarkable events to the most remarkable (higher signal-to-noise ratio).

3 Signal Processing

The observed signal is very noisy (cf. Figure 1). A first idea was to look at its spectrum. We then observe:

We can see that the noise is localized in the frequency band $[0.1, 1]$. The height and width of these bumps vary with the seasons but are not the subject of our study (see [1] for more information). The noise below 0.05Hz is due, at least in part, to the decrease in sensor sensitivity.
Now, let's observe the spectrogram of a signal containing an earthquake. Since the previous spectrum was computed over the entire

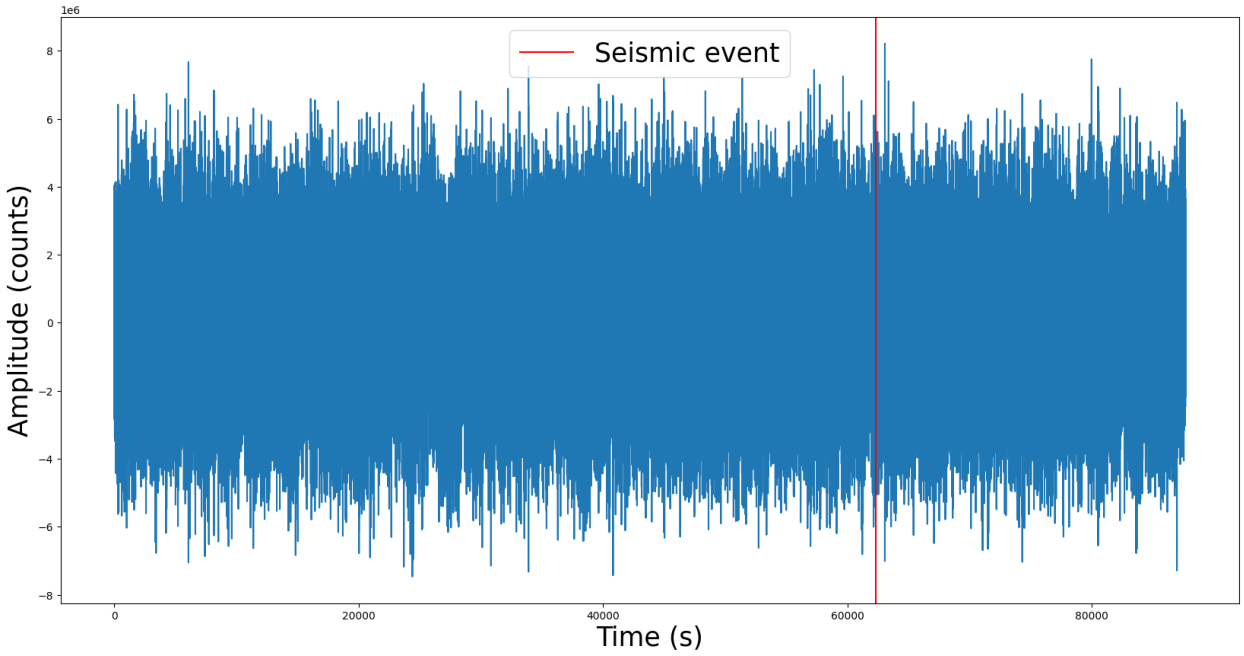

Figure 1: Example of a MERMAID recording containing an earthquake

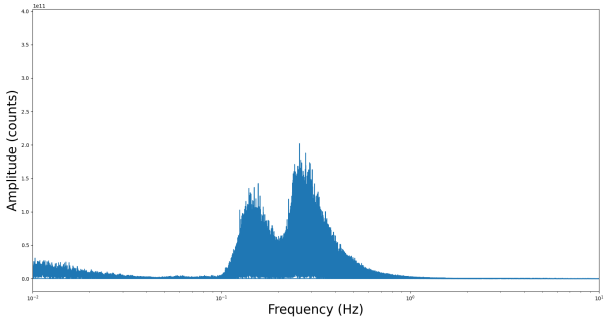

Figure 2: Frequency spectrum of a recording

duration of the signal, the trace of a potential earthquake would have been erased by averaging. The spectrogram avoids this phenomenon.

Two recurring phenomena then appear in the frequency band $[0.05, 0.1]$. First, a distinctive break at the moment of the P-wave arrival; this break corresponds to its frequency signature. Then a second wave, longer and more powerful, which gradually attenuates: this is the surface wave. It therefore appears that the frequency band between 0.05 Hz and 0.1 Hz is the one to watch. This seemed very surprising to us at first since the algorithm onboard MERMAID filters below 2 Hz to keep only high frequencies. Indeed, at the time of the MERMAID algorithm design, no earthquake had been recorded by hydrophones; the designers were therefore inspired by terrestrial seismograms on which P-waves are detected using frequencies above 2 Hz. In theory, we should observe a break marking the P-wave arrival between 0.05 Hz

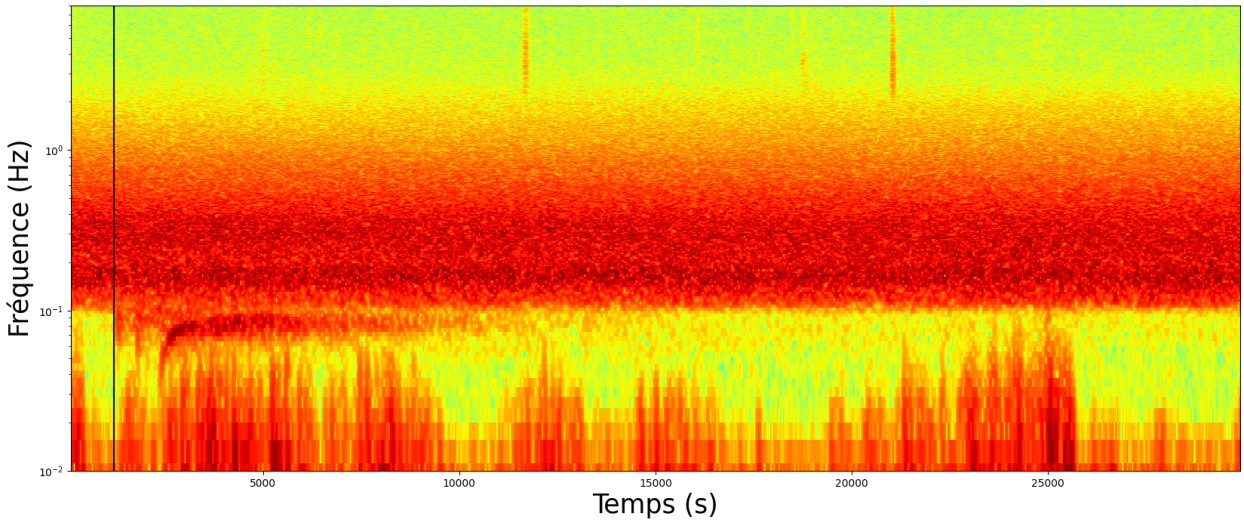

Figure 3: Spectrogram of a manually picked earthquake; the black line corresponds to the P-wave arrival time

and 8 Hz; in practice, we rarely observe the break between 1 Hz and 8 Hz and never the one between 0.1 Hz and 1 Hz due to noise. Furthermore, note that on the spectrogram, we clearly find the two noise bumps between 0.1 and 1 Hz from Figure 2.

We therefore filtered our signals with a 3rd-order Butterworth band-pass filter and down-sampled by a factor of 100 to increase calculation speed. Then, since we are interested in the signal energy in this band, we took a sliding average over 150 seconds of the squared signal. Illustration of this pre-processing in Figure 4.

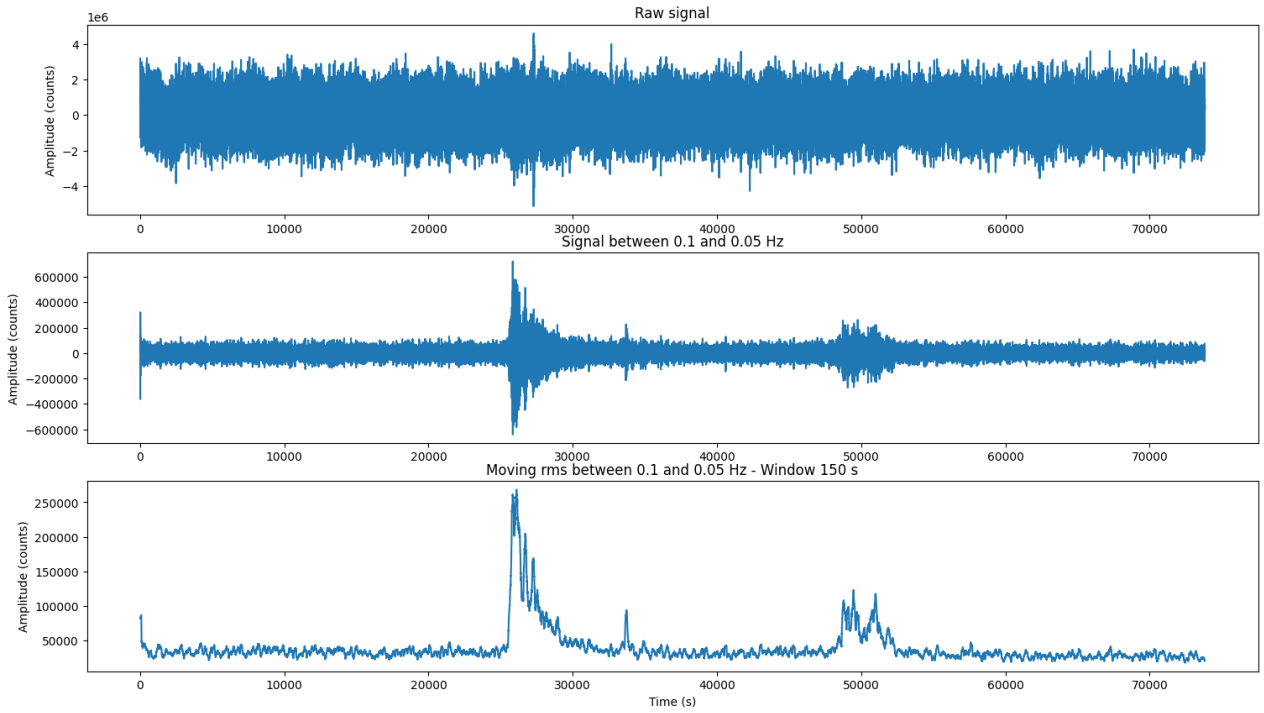

Figure 4: Data pre-processing, from top to bottom: raw signal, signal filtered between 0.05 and 0.1 Hz, squared filtered signal averaged over 150 seconds.

This method allows seismic events to appear clearly with a positive signal that varies less than the initial signal and contains far fewer points.

4 Detection Algorithms

The challenge of this section is to find an algorithm capable of detecting the "peaks" corresponding to earthquakes like those in Figure 4. Obviously, they are not always as remarkable as the 2 earthquakes present in Figure 4, and sometimes there are peaks but no earthquake. Another important point is that earthquakes are very rare events and the buoy has significant energy constraints; it must only surface if absolutely necessary. We therefore want to limit false positives as much as possible.

4.1 Random forest on the time series

Our first idea was to run a random forest directly on the pre-processed series (the one at the bottom of Figure 4). Indeed, it is a robust model that we master well because we have worked on it extensively. It is simple to implement, and very poor results allow for immediate rejection of the ap-

proach. Concretely, we first created 2h windows (2880 points). For positive labels, an earthquake in the middle, and for negative labels, only noise. We thus had 120 positive labels and 1600 negative labels. Indeed, initially, we had 213 earthquakes, but some did not have a 1h margin before and after the earthquake arrival, and we excluded category * and S1 earthquakes because they had peaks that were too weak and caused confusion with noise. To increase the number of positive labels and make the model more robust—in the sense that the earthquake does not necessarily have to be exactly in the middle of the 2-hour window to be detected—we shifted each earthquake by a small Δt taken from a normal distribution with zero mean and standard deviation $\frac{1h}{1000}$. We thus created 800 positive labels with an earthquake at $1h+\Delta t$ in the window. Regarding the main parameters of the forest: 50 trees, each tree having a maximum depth of 20. The model performance is shown in Figure 5 and Table 1.

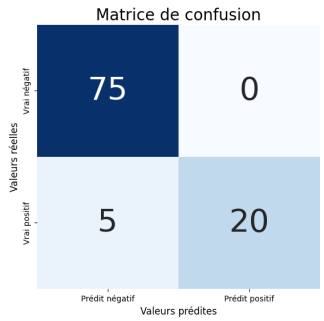

Figure 5: Confusion matrix (in %) of the random forest with a 10% test set

These results seem very satisfactory. However, they are overestimated because, in practice, an earthquake is a very rare event. Over the entire hard drive:

accuracy	0.95
precision	1.0
recall	0.8
f1-score	0.89
kappa	0.86

Table 1: Random forest performance with a 10% test set

- 132 windows contain an earthquake
- 6473 windows do not contain an earthquake

So the *prior* probability of encountering a two-hour window with an earthquake is less than 0.1. However, the algorithm was trained on classes much less imbalanced than this, and there is always the energy constraint to limit false positives. We must therefore change the detection threshold. Indeed, until now, the random forest classifies a time series by a majority vote among all binary trees. We can decide that a window is classified as containing an earthquake if more than 70% of the binary trees think it is an earthquake. In this way, we will limit false positives since we increase our requirement for the quality of detected earthquakes. To refine the model, we will study the threshold effect on three performance measures:

- False Positive Rate: probability of false alarm

$$FPR = \frac{FP}{FP + TN} = \frac{FP}{N}$$

- True Positive Rate: probability of correct alarm

$$TPR = \frac{TP}{TP + FN} = \frac{TP}{P}$$

- Precision: percentage of justified alarms

$$Precision = \frac{TP}{TP + FP}$$

When we plot these measures as a function of the thresholds—that is, the minimum rate of trees that must classify the time series as an earthquake for the forest to classify the time series as an earthquake—we obtain Figure 6.

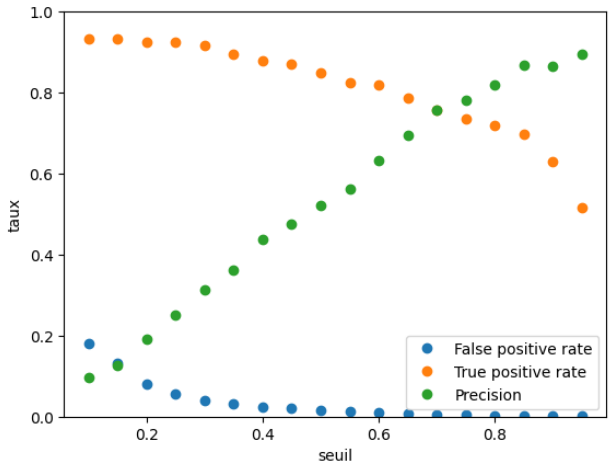

Figure 6: True Positive Rate, Precision, False Negative Rate, as a function of the precision threshold

We can see in Figure 6 that the false alarm probability is not very constraining, which is logical because the class corresponding to the alarm triggering (presence of earthquake) was smaller than the other, so there is little chance of creating a false alarm. Since we only have 2 measures left, we can plot one against the other and see the threshold that allows approaching the ideal point (1,1) (precision of 1 and True Positive Rate of 1). This curve is called the precision-recall curve (another name for the True Positive Rate); it is represented in Figure 7.

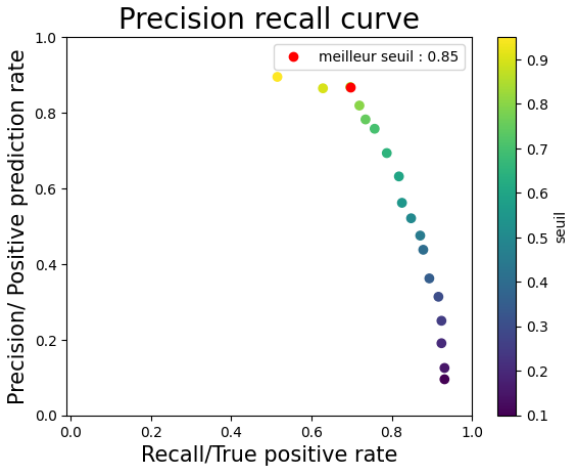

Figure 7: Precision-recall curve, with a heatmap corresponding to the threshold value

After refining the threshold and running the algorithm in "real" conditions on the hard drive, we obtain the results in Table 2:

accuracy	0.95
precision	0.87
recall	0.7
f1-score	0.77
kappa	0.81

Table 2: Performance of the random forest on the hard drive after refining the threshold

These results are slightly lower than the initial results, but this was foreseeable since we are here in real conditions with sometimes earthquakes that were missed in the hard drive, or because earthquakes are much less frequent than during algorithm training. Moreover, to avoid the buoy surfacing too often, we can decide, for example, to only immediately report earthquakes detected with a rate greater than 0.95 and report all those between 0.85 and 0.95 at that time. The advantage of surfacing right after detection is to be more precise on the earthquake location; however, the position can be estimated by interpolation between the different surfacings.

4.2 Lightweight Random Forest

The previous model includes 2880 features highly correlated with their neighbors, and if the model manages to be performant with such a small test set relative to the model size, it means we can reduce the model size. To do this, we started by looking at the importance of each feature in a model trained with the earthquake always exactly in the middle of the window to see which time step the model looks at first (cf. Figure 8).

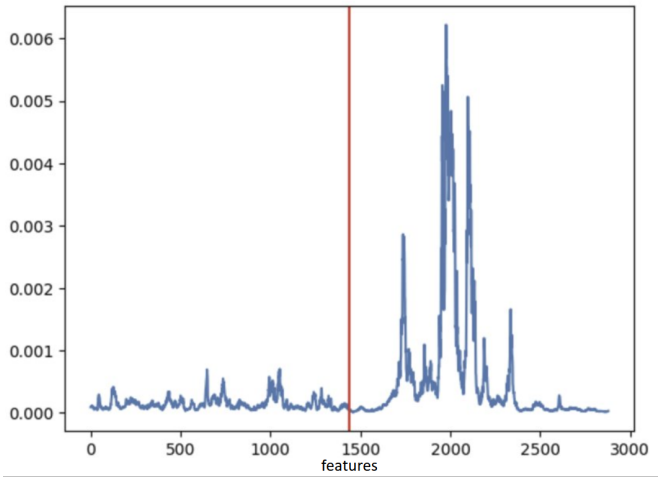

Figure 8: Relative importance of each time step in the random forest, averaged over 100,000 trees; in red, P-wave arrival.

We were very surprised to see that the model almost only looks at what happens in the minutes following the P-wave arrival. However, if we refer to Figure 3, it is the surface wave that arrives in the minutes following the P-wave. It is therefore sufficient to detect the arrival of this wave. Indeed, the surface wave is more intense than the P-wave and thus logically easier to detect. For example, in Figure 9, the P-wave is barely visible, but the surface wave is very visible.

We therefore developed this algorithm to reduce the model size. We take the raw signal, keep only the 0.05-0.1 Hz band, position ourselves at the P-wave arrival, and then separate the signal into 4 intervals (details will

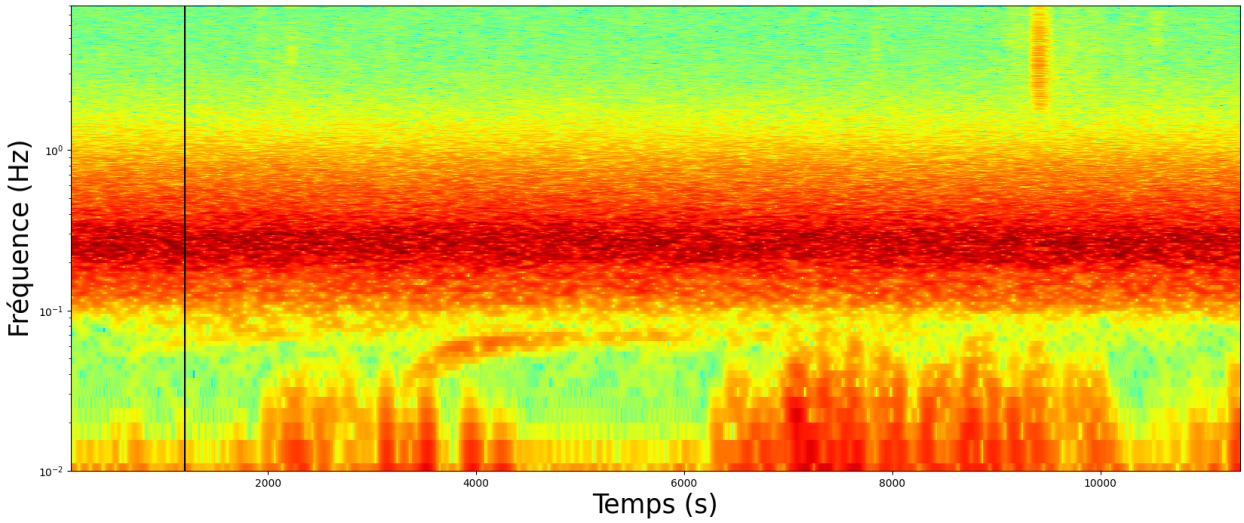

Figure 9: Spectrogram of a * earthquake, the black line corresponds to the P-wave arrival time; only the surface wave is visible

be in the appendix). The 4 features are the average of the rolling standard deviation on these 4 segments. We have thus reduced our model from 2880 to 4 features. Here are the obtained performances 3, 4.

	Predicted positive	Predicted negative
True positive	17	7
True negative	1	47

Table 3: Performances of the 4-feature random forest

accuracy	0.88
precision	0.94
recall	0.71
f1-score	0.81
kappa	0.71

Table 4: Performances of the 4-feature random forest

The performances of this algorithm are quite similar to those of the first one, with many other advantages. First, the model being small, the risk of overfitting is much lower than in the first case. Next, this program consumes less energy than the first since it is lighter. Finally, this model is not optimized at all because we did not have much time to dedicate to it. To start, it would be necessary to manually pick, or use a model to pick, all surface wave arrivals because, in this model, the arrival was roughly estimated relative to the P-wave arrival, and the time delta between the two arrivals varies greatly depending on the distance traveled by the waves. Finally, we did not test many signal processing methods for this technique, and another method than the average of the rolling standard deviation could be more effective.

4.3 STA/LTA on low frequencies

This last algorithm does not use any machine learning model; it is therefore more reliable in the sense that there is no overfitting problem or "black box" phenomenon. It is based on the same method as the algorithm currently used but on the 0.05-0.1 Hz frequency band. This algorithm is widely used in the study of seismic signals; it consists of taking a first rolling average of the squared signal over a "short" window, Short Term Average STA, and another rolling average over a "long" window, Long Term Average LTA, and then displaying the ratio of the two, STA/LTA. At the arrival of an earthquake, since the short window is more reactive than the long one, the ratio will increase before decreasing when the second window reacts; for more details see [2]. For this method, we chose a first window of 1h40 and a second window of 4h. This is one of the first problems of this method: very long windows are needed to have sufficiently stable ratios. However, this method does not require knowing the future to be executed; MERMAID can be surfaced as soon as the threshold is exceeded, unlike random forests which require a 1h offset to function. A first example for this method is given in Figure 10.

The method seems to work well; it highlights forgotten events. Like the other methods, it struggles with low-amplitude earthquakes that do not exceed the threshold. It has the advantage of being very economical because convolutions do not require great computing power. We had difficulty quantitatively analyzing the performance of this method. This method has the advantage of being able to estimate the signal duration by recording the moment the ratio drops back below the average (cf. Figure 11). The results of this method are very satisfactory, although it still misses many low-amplitude earthquakes.

5 Conclusion

Our study showed that the seismic event detection performance of MERMAID buoys could be significantly improved by looking at the [0.05, 0.1] Hz frequency band rather than [2, 8] Hz. To this end, we implemented different detection algorithms: two random forests and the STA/LTA analysis. The time allocated to us (3 months) did not allow us to evaluate the different models with sufficient precision to distinguish one as particularly effective compared to the others. We also showed that it is essential to look at surface waves to detect low-amplitude earthquakes. Nevertheless, these algorithms remain less effective than a human operator looking at the spectrogram or listening to the earthquakes; thus, very low-amplitude earthquakes (class * and S1) are rarely detected by our algorithms.

6 Perspectives

For the continuation of this project, it would be interesting to pursue the "lightweight" random forest by trying to find surface wave

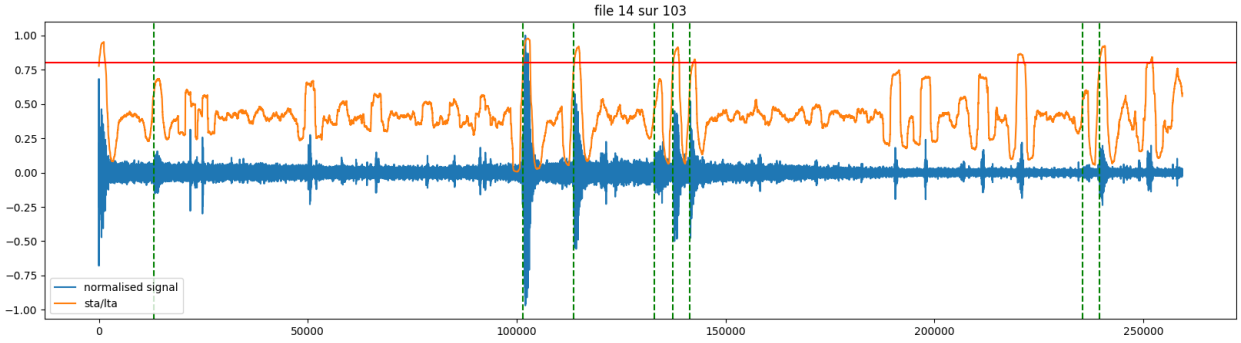

Figure 10: STA/LTA analysis on an entire time series, the detection threshold is in red, known earthquakes in green

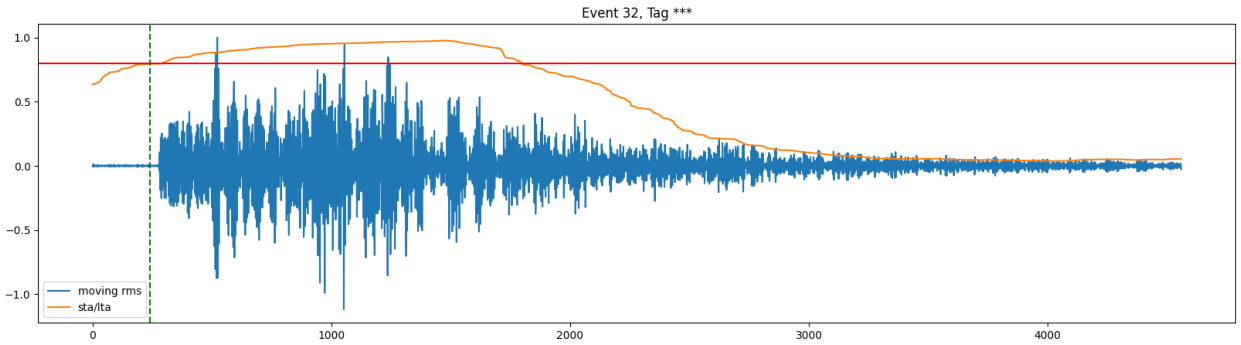

Figure 11: STA/LTA analysis on an earthquake, the detection threshold is in red, P-wave arrival in green; earthquake duration can be estimated as the time between intersections of the threshold with the STA/LTA curve

arrival times, either manually or using a physical model based on P-wave arrivals. It will then be necessary to choose the optimal way to slice the intervals after the surface wave arrival and process them. Thus, the algorithm will be trained on more precise labels. Another avenue would be to continue the STA/LTA algorithm by trying to adjust the threshold and the duration of the sliding windows to obtain a robust algorithm without risk of bias since it requires no training. Finally, it could be interesting to study the noise in more detail to understand how to suppress it since part of the seismic signal is lost in this frequency band.

References

1. Sirawich Pipatprathanporn, Frederik J Simons, *One year of sound recorded by a MERMAID float in the Pacific: hydroacoustic earthquake signals and infrasonic ambient noise*, Geophysical Journal International, Volume 228, Issue 1, January 2022, Pages 193–212

2. Allen, R. V. (1978), *Automatic earthquake recognition and timing from single traces*, Bull. Seismol. Soc. Am., 68(5), 1521–1532.

3. Sukhovich, A., J.-O. Irisson, F. J. Simons, A. Ogé, Y. Hello, A. Deschamps, and G. Nolet (2011), *Automatic discrimination of underwater acoustic signals generated by teleseismic P-waves: A probabilistic approach*, Geophys. Res. Lett., 38, L18605,

4. A. Douglas; *Bandpass filtering to reduce noise on seismograms: Is there a better way?*. Bulletin of the Seismological Society of America 1997; 87 (3): 770–777.

5. A. Sukhovich *et al.* What is MERMAID?

GitHub

The source code for the signal processing pipeline, the Random Forest implementations (both full-scale and lightweight), and the STA/LTA analysis scripts can be found on GitHub.