Hackathon Fluvius

Failure prediction based on temperature and humidity sensor data of electrical substations

CARL SOFTWARE Berger Levrault

Table of contents

1. Our understanding of the issue ... 3
1.1. Current system ... 3
1.2. Desired improvements ... 3
1.3. Observations ... 3
1.3.1. Temperature ... 3
1.3.2. Humidity ... 5
1.3.3. Basement measures ... 7
2. Our suggestions ... 8
2.1. Timeseries forecasting to raise alarms ... 8
2.2. Level the data by subtracting the rolling mean ... 10
2.3. Adapt the model with change detection ... 13
2.4. Improve the thresholds with rolling mean ... 15
2.5. Detect sensor failures with multivariate analysis ... 15
3. Outcome of this challenge ... 16
4. Annexes ... 16
4.1. ARIMA/SARIMA ... 16
4.2. Change detection ... 17

CONTACTS

Prénom Nom
Fonction
prenom.nom@berger-levrault.com

Prénom Nom
Fonction
prenom.nom@berger-levrault.com

ADRESSES ET LIENS UTILES
BERGER-LEVRAULT
Siège social :
892 rue Yves Kermen
92100 Boulogne-Billancourt

Centre administratif et opérationnel :
64 rue Jean Rostand
31670 Labège Cedex

www.berger-levrault.com
boutique.berger-levrault.fr

1. Our understanding of the issue

1.1. Current system

Electrical substations are currently equipped with humidity and temperature sensors. Because condensation resulting from high humidity and low temperatures can reduce the lifespan of electrical equipment, it is necessary to detect failures of the heating system and water in the basements.
Currently, this detection is made by a fixed threshold on the temperature and humidity. This can be adjusted manually but there is no intelligence using historical or contextual data to improve the thresholds. Such system is prone to high false positive rates which mean the raise of too many alerts and a trend to ignore them. However, reducing the thresholds to reduce the number of alerts can cause missing failures. In both cases, the alerting system becomes useless.

1.2. Desired improvements

The main objective of the project is to improve the alerting system. It is highly suggested to do this by defining smarter thresholds based on historical data (previous observations) and/or contextual data (weather, type of building, etc.). False positives should be avoided, or at least detected to cancel the alert.
The wanted delivery is not a complete solution but a list of processing steps on data to make the threshold smarter.

1.3. Observations

We have been able to detect various kind of phenomena on the data, depending on the sensor. These phenomena will be listed below. The figures are extracted from data from the site B00061d77COMP.

1.3.1. Temperature

- **Periodic peaks**

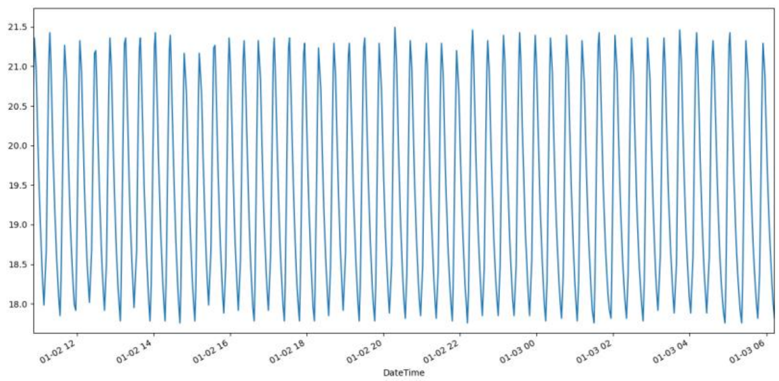

Figure 1 - Periodic peaks in temperature from the regular functioning of the heating system

These periodic peaks seem to math the regular functioning of the heating system, which start when the temperature reaches a lower threshold and stop when it reaches a higher threshold. The periodicity in the peaks seem constant, but it can vary over time.

- **Periodicity changes**

The periodicity in the peaks seem constant on the previous figure, but it can vary over time which make the signal difficult to forecast.

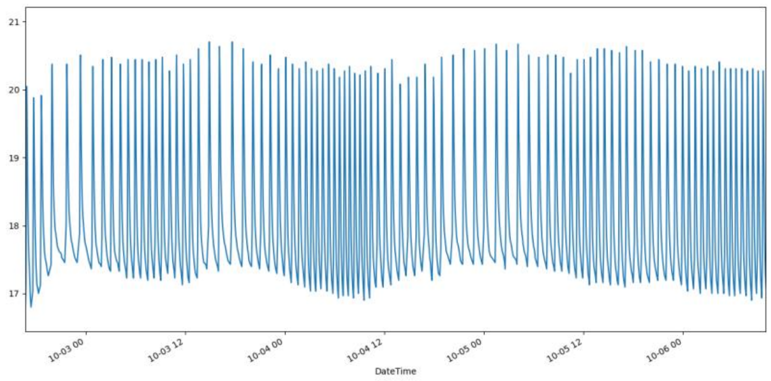

Figure 2 - Variations of the periodicity of the peaks

■ Variations in amplitude

The Figure 2 also show variations in amplitude for the signal. These variations seem to be daily and should not have a great impact on the threshold. However, more important variations can appear over several months as shown on the following figure. We are not able to say for sure if this is a normal behavior of the heating system during winter or a failure of the system that need to be detected. However, in the suggestions that we will make, a way to raise an alert when it occurs will be proposed.

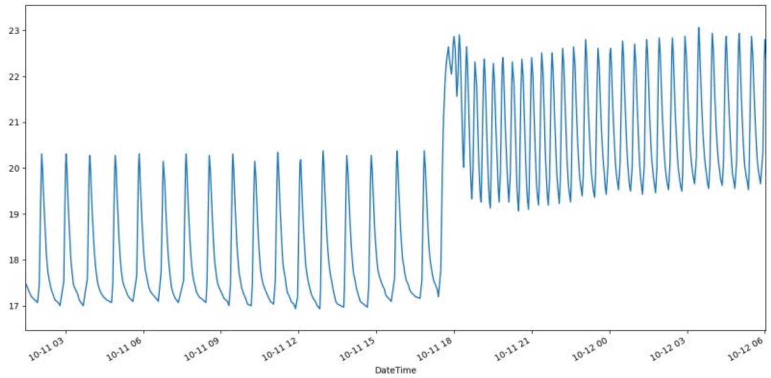

Figure 3 - Beginning of the change in amplitude

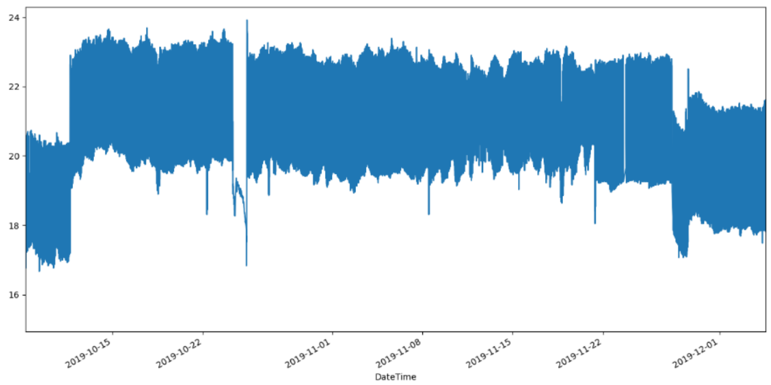

Figure 4 - Change in the amplitude from October to December

■ Heating system failure

In the Figure 4, we can clearly see a failure of the heating system around the end of October. Because of the high temperature during this period, the failure would not be detected by manual thresholds. Another failure, easier to detect, can be seen in the Figure 5.

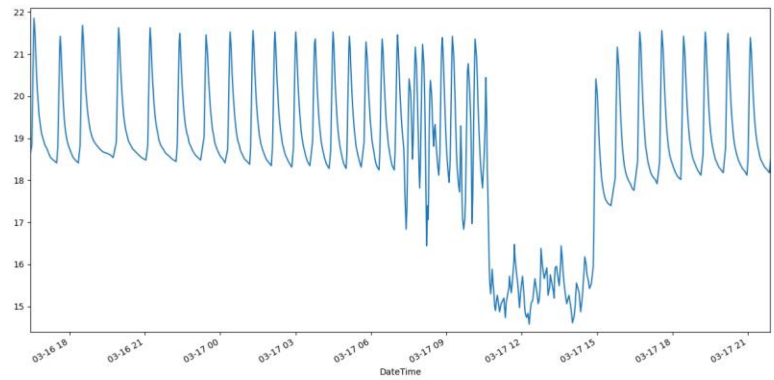

Figure 5 - Failure of the heating system

■ Missing values

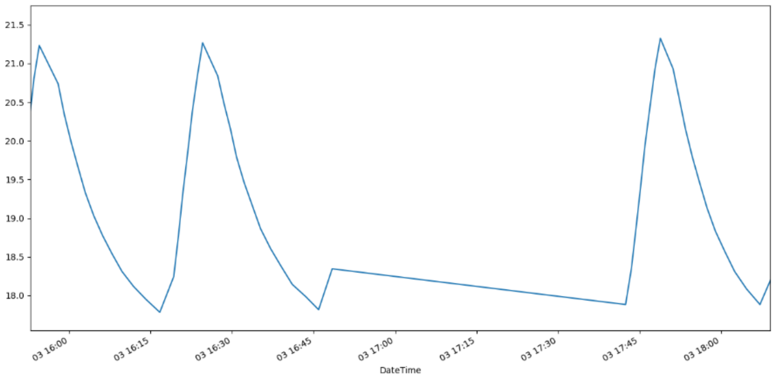

Figure 6 - Missing values for almost one hour

Missing values is a usual issue with sensor data which can be detected as an anomaly when it breaks the continuity of the signal. One missing value alone is not significant enough, but it can be critical when it lasts for minutes, hours or days.

1.3.2. Humidity

■ Periodic peaks (linked to the heating system)

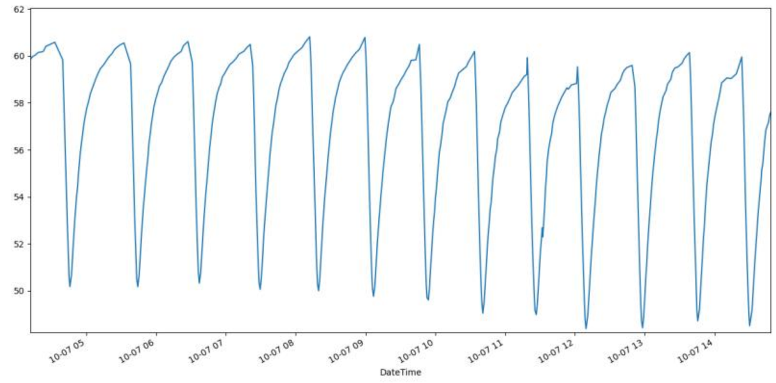

Figure 7 - Periodic peaks in humidity, matching the inverse of those for temperature

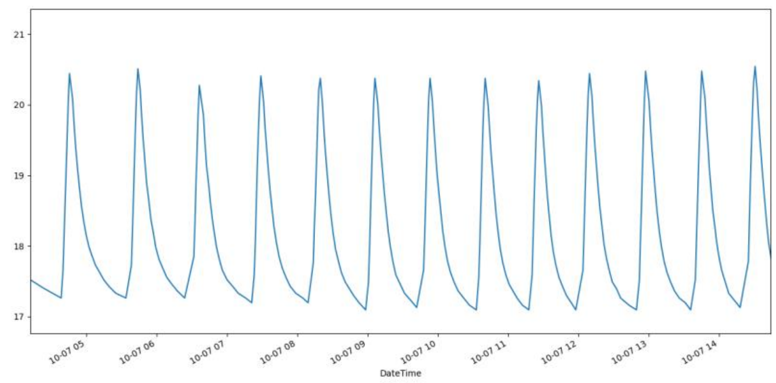

Figure 8 - Temperature with timestamps matching those in Figure 7

There is a clear correlation between peaks in temperature and humidity, caused by the heating system.

■ Variations in amplitude

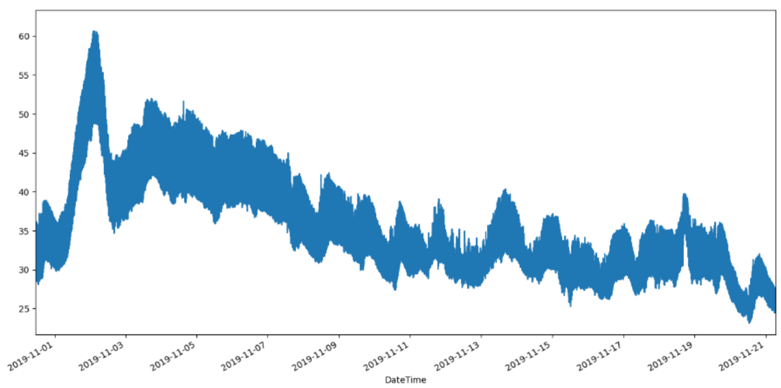

Figure 9 - Variations in humidity, far more noticeable than the ones for temperature

The major difference between humidity and temperature is that humidity carries a component that greatly vary between days, influenced by the weather.

■ **Heating system failures**

The failures of the heating system can also be detected through the analyse of the humidity because of the disappearance of the peaks. However, the impact on amplitude is less visible.

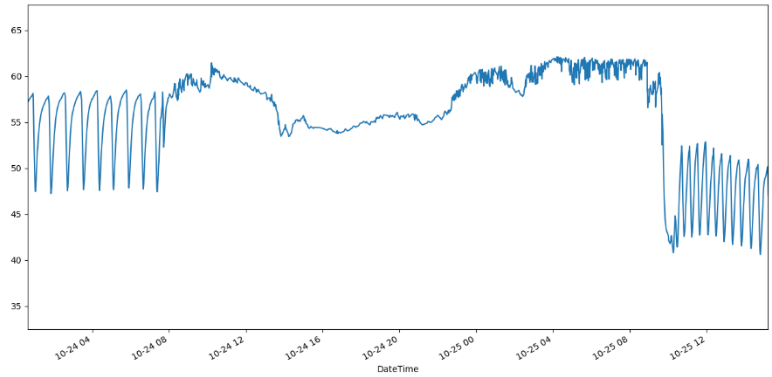

Figure 10 - First failure in the end of October

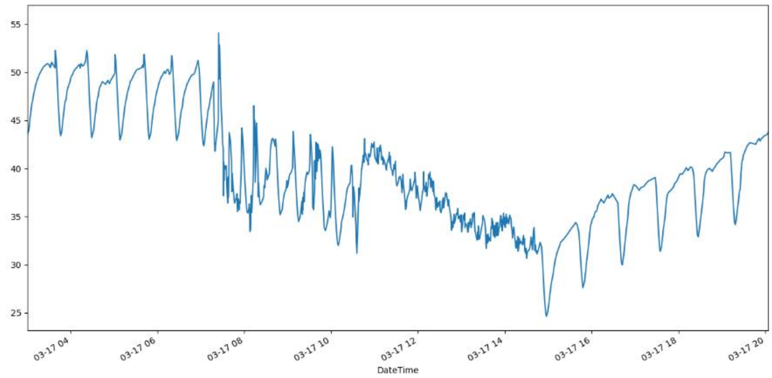

Figure 11 - Second failure in March

1.3.3. Basement measures

The measures of temperature and humidity in the basements under the substations is sometimes available. It is particularly of interest for the humidity because there is a correlation between the variations in amplitude. However, the heating system has no impact on the signals in the basement. In the Figure 12, the variations in amplitude from the Figure 9 can be retrieved.

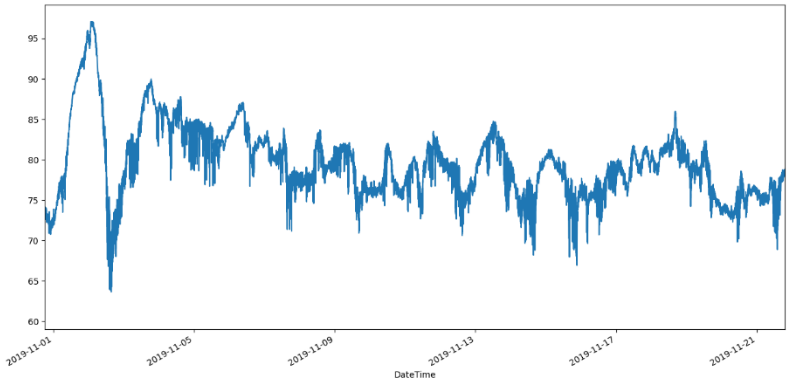

Figure 12 - Variations in amplitude for humidity in the basement, similar to those from the humidity in the station

Hence, to our understanding, the measures of the temperature and humidity in the station should be the ones to focus on for alarm raising while the ones in the basement should, when available, be used as contextual measures.

2. Our suggestions

Because the wanted delivery is a list of steps to improve alarm raising, we will submit in this second part a list of suggestions to better approach the data. Some visual results with the same data than in the observations part will be given in order to help figure the intuitions. **However, every suggestion has not be tested yet.**
Rather than focusing on the computation of a dynamic threshold, we focused on improving the alerting system to raise smarter alarms through several means.

2.1. Timeseries forecasting to raise alarms

When we observe the measures of the temperature in the station, it appears that there is no obvious trend in the data and a periodicity partly consistent. Hence, we think that a timeseries forecasting model which take periodicity into account such as SARIMA could be able to predict the next value at each time with a confidence interval. When the next value does not lie in the confidence interval, it gives a precious information about the current state of the heating system. Also, such a model will not be affected by missing data if it is precise enough because it will be able to predict few steps in the future (with a decrease in precision at each step because it relies on previous predictions, not on real observations).
However, if the periodicity is not consistent enough, and because we just want to predict the next value, a simple ARIMA model will give as good results. Figure 13 and Figure 14 present some results with an even more simple ARMA(3, 5) model *(the choice of the parameters will be discuss in the annexes with a description of ARMA, ARIMA and SARIMA models and some useful links to learn more)*.

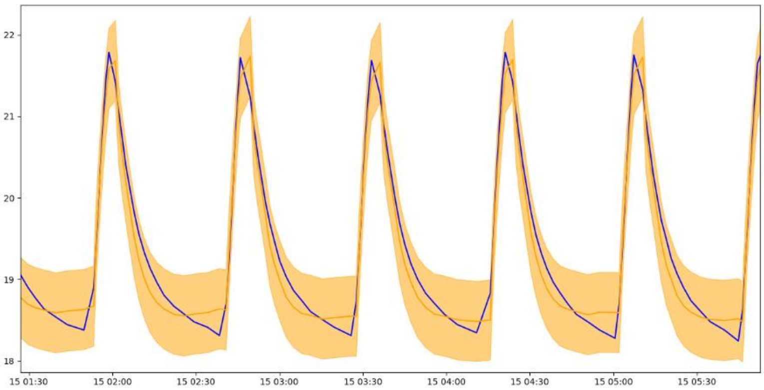

Figure 13 - Predictions (orange) of the temperature (blue) from ARMA(3, 5), with a confidence interval (orange area)

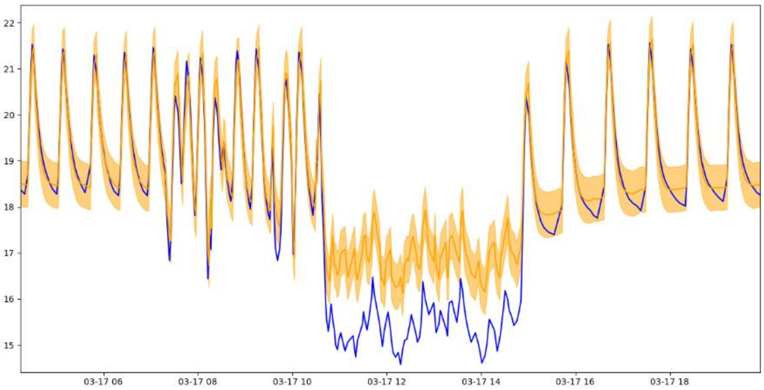

Figure 14 - Predictions for a failure

However, the model does not detect the lack of periodicity in the Figure 14. This is due to the fact that seasonality was not taken into account. In the end, this model is good in amplitude only. It was trained from the third level of amplitude in the data and gives poor results of prediction for the two other levels (Figure 15 and Figure 16).

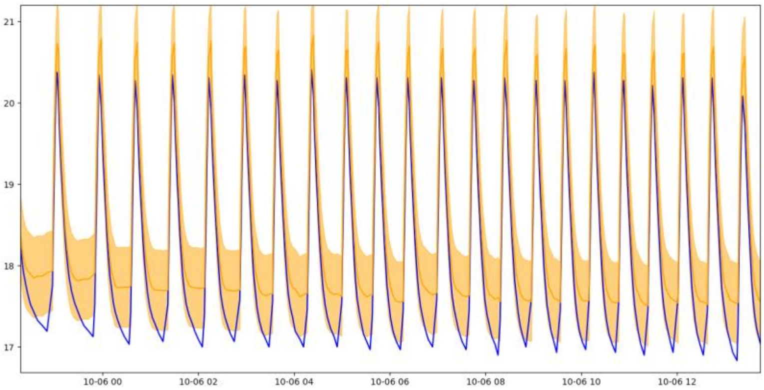

Figure 15 - The first level is lower than the third, the predictions are hence higher than the truth

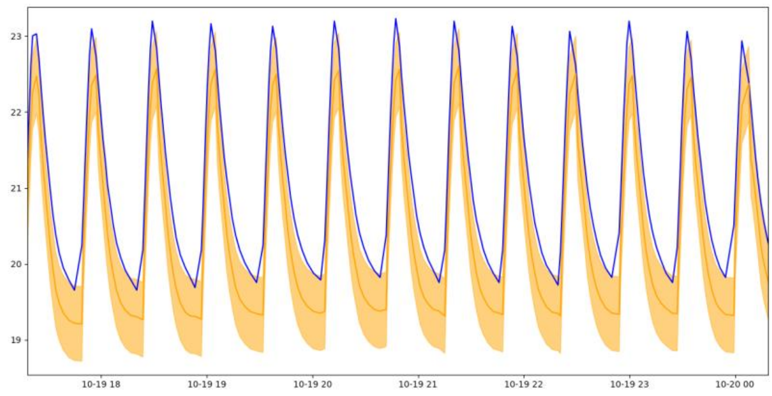

Figure 16 - The second level is higher than the second, the predictions are hence lower than the truth

2.2. Level the data by subtracting the rolling mean

A first solution to avoid the prediction error issue due to different levels would be to level the data by subtracting the rolling mean. For each data instance, we take a sliding window of a fixed time length (e.g. 6 hours, could be more or less; if a lot of peaks are taken in each window, the curve will be smooth and we will keep the behaviour of the timeseries after subtracting; if the time window is too wide, we will lose the break in level and it will appear in the final timeseries, see Figure 18) of data and we compute the mean. The result is the rolling mean that we subtract to the initial data to make it equally levelled (see Figure 17).

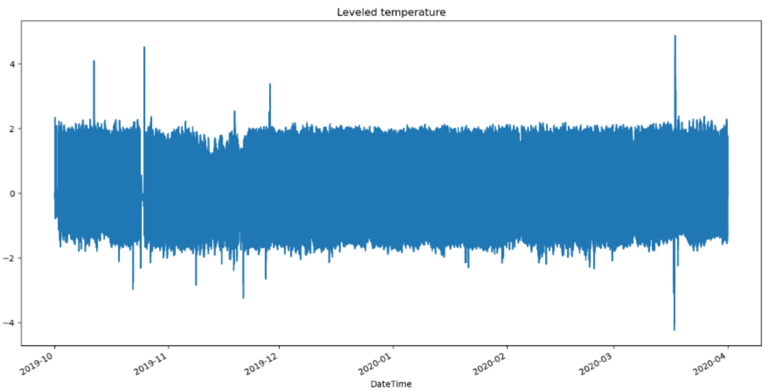

Figure 17 - Levelled temperature, after removing of the rolling mean

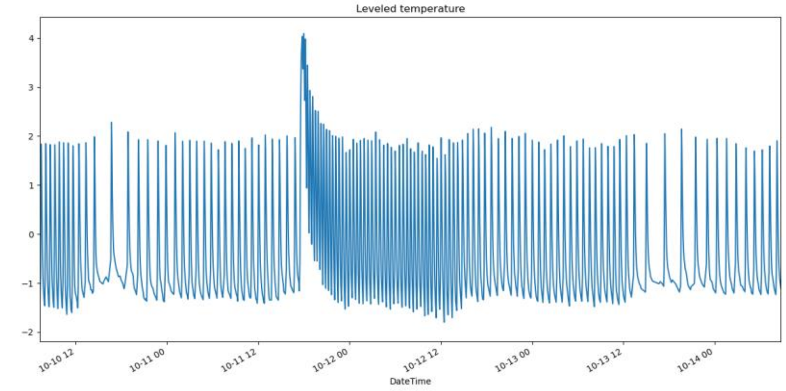

Figure 18 - Visible level change on the levelled temperature, 6 hours could be too wide

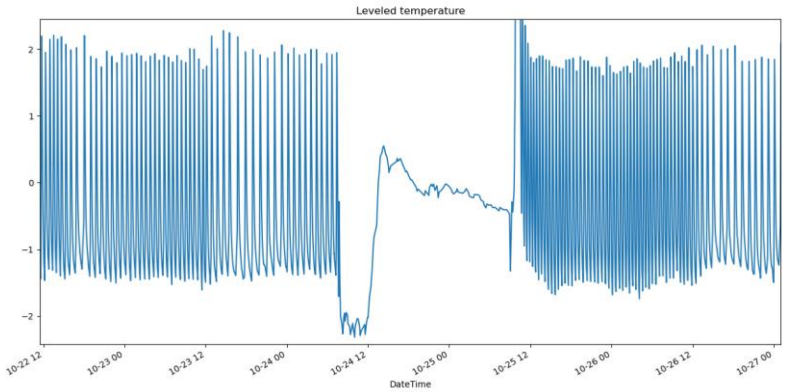

Figure 19 - Effect of a failure on the levelled temperature

The adjustment of the time window should be made according to the desired behavior of the alerting system. If we want to raise and alarm when a change in level appears, 6 hours seem to be a good value for this dataset according to Figure 18. However, the second approach in the next subsection could be of greater interest in the case we do not want to raise alarms on level change. Also, if it is important to raise alarms on strange short peaks, the window size should not be too low.

The Figure 19 shows the effect of a failure on the levelled temperature. The seasonality would also be necessary in this case to perfectly detect the failure. However, and because the time window is wide enough, values at the beginning (and the end) of the failure are sufficiently low (respectively high) to go out of prediction if we train the ARMA model, as shown in Figure 20 and Figure 21.

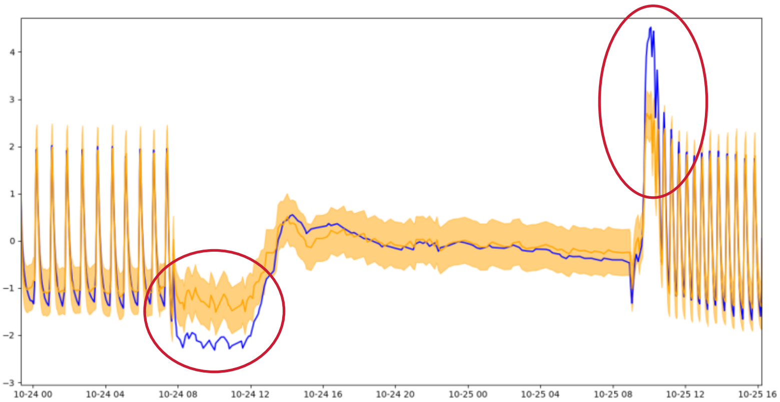

Figure 20 - Detection of the first failure with levelled temperature

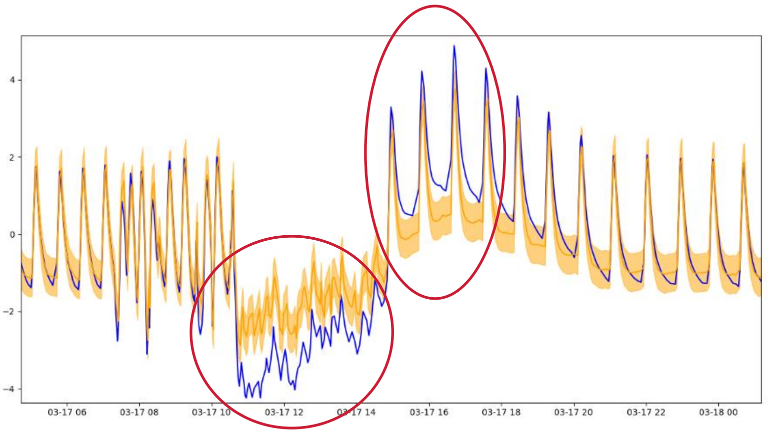

Figure 21 - Detection of the second failure with levelled temperature

The following figure shows that the level change would also be detected as anomalous and raise an alarm.

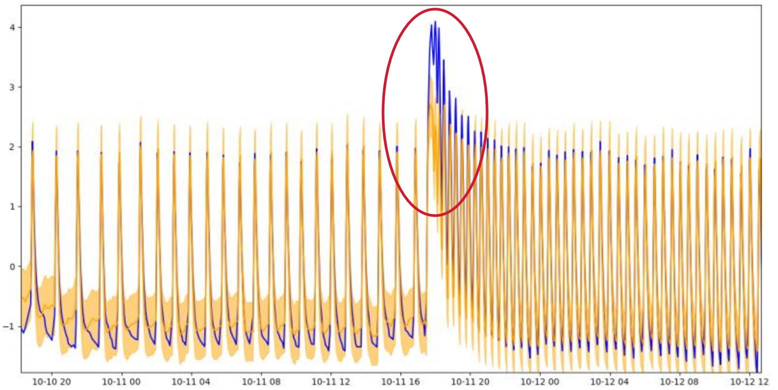

Figure 22 - Detection of level change as anomalous with a time window of 6 hours

It is also important to note that the results of this subsection would be similar with the use of the humidity. Indeed, the raw humidity would have been difficult to predict due to its variations in amplitude. The use of the humidity in the basement and maybe external weather data would have been required to limit these variations. However, the subtraction of the rolling mean do a similar job to level the data (see Figure 23).

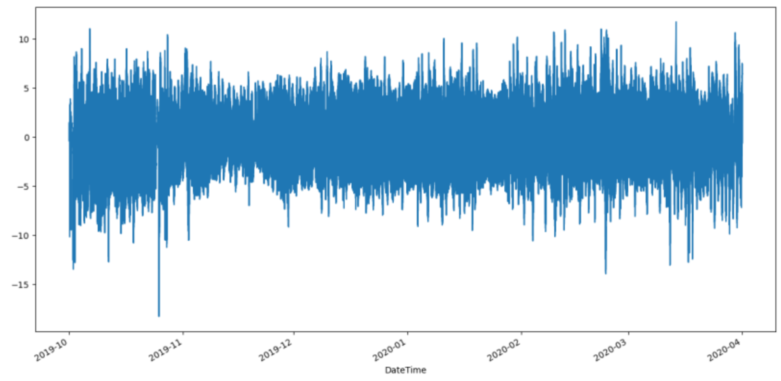

Figure 23 - Levelled humidity

2.3. Adapt the model with change detection

To solve the issue of level changing, we first tried to make all the data on the same level. A second solution would be to learn a new model each time a change occurs. This requires detecting the changes which we can do with a method called change detection (that will be described in the annexes with a referenced book to learn more details and methods).

The advantage of this method is that if we know when a change occurs, we can avoid sending an alert for it. However, it can be difficult to parameterize depending on the data and there is an important risk for the failures to be detected as changes. We did not have the time to test the change detection on the provided data, but we can say by experience that if we want to apply the change detection on the rolling mean with 6-hour windows previously used, the failures will clearly appear as changes (see Figure 24).

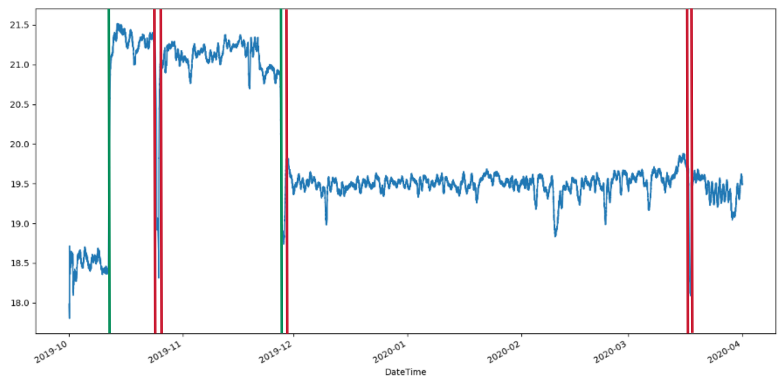

Figure 24 - Wanted results (green) and expected but unwanted results (red) of change detection

To avoid this, it will be necessary to increase the length of the rolling mean time window (which will make the curve smoother) and decrease the sensibility of the change detection. That is why this solution may prove to be difficult to parameterize for different substations.

While it can prove to be even more difficult to use, we recommend considering to apply change detection on a rolling cross mean, similar to the rolling mean in its computation but with the use of the cross mean (how many time the data cross the mean of the window in each window). The rolling cross mean gives the evolution of the periodicity of the data. Hence, in cases where the periodicity is stable on sufficiently wide sections, it could be interesting to compute a new model for each change in periodicity that take seasonality into account to get better predictions.

The following figures show the rolling cross mean applied on the temperature. Even if we can observe different levels on periods (Figure 26 and Figure 27), the great variability of the cross mean would make the change detection unusable Figure 25. Also, it is interesting to notice that the first failure is clearly visible on Figure 28 while the second is far less noticeable (see Figure 29).

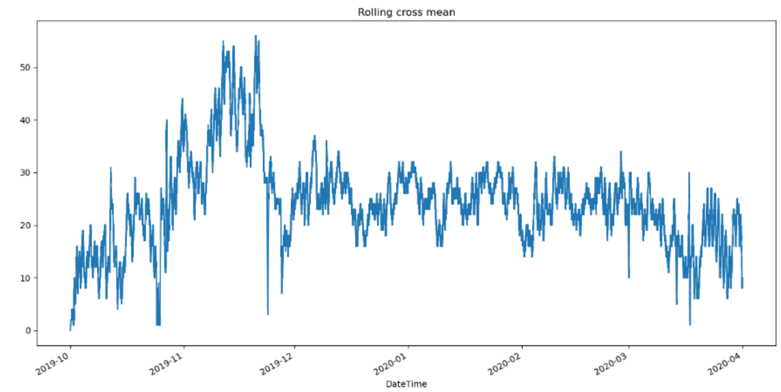

Figure 25 - Rolling cross mean of the temperature

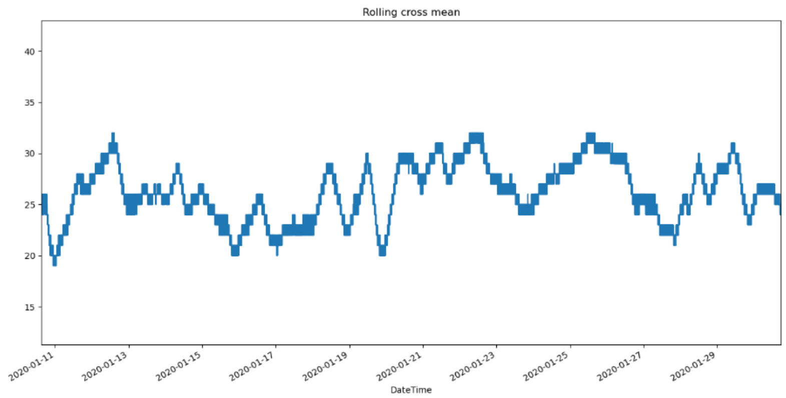

Figure 26 - Rolling cross mean between 20 and 30 for almost 19 days

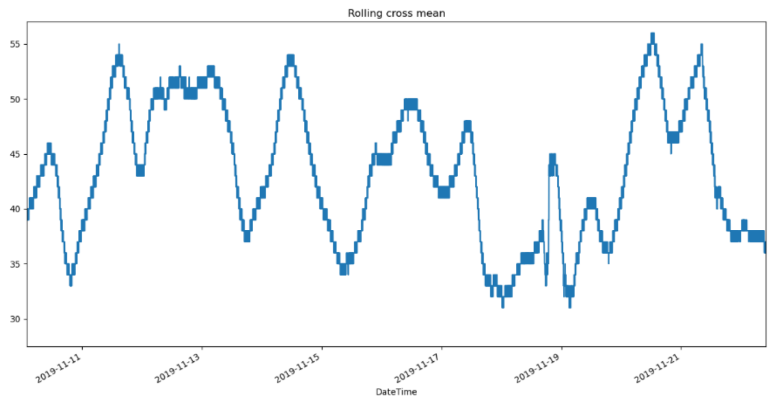

Figure 27 - Rolling cross mean between 35 and 55 for more than 10 days

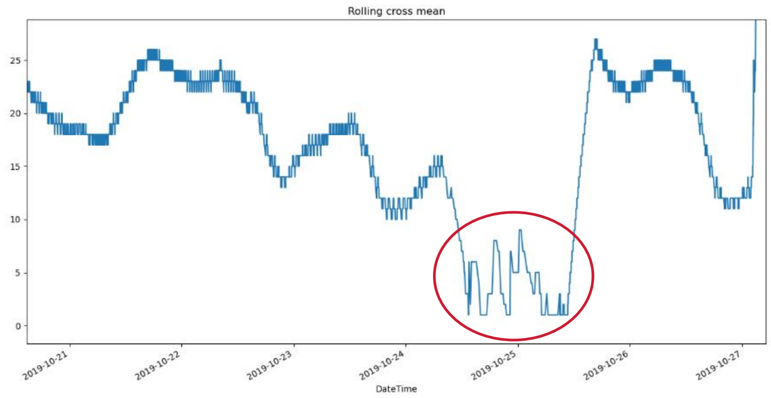

Figure 28 - Rolling cross mean around the first failure (red)

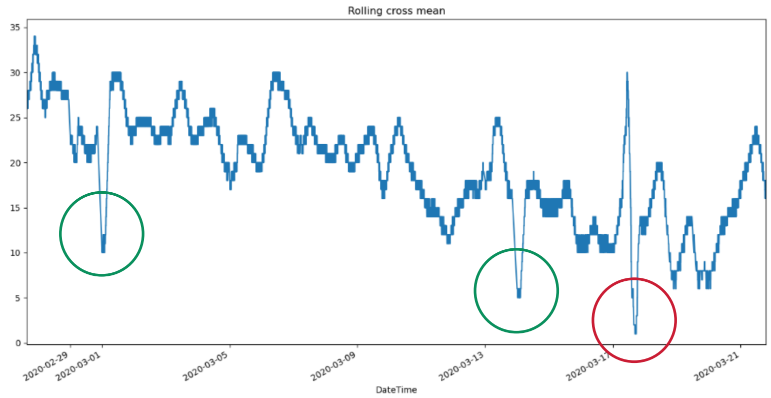

Figure 29 - Rolling cross mean around the second failure (green), similar observations appear with normal values (green)

2.4. Improve the thresholds with rolling mean

Until this point, we suggest solutions to detect failures while keeping in mind the idea of avoiding false positives. The use of a rolling mean can nonetheless be used to compute dynamic thresholds. This way, depending on the size of the chosen window, only high peaks will be able to raise alarms and not long changes in the amplitude. It can be useful to isolate potential absurd values as warning sign of a faulty sensor. Meanwhile, the alerting system based on timeseries predictions could be limited to the case where a sequence of observations is abnormal (out of prediction). This will help reduce even more the false positive rate due to an unprecise model.
The Figure 17 showed that peaks were still visible when subtracting the rolling mean when the changes in amplitude occur. It means that the use of the rolling mean as a base to the dynamic thresholds will still lead to false positives when the level changes rashly compared to the time length of the rolling mean windows. To avoid this, a solution could be to combine the change detection with the rolling mean. Considering the data between two changes as a same level and computing the mean on this whole level, we can obtain a baseline from which the thresholds can be computed.

2.5. Detect sensor failures with multivariate analysis

We saw that the measures were correlated:

- the temperature and the humidity in the substations are correlated because of the effect of the warming system,

- the same measures in the substation and in the basement are correlated, particularly for humidity,
- it is highly possible that a correlation exists between the temperature/humidity in the station and outside.

Hence, for each measure, the other measures can be used to improve the predictions. A linear regression should be learned for each measure, and the residuals can be used to detect anomalous values. This method would be of high interest in the detection of faulty sensors. Indeed, if it is possible to correctly predict the value of a measure from others, then when the prediction is far from the observation it means that a sensor probably returned a faulty value; a system failure would affect all the kind of measures, not only one, while a sensor failure only affects its own measures, not the others. This requires crossing the results with those from the other methods.

The main issue with this method is the mismatching between the sampling strategies for the different measures. To correctly use the correlations between these measures, it will be necessary to aggregate them to a same sampling with the same timestamps. This could lead to a loss of valuable information.

3. Outcome of this challenge

Unfortunately, as we meet the deadline of this challenge, we did not take the time to explore every possibility we suggested. We chose to answer this challenge with a list of suggestions, but we mainly focused on the use of ARIMA models to correctly predict the given timeseries. However, our experiences helped us suggesting other possibilities to explore with the aim of improving the alerting system.

In our opinion, the several suggestions made should be used altogether and adapted to each substation. The objective is for the users to receive alarms from different sources with the ability to:

- disable the ones that are not useful,
- give priority to the more interesting ones,
- cross them to generate new ones (e.g. ALARM_3 = ALARM_1 AND ALARM_2 or ALARM_6 = ALARM_4 NOR (ALARM_5 AND ALARM_3)),
- adjust the parameters of their thresholds according to the trust in the other alarms.

If it is required by the Fluvius company, we would gladly send them datasets containing the shown examples. It would also be possible to develop a data visualisation letting the users play with the data and the solutions to experiment in a playground.

Regarding suggestion on the current analysis strategy, we do not see the necessity to change the current daily reporting scheme. However, as mentioned in the part 2.5., it would be more convenient to have the same sampling strategy (in other words, a same period between measures) for all sensors to make a better use of correlated variables.

4. Annexes

4.1. ARIMA/SARIMA

ARIMA and SARIMA models are adaptations of ARMA models. The objectives of these last ones are to make timeseries forecasting (predict future observe with the sole knowledge of the past observations). It is close to linear regression with the use of past data as predictors.

An ARMA model is composed of two parts. The first one is its AR (Auto-Regressive) component, which definite a relation between an observation at the current time and the ones at previous times. The second one is its MA (Moving Average) component, which definite a relation between an observation at the current time and the residual errors from the current and some previous timesteps. It is important to note that, theoretically, a process must be stationary to be able to be strictly described by an ARMA model.

Formal definitions:

A process X_t has an $AR(p)$ representation (auto-regressive representation of order p) if the following holds true:

$$X_t - \phi_1 X_{t-1} - \phi_2 X_{t-2} - \cdots - \phi_p X_{t-p} = \varepsilon_t$$

where ε_t is a white noise (as well as the residuals of the regression) and ϕ_i are real coefficients with $\phi_p \neq 0$.

A process X_t has an $MA(q)$ representation (moving average representation of order q) if the following holds true:

$$X_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

where θ_i are real coefficients with $\theta_q \neq 0$.

Hence, an $ARMA(p, q)$ process X_t holds:

$$X_t = \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

However, the stationarity of the process is difficult to obtain because timeseries has non-stationary components (trend and seasonality). It still exists ways to transform the process until stationarity can be accepted.
The first one is to differentiate the timeseries in order to remove the trend. The first order of differentiating apply the operator $1 - L$ to the process, the second order the operator$(1 - L)^2$and the d-th order the operator $(1 - L)^d$, with L the operator defined by $L(X_t) = X_{t-1}$ $((1 - L)(X_t) = X_t - X_{t-1})$. If $Y_t = (1 - L)^d (X_t)$ is an $ARMA(p, q)$ process, then X_t is an $ARIMA(p,\ d,\ q)$ process.
The second one is to also take the seasonality into account with a $SARIMA(p,\ d,\ q) \times (P,\ D,\ Q)_S$ process where the extracted process of the $Y_t = X_{i+s^*t}, i \in [\![0,\ s-1]\!]$ is an $ARIMA(P, D, Q)$ process.

Explanation of SARIMA processes:
https://www.youtube.com/watch?v=YIoBRDueHKo
Even if it is written for Python users, the following guide is useful to understand how to parameterize an ARIMA model:
https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

4.2. Change detection

Abrupt changes detection regroups statistical methods designed to help detecting changes in the process that generate the observations. Basically, these methods use successive hypothesis tests on rolling windows to check if the likelihood of the alternative hypothesis (i.e. the data is generated by a new statistical distribution) is greater than the null hypothesis (i.e. the data is generated by the same statistical distribution) by a fixed threshold.
In its simplest application, the abrupt changes detection makes the hypothesis that the data is generated by a gaussian distribution of fixed standard deviation with the changes appearing only in the mean. In the studied cases, this method would probably be enough. If it does not give good results, you would probably want to simply make the changes also appear in the standard deviation, which is not much more difficult.

Download the pdf of the book Detection of Abrupt Changes: Theory and Application:
ftp://ftp.irisa.fr/local/as/mb/k11.pdf