Anomaly detection in signals

Hugo BRINES, Mathis FABRE
Advanced Artificial Intelligence
IAMD TELECOM Nancy, 2025-2026

Teacher : Hamed KHAKZAD

1 Introduction

1.1 Goal

The goal of the project is to detect anomalies (peaks) in periodic signals with drifts.
In particular, the system has to make distinction between drifts and peaks.

1.2 Definitions

Let's take a signal with 7 harmonics : 60, 70, 84, 105, 140, 210, 410.
Their lcm (Lowest Common Multiple) is 420 so the global period of the signal is 420 points.
Here is what one period of the signal looks like :

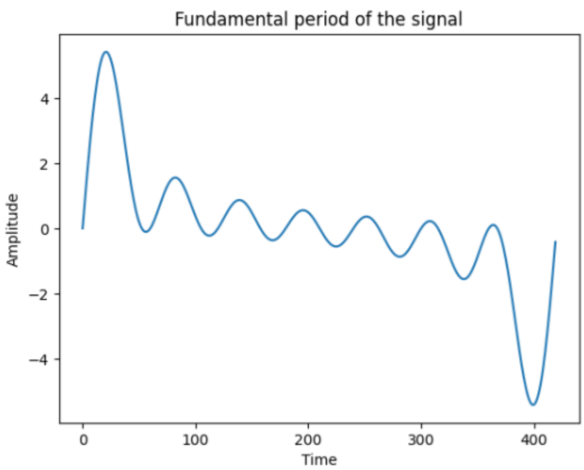
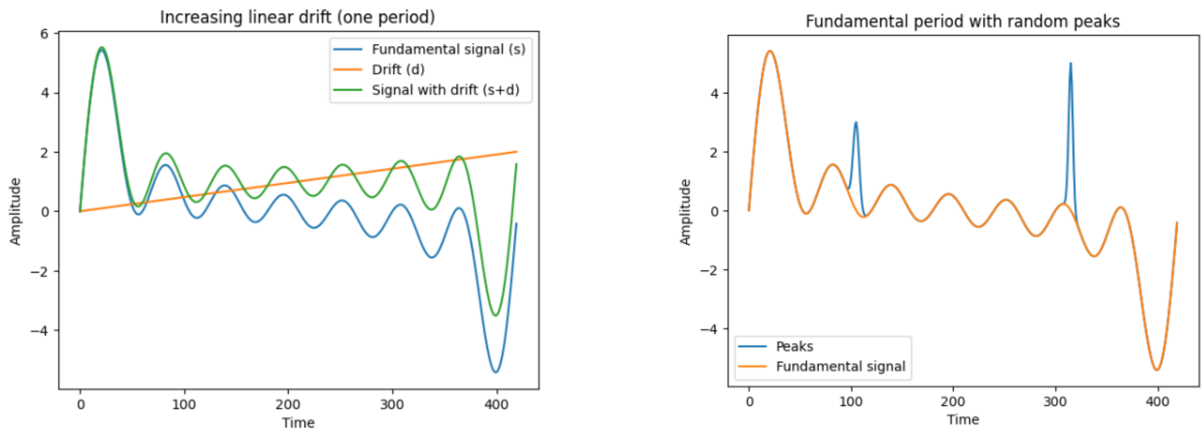


Figure 1 – Fundamental period example

A drift is a change of the signal but during a long time, while a peak is an abnormal change of a signal at just one point.
Here are two figures showing the difference, with the same 7 harmonics :



(a) Example of linear drift (slope : 2) (b) Example of random peaks

FIGURE 2 – Difference between drifts and peaks

If we combine both :

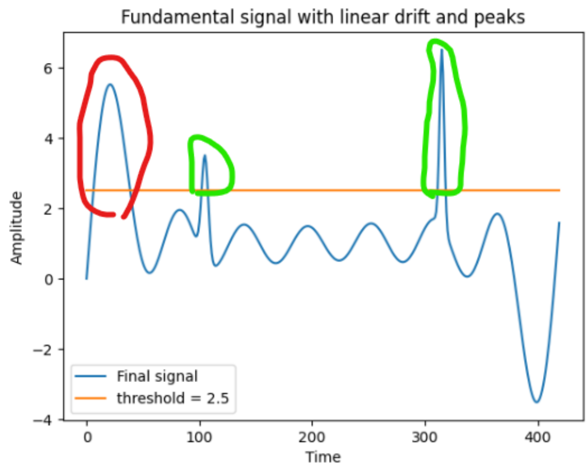


FIGURE 3 – Modified period example

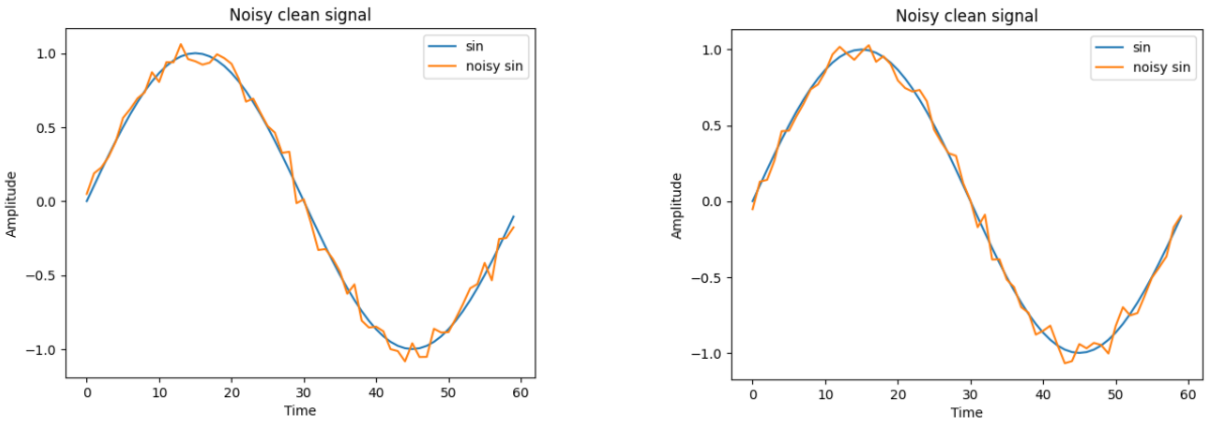
If we could first think of just applying a threshold through all the signal, we can see above it would not work because it would find a false positive : the first peak is just part of the fundamental period.
Moreover, the linear drift would make all the signal superior to the threshold after a moment, so this is not a good approach. We need to use Deep Learning for this task.

2 Dataset

We generated a signal with 7 harmonics, and of period $T = 60$

The reference period is repeated, and with each repetition, random noise is added to increase the model's robustness to variations.
Here is an example of two different repetitions with a simple sin :

For each repetition, we scroll through the sequence so that the model sees different phase shifted versions of the period. Here's an example :



(a) Noisy sin 1 (b) Noisy sin 2

FIGURE 4 – Two different repeated sin periods with different noises

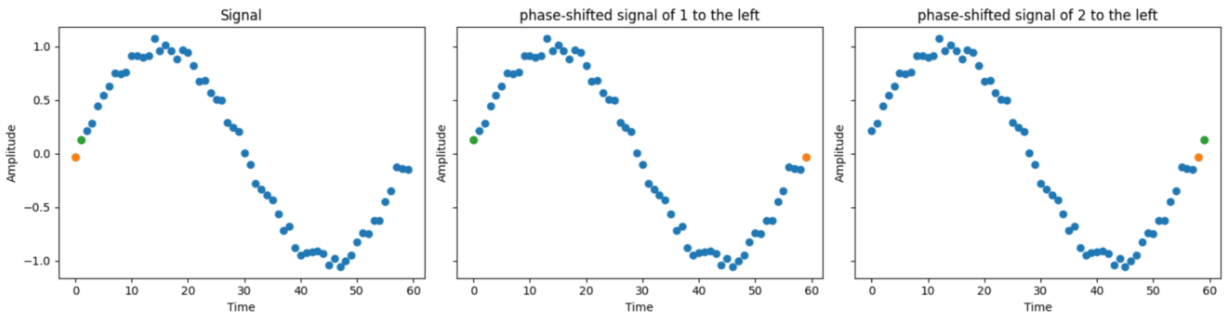


FIGURE 5 – Shifting the signal

To create more data, we also use different type of drifts depending on the period. Let's write d(t) the drift during the time t (with $t = 0$ being the begining of the period), then the different drifts are defined as :

- Exponential : $d(t) = 5\frac{e^{2.5t}-1}{e^{2.5}-1}$
- Linear : $d(t) = -4t$
- Steps : 5 five steps with an amplitude of 6.
- Sin : $d(t) = 4.5\sin(3)$
- Polynomial : $d(t) = 5.5t^2$
- Steps + Sin

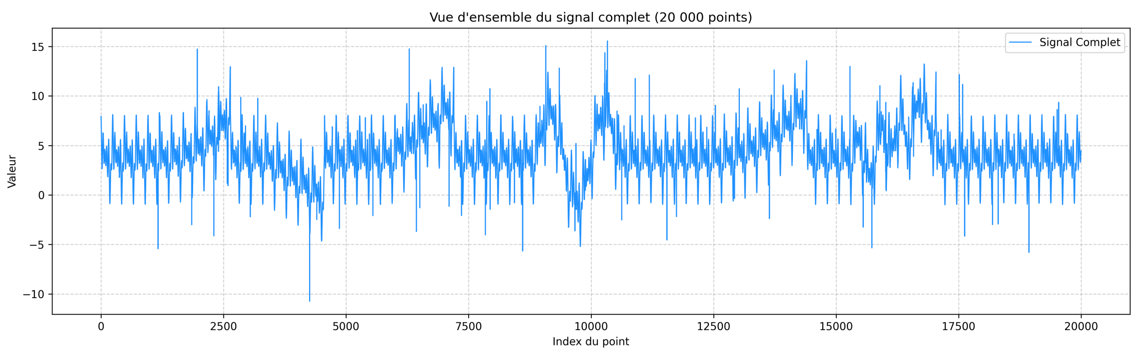


FIGURE 6 – Final signal

The final signal used is also visible in the iav folder, as signal_visualization.png

3 Model used

3.1 How does training work ?

Reminder : our signal has a fundamental period of 60 points.

Our Neural Network will take 59 points of a phase-shifted noisy period to predict the value of the last one, and significant error between the predicted value and the actual value indicates an anomaly. Here's a visualization to better understand it :

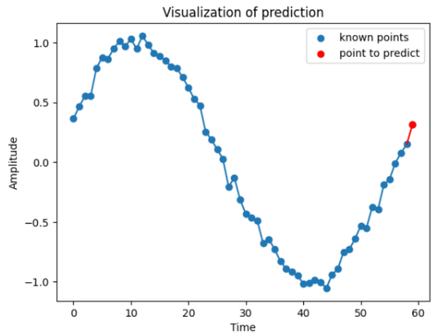


FIGURE 7 – Prediction visualization

3.2 The structure

Here is the structure of our Neural Network :

We first have the input layer. It takes a (59, 3) parameters. The "3" here delivers the signal, but also the sinusoidal encoding of the phase in the period, and the cosine encoding of the phase in the period. Basically, it helps knowing "where we are in the period".

We have six blocks, and each block follows this general structure :

— 1. Conv1D (128 neurons, kernel size = 7) → ReLU → SpatialDropout1D
— 2. Conv1D (128 neurons, kernel size = 7)
— 3. LayerNormalization
— 4. Self-Attention (MultiHeadAttention) + Dropout
— 5. Residual Add + ReLU

After the 6 blocks, we have :

— GlobalAveragePooling1D : Converts the sequence into a fixed-size vector
— Dense (32) : Compact representation of the observed period
— Dense (1) : Predicts the next time step

4 Results

All the results can be checked by going to the src folder and puting in the terminal (after installing seaborn and sklearn packages) :

python LSTM/visualiser_anomaly_csv.py
–original data/test_1.3.csv –analysis result/test_1.3_anomalies.csv

Here are the results obtained :

	Pfredicted Normal	Predicted Anomaly
Real Normal	19909	2
Real Anomaly	73	16

TABLE 1 – Confusion Matrix

Metrics	Value
Precision	0.8889
Recall	0.1798
Accuracy	0.9952
F1	0.2991
Total points used	20 000
Threshold used	0.7392

TABLE 2 – Other scores

The threshold can be changed manually, and the tables will then update at the same time !

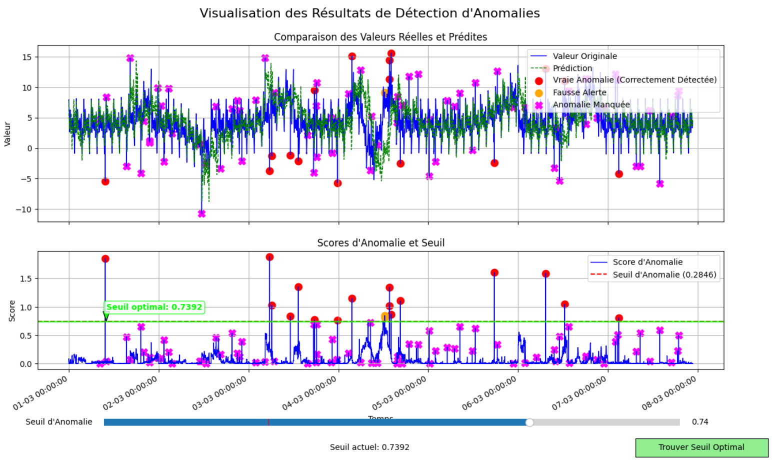


FIGURE 8 – Final results on the signal

5 Conclusion

The project's goal was achieved.
The scores are excellent :
- Accuracy > 99%
- F1 score is low because the drifts are important, but most of the errors are predicted.
- Content of the confusion matrix.

There were still avenues to explore, such as explaining how to select the best possible threshold.