SAPIENZA
UNIVERSITÀ DI ROMA

Water Leakage Analysis and Forecasting for Anomaly Detection using Smart Grids

Information Engineering, Informatics and Statistics (i3s) Faculty

Corso di Laurea Magistrale in Data Science

Candidate

Alessandra Anna Griesi

ID number 1578970

Thesis Advisor

Prof. Ioannis Chatzigiannakis

Academic Year 2020/2021

Thesis not yet defended

Water Leakage Analysis and Forecasting for Anomaly Detection using Smart Grids

Master's thesis. Sapienza – University of Rome

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: griesi.1578970@studenti.uniroma1.it

Abstract

Cloud computing makes possible to measure, to keep track and to store the water information provided by the smart water grids.

Nowadays minimize the water leakages in the grid is an important task to consider, due to the ecologic emergency we are facing.

The first approach was to calculate the minimum night flow of the water. It will lead to the identification of atypical behaviors in the water system considered.

The aim of this research is to compare unsupervised learning algorithms such as Random Cut Forest (RCF) and Isolation Forest (IF) that can help in finding these behaviors. Once implemented, these algorithms work well in real-time setting. The RCF exploits the random sub-sampling of the data and averages the results for building the usual behavior of the data, "cutting" off the anomalies assigning a score per each data point. The IF instead, construct a bunch of Isolation Trees and, for each of them, it computes the average length for reaching a general node. Those nodes who don't reach the average path length are classified as anomalies. IF provides a fast detection of anomalies without implying too many computational performance; this reduces the time reaction of the pipeline reparation and the general costs of water supply. In this setting, we studied all the possible cases, concluding that two of these algorithms combined together can outcome in the best solution, reaching better performance and speed.

Contents

Chapter 1

Introduction

This work of thesis wants to give the insight about the water consumption and the methodologies for anomaly detection that can arise in the water viaducts. Smart water grids are the most modern tool for carrying out the monitoring of the flows. Thanks to data analysis and machine learning algorithms, it is possible to detect water leakages and save water. We live in a critical age, where the water is considered to be quoted in Wall Street [1][2] as oil or gold. The amount of water lost in the world through Water Distribution Systems (WDSs) [3] varies between 15% and 50% of the water produced [4]. Water loss causes environmental and social-economic costs and could have serious impacts on urban infrastructures.

Thus, this concept of smart water grids it's a step for moving towards sustainability. At the same time, these technologies can also help to save money.

The smart water grids are composed of different sensors which are responsible for measuring the data. The benefit of the Internet of Things (IoT) [5] facilities is that it's possible to gather a massive amount of data in real-time from the sensors of the smart grid, in order to perform techniques for extracting useful information per building.

For detecting leakages, physical procedures are often used (i.e. leak audit procedures): but manual procedures are time-consuming and expensive for huge water grids. In the era of Big Data, it is possible to use the data gathered for finding the water losses in a less intrusive way, in a software-vision.

In this thesis, an Exploratory Data Analysis (EDA) [6] and different learning algorithms have been implemented for tracing the path of identification of the best procedure of forecasting anomalies (the Random Cut Forest, Automated Minimum Night Flow etc.) and we compared them in several aspects such as speed, precision and costs.

In particular, two algorithms compounded together satisfied these goals: the Automated Minimum Night Flow (AMNF) and the Isolation Forest (IF). The Isolation Forest detected the peaks of consumption in our data set, meantime the AMNF detected those consumptions slightly lower the peaks but that were still abnormal quantities.

The AMNF was a key point for move our studies on the night flow, together with the EDA (Figure 4.3).

Leakages need to be detected in a fast way for intervening in proper times and effectively, preventing the false alarms.

The work behind this thesis is structured in 6 sections. Section 2 provides a description Previous Work done in this research field; while section 3 contains the definition of the data set and giving a general view of the Amazon Web Services (AWS)[7], that provides cloud-based services relevant to anomaly detection. Section 4

and 5 show the descriptive plots and the techniques used for the anomalies detection respectively.

1.1 Case study

The city of the interested area is part of the 100 Resilient Cities (100RC) network, being selected as one of the eight cities contributing to the development of the City Water Resilience Approach.

Smart Water Grid Description

The grid is made by 80 Internet of Things devices installed in 25 buildings.

These devices measure different physical variables of the water such as the quantity, the temperature, the pressure.

We will deep the data set description in 3.

Chapter 2

Previous work - Smart water infrastructure

The work presented in this section utilizes the benefits of the fog computing paradigm in terms of processing the data collected from the smart water metering devices[8]. General characteristics of Smart Water Grid are:

- to communicate and keeping information on-cloud
- to track and to understand habits
- to study the water quality

The cloud has to address several requirements: keeping substantial metering data sets and protecting them from suspicious accesses. But the cloud has also a centralized architecture, hence it faces latency issues.

Here it is been developed a new approach of collecting data based on the fog computing (instead of the cloud computing), an alternative technology that exploits resources in the network, allowing speed and optimization in the network itself, guaranteeing a fine-grain data access.

The pros of the fog computing are

- Reduction of bandwidth requirements: the fog computing approach allows a fine-grained access to the relative data from and to devices, where the size of the data is considerably decreased. This helps the balance of the information load between the devices and the cloud.
- Improved responsiveness and reliability: thanks to the first point, the network benefits in terms of speed and delivery of the packets in both the directions (towards the devices and the cloud).
- Improved privacy: the fog architecture enhances the data security because it consists in two additional layers between the devices in the buildings and the cloud. Its nodes are intermediates for providing security and sensor management capabilities, such as deep packet inspection or message encryption and benefiting from context and location-related information to easily detect threats [9].

Fog Architecture Structure

Giving a snapshot of the fog architecture, as first step we have sensing devices and remote controlling smart meters. This is the area called SAL, that stands as Sensors and Actuators layer.

Before continuing, we have to assume that all the steps of passing information from a layer to another are encrypted.

Through the buses, the information passes to the Mox nodes, where a first authentication is performed. The Mox nodes are located in the second layer of the architecture, the sub layer-A. The mox nodes also execute data analysis and prioritize sendings to the following layer.

The mox nodes pass the information to the next sub processing layer-B to the LoRa gateways, that lead the information to the Tergo nodes. These nodes are more powerful than the Mox nodes, and they are responsible of checking on the packet rates, to assure the data privacy, to store the data and to communicate with the cloud. We talk about Tethys cloud, that is the Cloud Based Layer.

The Tethys cloud is the centralization of our data, data are stored after the processing in the sub layer A and sub layer B, the so-called Edge-based processing layers. The processing is based on filtering and cleaning the data from the two main issues: malicious packets and big loads of information.

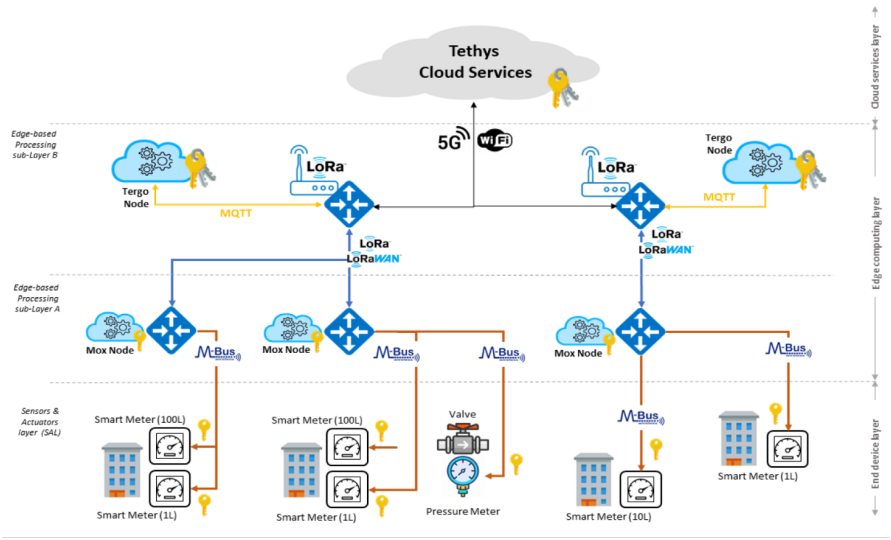


Figure 2.1. The fog computing-based data hierarchy

Anomalies in water infrastructure

The saved data sets represent the historical of the story of the metering devices. Based on these data we can define what are the consumption patterns, thus it is possible to detect the anomalous water usages.

Malfunctions Alerting

Due to the continuous data exchanging between valves (characteristic of this technology), it is possible the monitoring and the alerting of those concerned in the managing of the viaduct for the interested area. The integration between the

conduits of water and applied technologies helps to take action way faster then in the case of human control. With this methodology the reaction time for fixing any leakage or for repairing any pipeline is potentially reduced.

Chapter 3

Material and Methodologies

The procedures are to measure the water flow and to collect all the information about the pressure and temperature of the water, the ambient temperature, and, of course, the amount of water that is present in the pipelines, per hour.

In each building there are sensors to report the:

- volume passing through the pipeline where the sensor is installed;
- temperature of the water
- ambient temperature;
- pressure of the water (to evaluate the speed of the water);
- consumption per hour.

Furthermore we have, together with the physical data obtained from the sensors, for each row in the data set:

- the ID of the device in that building;
- the sensor id: it specifies whenever the sensor is measuring the pressure, the water or ambient temperature, the volume, the water flow or other messages of alert. This can look trivial, but is decisive in order to slice the data set for the right physical variables;
- the unique value, called timestamp, for the sensor that is acting at the specific time, for that sensor ID, in that building;
- the time is dictated by a date time value, but we created columns with:
  - the day,
  - the month,
  - the year per each row of the data set.

This was crucial in order to plot the data by months, or years, or grouping and understanding which consumption could be dictated as unusual, telling us a leakage or a disruption occurred.

3.1 Features

We analysed the data to extract the previous leakage and to forecast those that can occur in the future.

An important fact was to trace the leaks studying the whole night flows per day. This is because the possibility of human water usage is lowered and almost null in the night time. Thus, the losses dictated by a fail or a rupture are easier to see.

We extracted the consumption levels for each building. The sensors can trace the level of water flow per hour.

Thanks to this, we've been able to manage the data set in order to aggregate the data based on these values, taking out the trends of water consumption per building and seeing (plot per months, per week, per day and during the year).

We extracted information about water flow: it's been possible to calculate the correlations between the temperature of the water and the pressure, the drift of the temperature during the year and other inspections (4).

3.2 Detecting outliers and removal

It's crucial to produce effective plots of the data we collected. For this reason, we need to take off the atypical values measured, to a proper sight of the habits concerning the water usage. For example, in the network of the grid there can happen a delay, or some values are not properly registered. In these cases, what could happen? Computing the consumption can end up having negative values, so it's important to erase these rows of the data set. Of course it's not prolific to delete all those atypical values mentioned above: high values of consumption are alerting that a leakage or an unusual event is happening.

At the same time, we need to identify and replace the outliers in our data set.

For this reason, we exploited the Hampel's filter [10]: once set up the window we want to work on, it passes through our time series data and checks if each point it's not "too far" from the median of the window we are working on plus a tolerable interval for keeping the points. In this way, all the data are set up to not deviate too much from the standard behavior of that window, and those who are not suitable in this sense are deleted.

This algorithm applies the Median Absolute Deviation, passing through our not-negative data, keeping those values that satisfy the general behavior of the slot (window) considered by the algorithm in that moment, following the condition of

$$\|\text{input window} - \text{median(input windows)}\| \geq n_\sigma \times k \times median(\text{input window and median window})$$

where, **k** is the scale factor of a Gaussian distribution, based on the notion of normal Gaussian distribution, thus we assume that the general comportment of our water consumption is almost normal (4.3); σ is the value of the standard deviation and both are fixed for every windows; n_σ is how many times the standard deviation can keep the values in the window.

3.3 Amazon Web Services

It was 2006 when Amazon launched what today can be defined as its main product: Amazon Web Service (AWS). AWS is a cloud computing platform that

gathers different kinds of services: from data analysis to virtual reality. With more than 200 services provided in 245 countries, AWS is the widest cloud platform in the world. A business that represents 58% of the revenues of the entire Amazon company. AWS comes out with several features that allow clients to develop sophisticated applications with low effort. For instance, in 2014, AWS launched Amazon Lambda, a developing tool that runs codes without requiring to manage a server. Later on, AWS realised Amazon SageMaker for all the machine learning (ML) experts and not. In fact, SageMaker provides developers both pre-trained and built-in ML algorithms. Thus, this service supports the early developers and, at the same time, the more experienced Data Scientists. Nowadays, more and more companies are choosing AWS to carry out their cloud services. This happens because AWS offers a secure infrastructure, data are monitored 24/7 and are encrypted before being shared. Besides, since it is present all over the world it represents a flexible and dynamic structure adapted to the needs of customers. Furthermore, the main advantage lies in the price. In fact, customers pay only for the service they use and not the entire AWS package.

3.3.1 SageMaker

SageMaker is a fully managed machine learning service. It offers Machine Learning methodologies and structures that allow speed and elasticity, together with an automatic hyper parameters optimization.

The reason why customers use SageMaker, it's because it reduces the running time of the algorithms and the costs of the machine types. SageMaker also provides classic deep learning approach:

- Linear Learner: based on stochastic gradient descent, it provides linear regression and classification models. The characteristic of this algorithm is to support the simultaneous exploration of different objectives by training multiple models in parallel.
- Factorization Machines: It is an extension of the Linear Learner, it can acquire interactions between features of high dimensional data sets.
- K-Means Clustering: K-Means Clustering is an unsupervised algorithm for clustering a dataset into k groups. It combines ideas from stochastic/EM optimization [11, 12], coresets [13, 14] and online facility location.
- Principal Components Analysis (PCA): this unsupervised algorithm use the dimensionality reduction of the features of the dataset, keeping those ones with the highest values of variance. They are intended to be the most important components in the data set. [13]
- Neural Topic Model: unsupervised algorithm that learns latent representations of large collections of discrete data. The algorithm is based on the variational autoencoder, to achieve fast inference compared to classic alternatives, which require iterative computations as in variational inference or Gibbs sampling.
- Time Series Forecasting with Deep AR: is a probabilistic forecasting algorithm based on recurrent neural networks.

[15]

3.3.2 Technical Functionalities of Amazon SageMaker

For the implementation of the classic ML algorithms, Amazon SageMaker offers an environment based on the Jupyter Notebook framework. Then, it is possible to initialize a Jupyter instance for the creation and the test of our project.

Integrated with the Jupyter Instances, Amazon SagemMaker provides also the Jupyter Notebook Lab for building models and also for establishing the connection with a GitHub repository.

3.3.3 Amazon S3

Amazon S3 is another service provided by AWS. Amazon S3 can be viewed as the main storage and backup for cloud applications. It allows to store an unlimited quantity of resources in their S3 console. It is possible to divide the data saved in different areas called buckets. This consents us to have several ongoing projects to work on and to get into the data whenever we want. Furthermore, in each Amazon S3 Bucket, we can create different directories in which to save our tables [16]. The Amazon s3 potentiality is not only collecting data. It also communicates with the other Amazon technologies. In our case, we exploited the **S3 Connection** that let us to read the data from Amazon SageMaker.

Chapter 4

Visualization of the data set

4.1 Visualization of the main features

4.1.1 Consumption trend over the year

The Figure 4.1.1 shows all the consumption computed in the volume dataset during the whole period of measurement, showing an outlier of more than 8000 $\frac{m^3}{h}$.

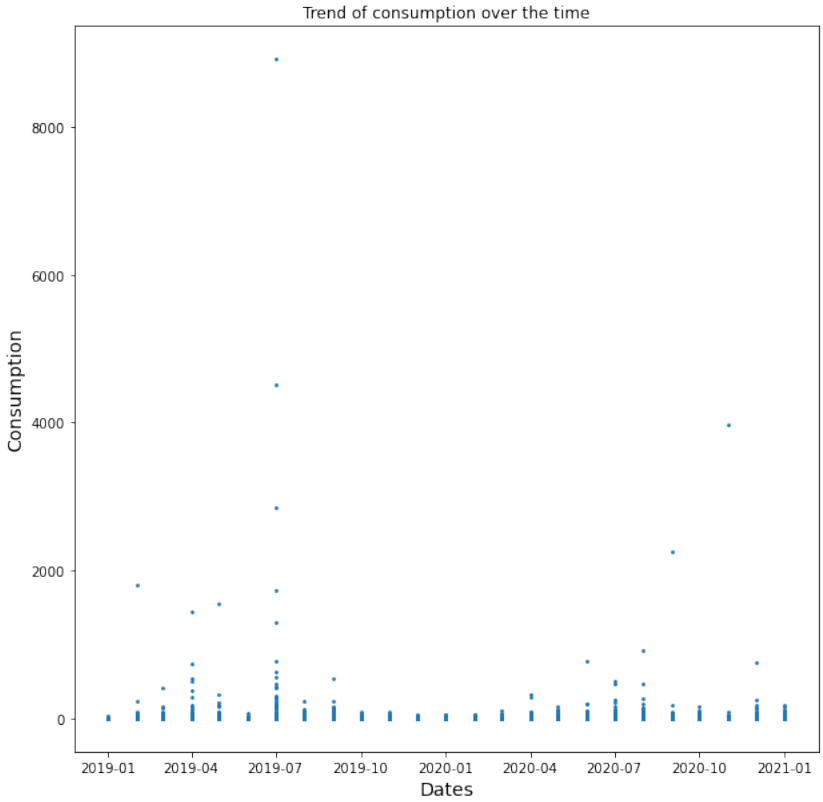


Figure 4.1. Consumption trend over the year

4.1.2 Volume water trend over the year without outliers

The graph shows the data points without considering specific time slots (Figure 4.2). We removed the outliers using the interquartile range, but we cannot clearly identify specific patterns or any shape of known distribution with the water volumes.

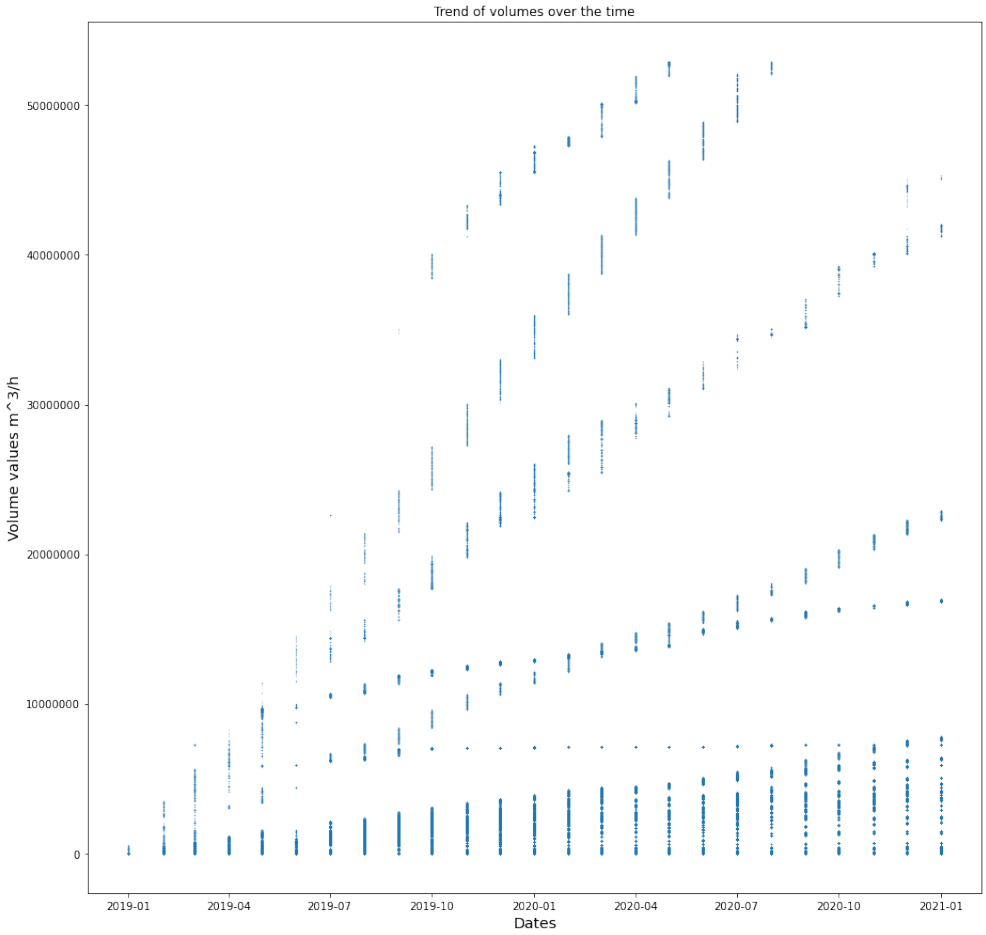


Figure 4.2. Overall Volumes without the outliers

4.1.3 Night consumption trend during the year

In the graph in Figure 4.3, the blue points represent the values of the night consumption divided by month, per all the buildings.

The values of water consumption can be referred to a maximum of 60k m^3/h, the points far away from those that concentrate between 0 and 20k m^3/h, can be seen as a first sign that something is not properly operating.

And this can be noticed most especially in the period of November, July and August, when there was a big loss of water.

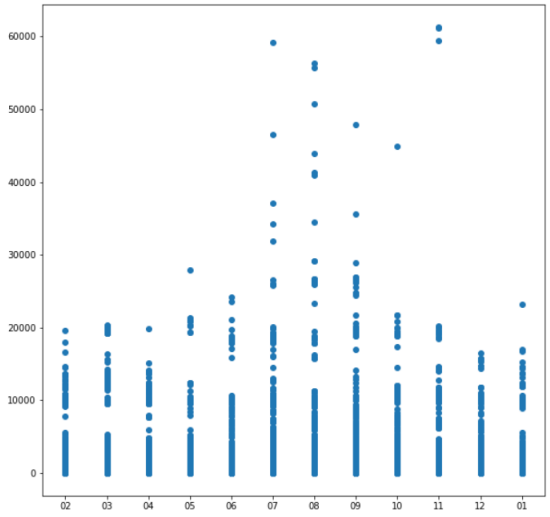


Figure 4.3. Night consumption trend during the year

4.1.4 Maximum water consumption plot per building

Plotting the overall consumption per building, from the beginning of the data collection, shows that who consumes more water is the Building 17, followed by the Building 11. We see there is a big difference between the 17^{th} and the 11^{th} one, telling that happened a huge loss in the first case.

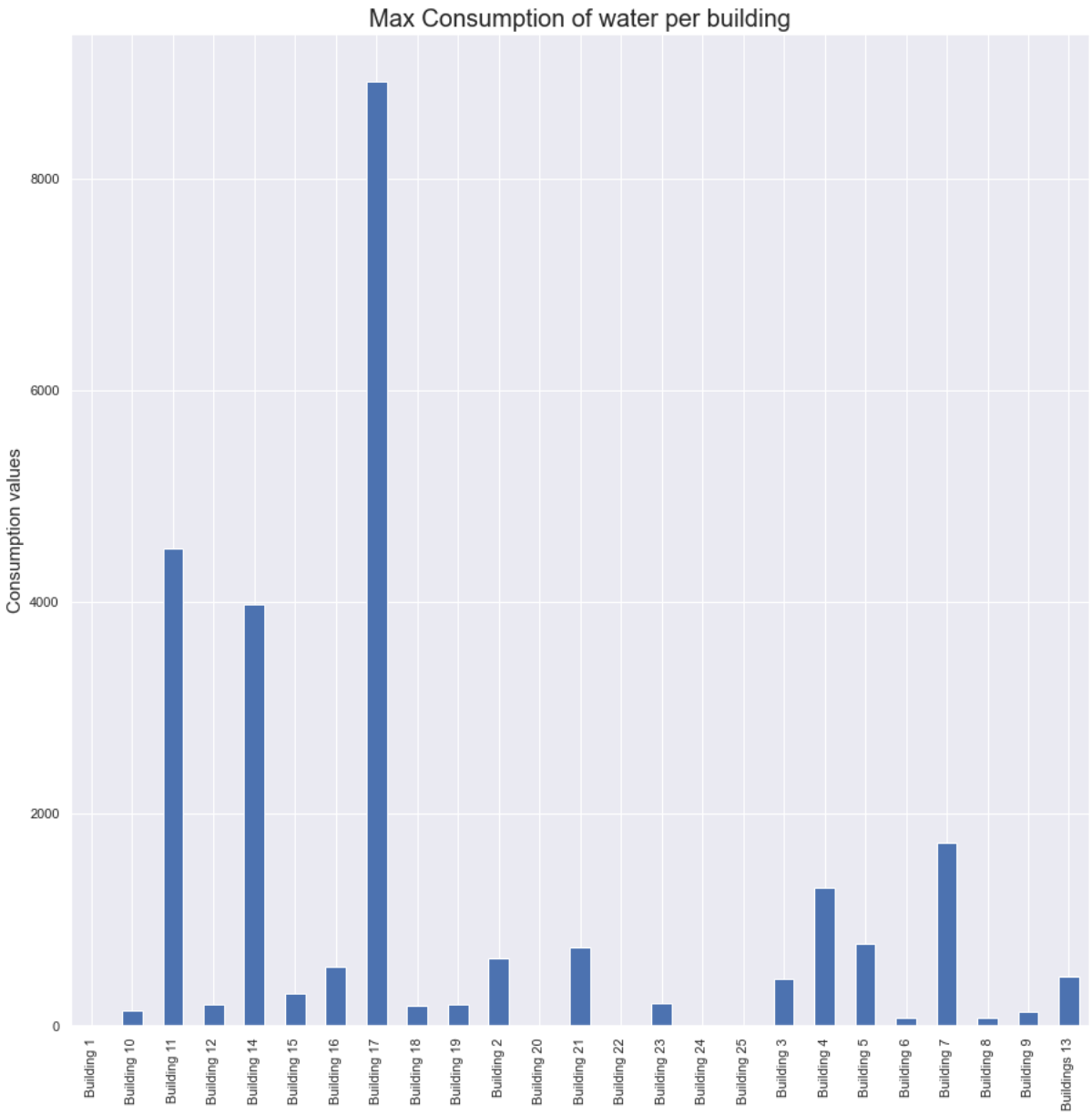


Figure 4.4. Maximum water consumption per building

4.1.5 Temperatures trend

We plotted the mean temperature corresponding to each day of the week (Figure 4.5). In the figure, it is worth mentioning that Sunday and Monday are the days which correspond to the lowest temperature values. This can be addressed to the less frequent use of water during the weekend. The lack of usage of water, especially of hot water for humane usage in the buildings, can lead to a water cooling in the pipelines. The temperatures are meant in Celsius degree.

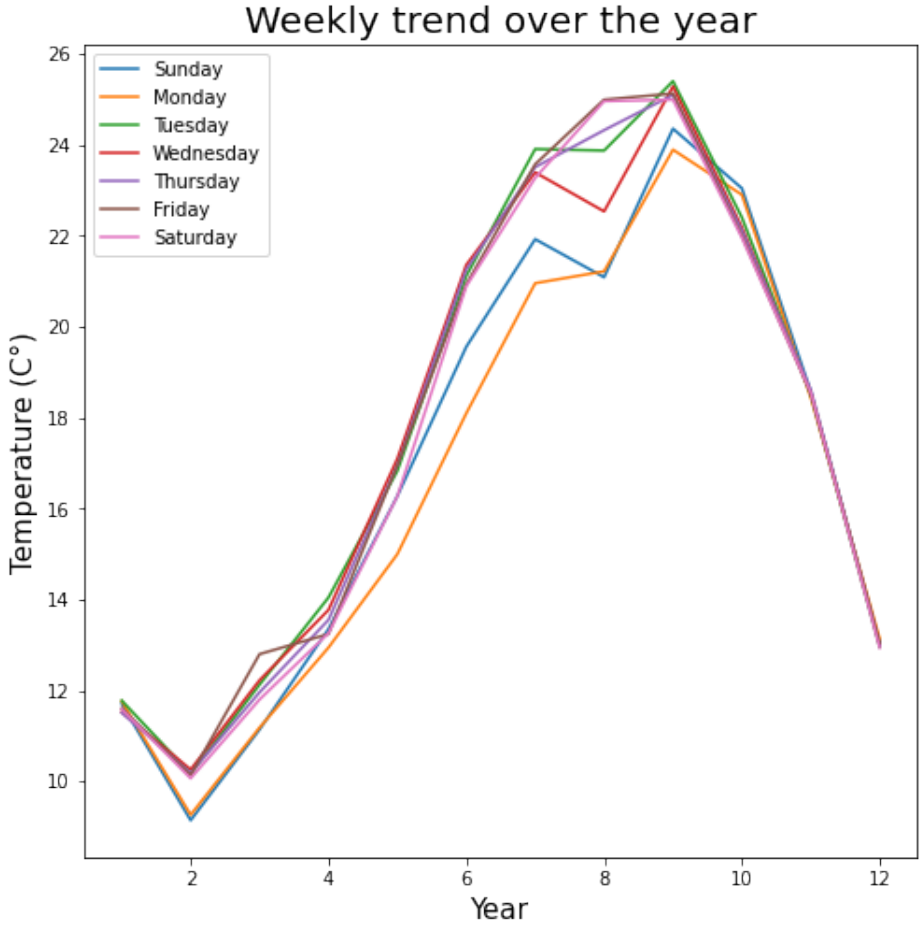


Figure 4.5. Temperature trends divided by weekdays

In the following plot we can see the average of water temperature during the year, compared with the maximum and minimum ambient temperatures for each month. During summer, the water temperature is lower in comparison with the ambient one: it seems that the water in the tubes are not reached and heated by the sun.

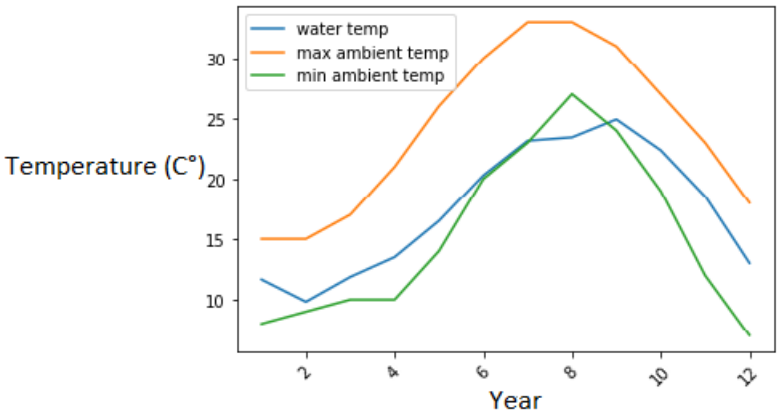


Figure 4.6. Water and ambient temperature over the year

4.1.6 Average flow over the day (24 hours)

We observed that the flow in the night is inferior in comparison to the day flow, it is visible in the graph from the shift between the 6 am and the next hours. The plot is to be intended as the behavior of all the buildings together.

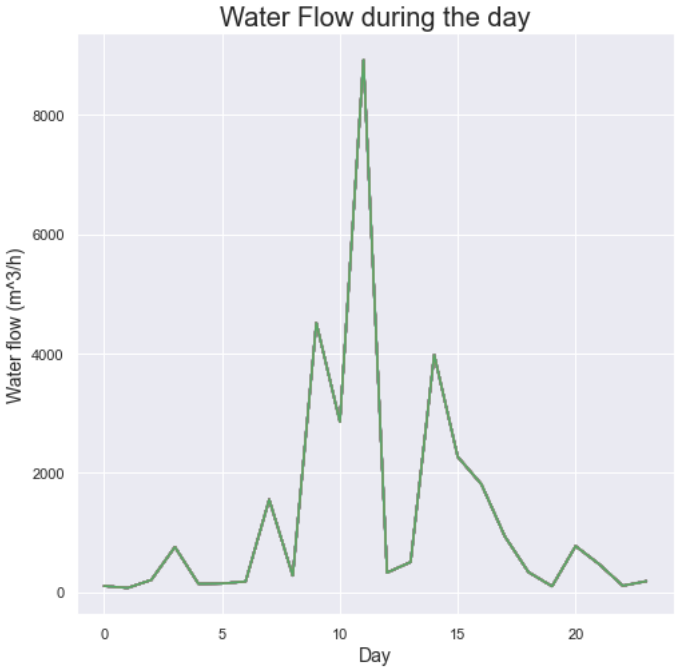


Figure 4.7. General water flow during the day

4.1.7 Water volumes and outliers

The interquartile range is a useful way in order to not to consider those values 'too far' from the real mean values, without letting the outliers to influence the mean of the normal values (like in the case of using the mean and the standard deviation).

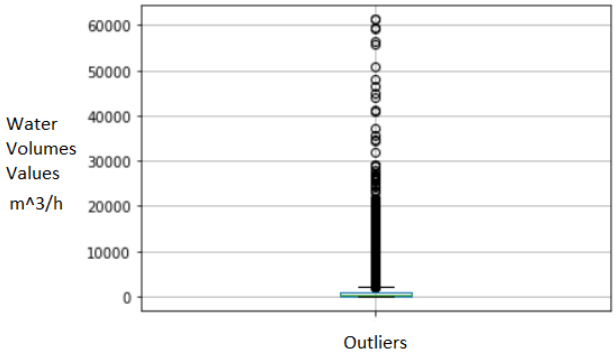
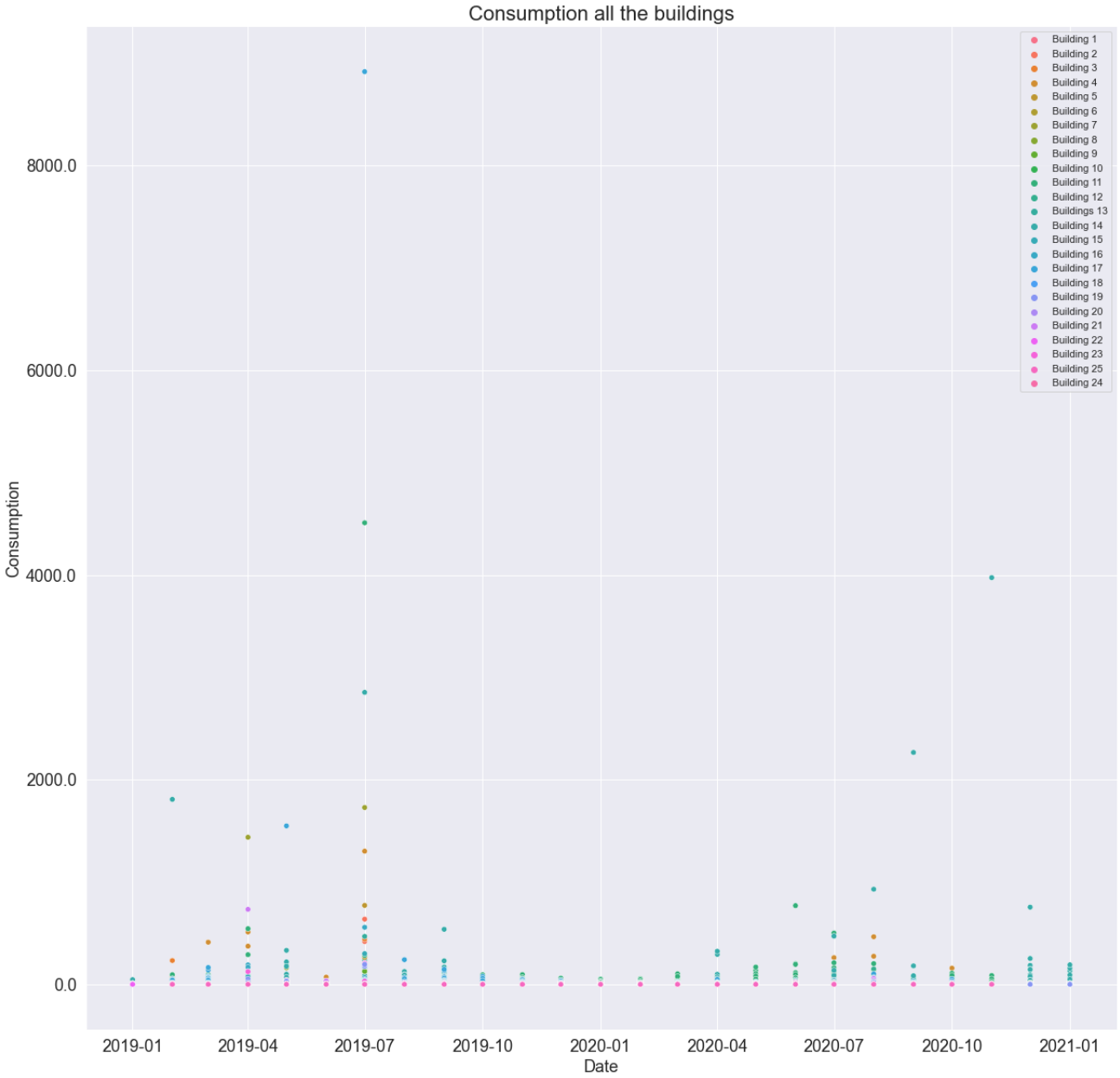


Figure 4.8. Whisker plot for showing the water volumes outliers

4.1.8 Consumption per building

The figure below shows the consumption during the mensuration period, divided by buildings. The biggest values are related to the Building 4, the Building 17 and the 14^{th}.


Consumption all the buildings
Building 1
Building 2
Building 3
Building 4
Building 5
Building 6
Building 7
Building 8
Building 9
Building 10
Building 11
Building 12
Buildings 13
Building 14
Building 15
Building 16
Building 17
Building 18
Building 19
Building 20
Building 21
Building 22
Building 23
Building 25
Building 24
8000.0
6000.0
4000.0
2000.0
0.0
Consumption
2019-01
2019-04
2019-07
2019-10
2020-01
2020-04
2020-07
2020-10
2021-01
Date

4.1.9 Hourly trend during the week

The mean of the water volumes during the day, divided by each day. On the y-axis we see the water volumes means, it can reach a maximum of 375 $\frac{m^3}{h}$. The weekend is visibly discernible from the rest of the week.

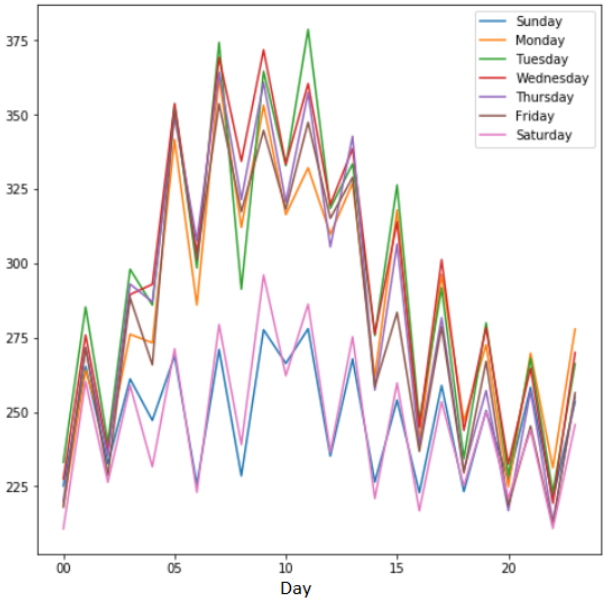


Figure 4.9. Water Volumes divided by weekdays

4.1.10 Night Consumption Top 5 Building

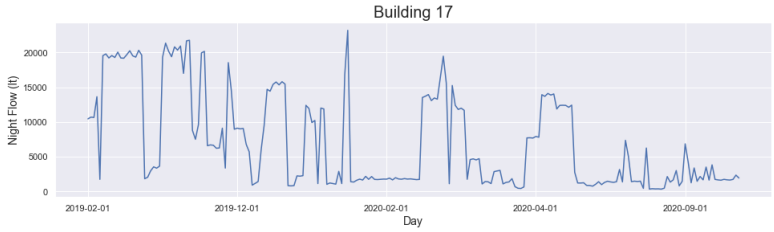


Figure 4.10. Night trend Building 17: This building is affected the most by the problem of water losses: we can see it during February March and May 2019, a little peak in January 2020.

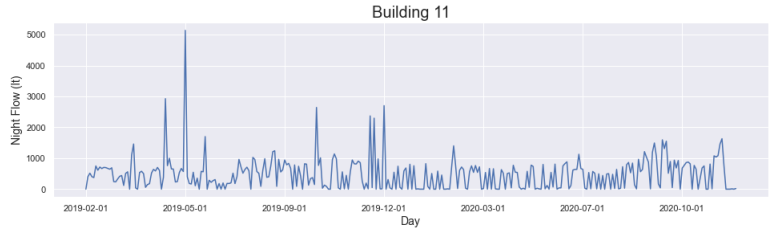


Figure 4.11. Night trend Building 11: the water consumption at night is really low, it reaches only one peak of 5000 during May 2019

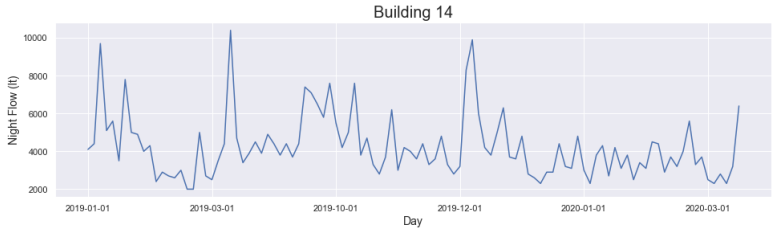


Figure 4.12. Night trend Building 14: the plot is not affected by any high values of anomalies: instead, the water consumption is pretty constant in the usage

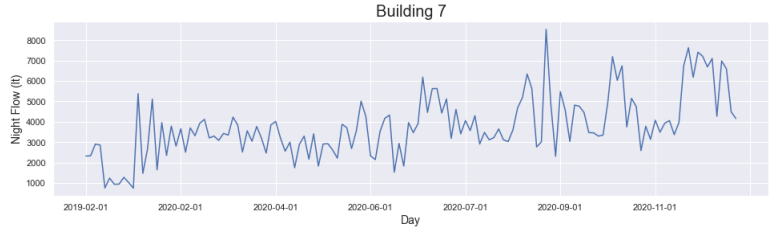


Figure 4.13. Night trend Building 7: the building got its highest values in the late period of measurement; especially during August 2020 and the end of the 2020

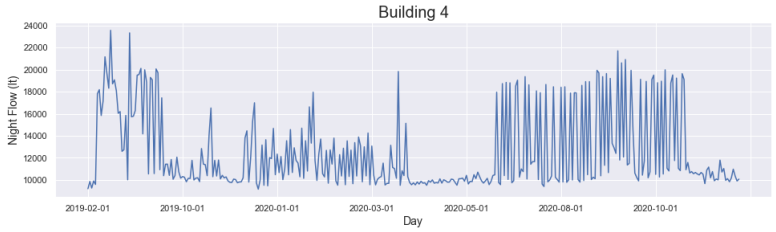


Figure 4.14. Night trend Building 4: this last plot shows high values of water losses during June 2019 and August 2019; there are also some irregularities in the whole period between June 2020 and November 2020. Here some anomalies are surely happening

4.2 Correlations

Observing the trend of the correlation between the pressure and the temperature during the year, it can be observed that the correlation between pressure and temperature is slightly positive, meaning that the temperature and the pressure are not affected to each other. This can be addressed to low temperature and low pressure (high temperature and pressure can predict a rapture).

About the negative correlation, low temperatures don't affect high values of pressure. At the same time, low pressure and high temperature is not a distress hence it does not contribute to any unpleasant event, such as a rupture in the pipelines.

In our case, the correlation barely varies and it's mostly around zero, meaning that the water pressure and temperature are not so influenced by each other. Except for February, in which the observed correlation is the biggest(maybe due to the winter time, in which pressure and temperature go together because of the need for hot water).

The hottest temperatures of water are even reported at the end of August, we can assume that in this case the negativity of the correlation is given by a high value of temperature and a lower pressure.

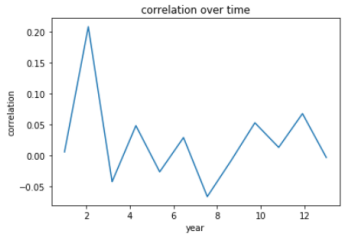


Figure 4.15. The total trend of the correlations between water pressure and temperature over the year (considering all the buildings together). The correlation is on the y-axis, and it is a Pearson correlation. The x-axis is represented by the 12 months. The study of the correlation is been conducted showing a year of correlation trend.

4.3 Further plots and considerations

4.3.1 Estimated time for the maintenance

We wanted to report how long the leakages in the pipelines would be observed by our data. If the leakage is no longer observed, it may be because the leakage is been fixed.

First, we detected the anomalies using the first algorithm of this study (5.7). Using these results we estimated the general time of reparation.

Year	Month	Building	Estimated Days
2019	02	Building 17	11
	03	Building 17	23
	05	Building 17	9
	06	Building 4	16
	07	Building 4	12
		Building 5	4
	08	Building 2	0
		Building 4	24
		Building 13	6
	09	Building 17	0
		Building 4	28
	10	Building 12	0
		Building 17	1
		Building 4	20
	11	Building 17	25
		Building 4	22
	12	Building 17	7
		Building 4	8
2020	01	Building 17	21
		Building 4	21
	02	Building 4	28
	03	Building 17	17
		Building 4	24
	04	Building 16	4
		Building 17	10
		Building 4	12
	05	Building 17	2
		Building 2	5
	06	Building 4	10
	07	Building 19	2
		Building 2	3
		Building 4	28
	08	Building 12	0
		Building 4	19
	09	Building 4	26
	10	Building 4	29
	11	Building 16	1
		Building 4	22
	12	Building 4	0

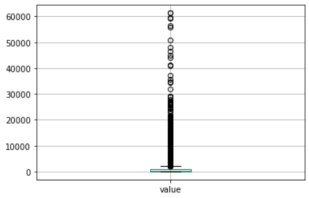
From this table we can estimate that the mean time for a reparation can be around 12.82 days. We see from the table that some times can also reach almost one month.

4.3.2 Outliers from the percentile

For the computation of the moving average (5.3, 5.4 and 5.5) we found the outliers and removed them from our data set. This cut doesn't look significant, since only 50 data points have been deleted, but actually it's a key point for the

contamination parameter in the Isolation Forest (5.4), and we will confirm it by the Cross Validation process in 5.12.

In [6]:
```
# retaining only the 5th and 95th percentile over the dataset
def take_off_outliers(data):
    data.boxplot(column=['value'])
    q1, q3= np.percentile(data['value'],[5,95])
    IQR = q3 - q1
    Lower = q1 - (1.5*IQR)
    Upper = q3 + (1.5*IQR)
    return data.loc[(data['value']>Lower) & (data['value']<Upper)]
night_flow = take_off_outliers(night_flow)
```



number of datapoints in the 5th and 95th percentile : 50

Previous number of datapoints - number of datapoints after detecting the outliers of the 5th and 95th percentile

9893 - 9843

In [165]:
```
50 / 9893
```

Out[165]:
```
0.005054078641463661
```

Figure 4.16

Chapter 5

Anomaly Detection

This solution emerges from the need of finding the water losses in a more precise way.

The lost water can return back in the pipeline, contaminating the quantity of water passing though them. This leads to a not-safe supply of water.

Altered quality of water can endanger the population's health. [17]

Saving water not only saves the costs of the water losses, but also the energies required for supply it in the grids (s.t. electrical, energy involved for the valves pressure).

Water losses are a real curse that affect especially the low-income countries that don't possess the necessary money and resources for investing in saving water [18].

Nowadays, we don't need hardware solutions in order to look for water losses. As hardware solutions, we talk about equipment that necessitates human intervention, together with chemical materials and/or mechanical designs[4]. Our approach is of course less intrusive, it requires less resources in the field and it is more responsive. At the same time, the technology of anomaly detection using statistics and machine learning results are really precise.

The availability of real-time data at high temporal frequency can help water utilities identify leaks and fixture malfunctions, timely schedule maintenance or upgrades of the infrastructure, and ultimately help them meet goals for sustainable water use[8].

Comparison between the expected behaviour and actual measurement can be done to isolate the leakage [19].

5.1 Automated Minimum Night Flow

For estimating all leakages, it's important to detect the inappropriate losses of water based on the current data. We accounted that during the night it is easier to compute the anomalies detection, because the human consumption is considerably reduced. Numerically, identifying the losses becomes straightforward.

As the graph in the Figure 4.7 tells us about the average trend during one day, we see it is almost null the water flow passing through the pipelines in the night (in exception of one peak).

According not only to our data (Figure 4.7) but also to [4], we took all the water volumes at 2a.m. and 5a.m. and computed the final consumption per night and per each building, from January 2019 until now. This calculus allowed us to find out the amount of water that every night was streaming in the grids per each volume device in each building.

Dividing by building we were able to see the Minimum Night Flow (MNF) during all the years of measurement.

5.1.1 Moving average and moving standard deviation

The operation of working out on the moving average (MA) helps us to smooth the not ordinary values inside the data, helping to create a constant, average, per-night consumption.

Before creating the vector of moving averages, we took off those outliers using the **percentile range between 5% and 95%**; that's a good technique instead of taking the average of the values, because it is not affected by the outliers, in this way it is stable to a real mean. Once found the points that determine which are the first and third quartile, taking off all of those where laying between the minumum one and the first quartile and those greater than the third quartile value.

And from his whisker plot in 4.3.2 we can deduce the water consumption value did not discord from usual values, in exception of the total 50 data points present in the first and third quartile.

The MA is computed passing through all the data points of the night flow data frame and calculating a vector of points that are based on the average of the past n values in the data frame (where n is the size of the window). Having around 10k data points, we found sufficient to operate over the data using a window of size 100.

The moving standard deviations are computed in the same fashion of the MA.

Once computed the simple moving average and moving standard deviation, we also want to prevent the false alarms.

This is conducted creating a flow threshold and comparing it together with the minimum night flow, showing when the MNF overtops the threshold. This threshold helps to identify the leakages.

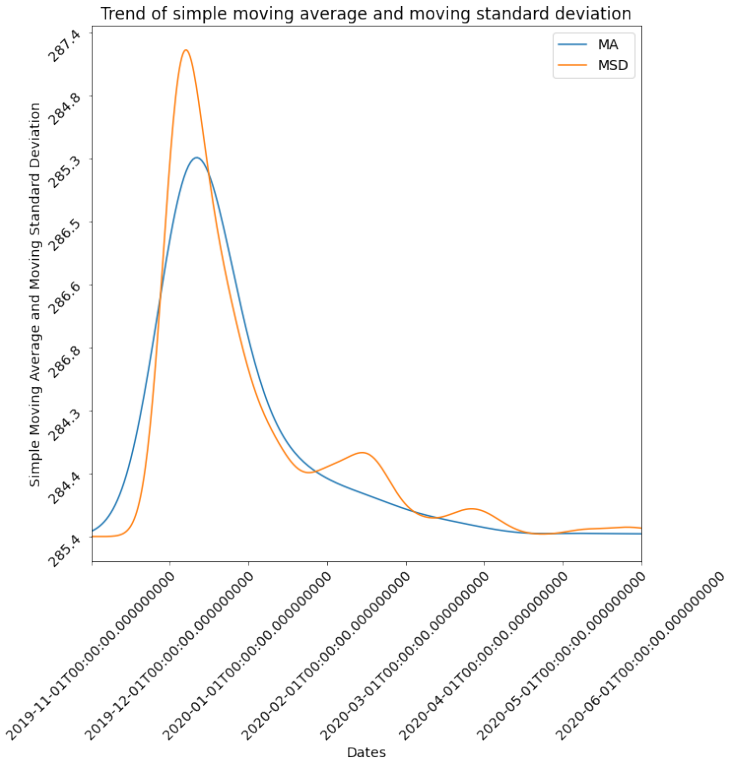


Figure 5.1. Simple moving average and moving standard deviation

Based on [4], we iterated over our values comparing the MA plus a value alpha multiplied by the Moving Standard Deviation of every single point together with the minimum night flow associated to that point.

$$\text{Night Flow value} < \text{Moving } \mu + \alpha * \text{Moving } \sigma \tag{5.1}$$

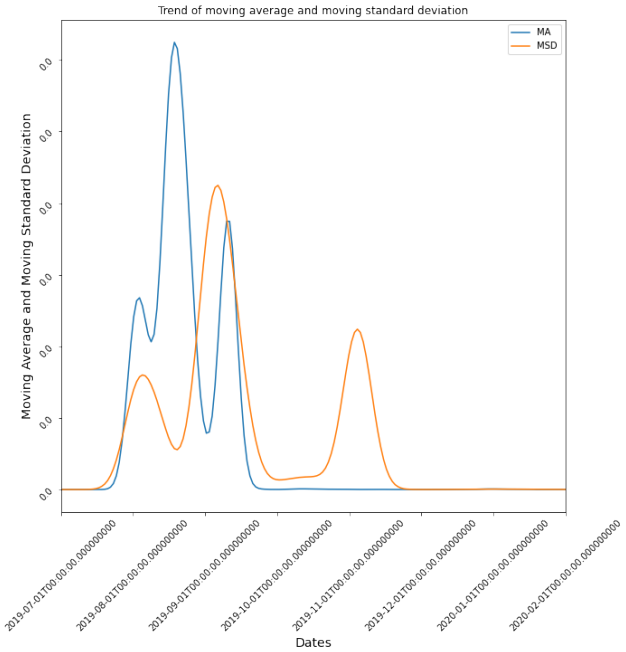


Figure 5.2. Moving average and moving standard deviation

This comparison is crucial for detecting the real losses and prevent false alarms.

With a value α equal to 5, we can reach the **accuracy of prediction** of leakages until the 97%, together with a **precision** of 62%.

Plotting the Minimum night flow, the moving average and the threshold obtained by the moving average.

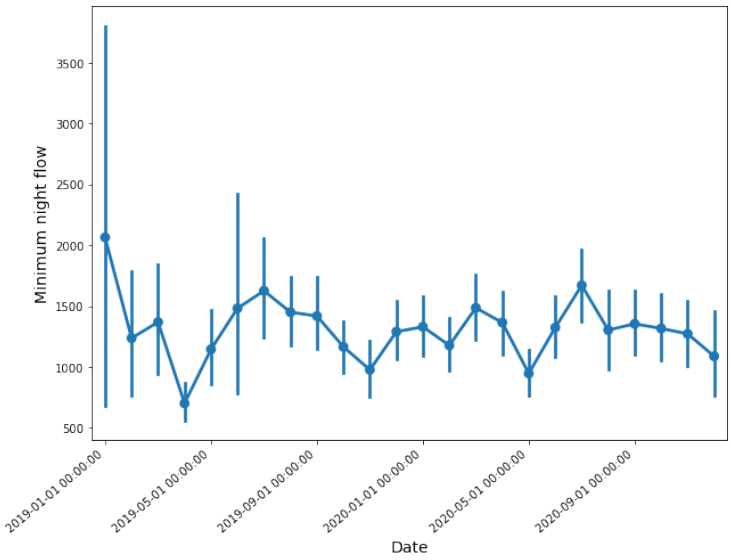


Figure 5.3. Trend Minimum Night Flow

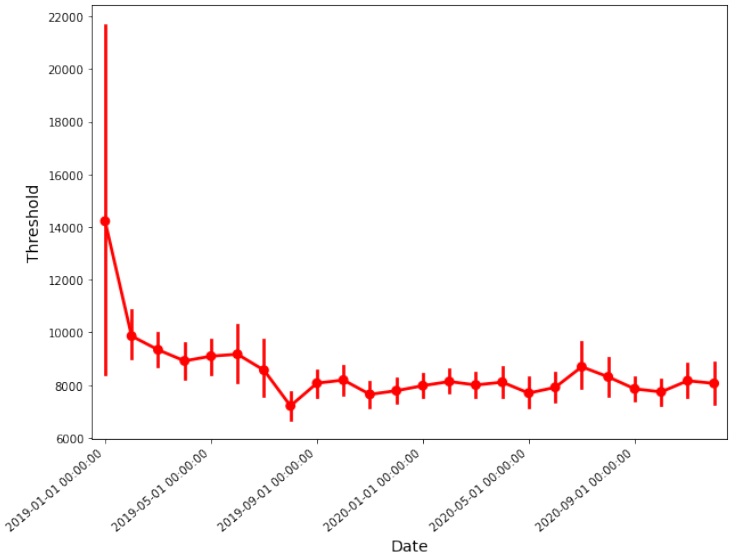


Figure 5.4. Trend threshold

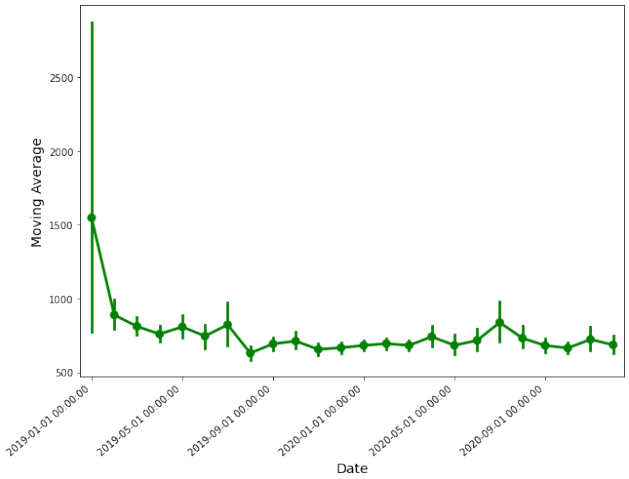


Figure 5.5. Trend of Moving Average

The figure below 5.6 shows the response of the binary vector used for the Automate minimum night flow, where the spikes show the cases in which the Minimum Night Flow exceeded the threshold. In some cases, during the same period, we see the values of minimum night flow exceeded also more than one time. We can see that the peak of anomalies is before August 2019.

There is a white space in the first months of 2020, where we can certainly say the water consumption didn't exceed the threshold we set at all; maybe it can be due to the lockdown.

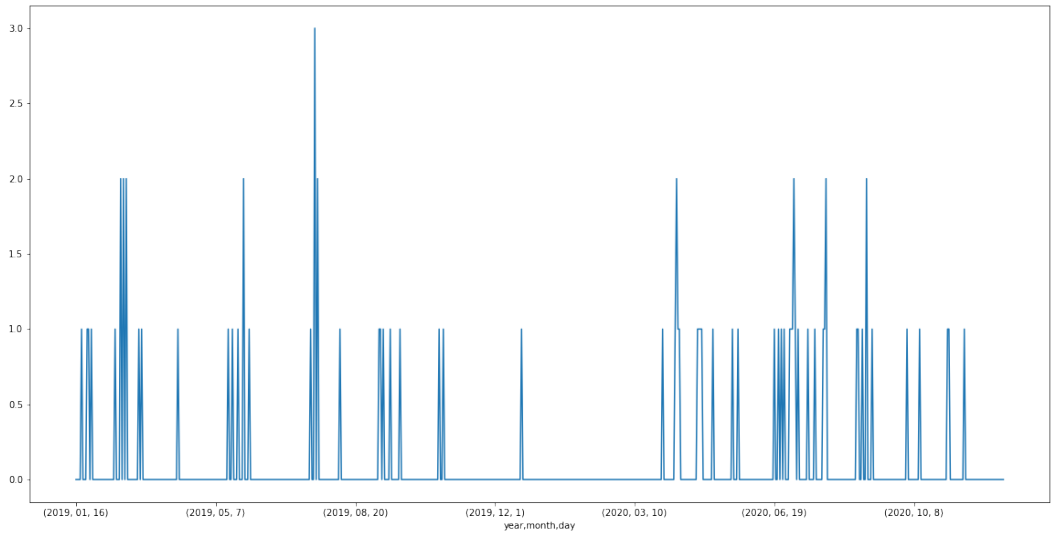


Figure 5.6. Water consumption Response on the threshold: on the x-axis the timeline, on the y-axis there are the responses of how many data points exceed the threshold. Thus, there is a visible case in which the threshold is exceeded three times

Year	Month	Building	Occurrence
2019	01	Building 14	1
	02	Building 7	4
		Building 25	1
		Building 16	1
	03	Building 7	1
		Building 13	1
		Building 25	2
		Building 16	1
		Building 12	1
	04	Building 13	1
	05	Building 13	3
		Building 25	1
		Building 2	1
	07	Building 5	1
		Building 21	1
		Building 2	2
		Building 8	1
	08	Building 21	1
	09	Building 20	1
		Building 9	1
		Building 12	2
		Building 8	1
	10	Building 12	2
	12	Building 10	1
2020	03	Building 25	1
	04	Building 25	1
		Building 21	1
		Building 16	3
		Building 10	1
		Building 2	3
	05	Building 20	1
		Building 2	2
	06	Building 19	3
		Building 13	1
		Building 3	1
	07	Building 23	2
		Building 5	2
		Building 19	2
		Building 13	3
		Building 24	2
		Building 3	1
	08	Building 12	1
		Building 10	2
	09	Building 25	1
		Building 21	1
		Building 3	1
	10	Building 25	1
		Building 8	1
	11	Building 25	1
		Building 16	2

Table 5.1. Anomalies Periods and Buildings detected by the AMNF

5.2 Random Cut Forest

The data are measured every hour per each device, thus this is a continuous stream of input. The random cut forest does not need to be reprogrammed for each learning, thus it can be used in real-time settings. Exploiting the data, dividing them by randomized sample, can guide to an effective methodology for anomaly detection [20]. In fact, recent work has shown that under suitable regularity conditions, averaging over predictions made by subsampled random forests produces asymptotically normal predictions[21].

The detection started using the Random Cut Forest algorithm [20], an unsupervised method to find the anomalies in a series of values.

Thus, a RCF helps looking for those values that are unusual, the 'spikes' in our series, those values of consumption too high reporting us the presence of one or more leaks.

The Random Cut Forest takes from the Random Forest implementation, a powerful tool that can avoid over fitting, in respect to other known predictive algorithms. It can be used in classification as in regression problems. It consists on taking random samples of the same data set in order to generate different classification or regression trees and predicting the data based on the n trees generated. Thus, the prediction of the data is based on the aggregation of different random trees that are been trained by different samples, that's why this kind of solution is less predisposed to over fitting.[22]

As already said, the RCF creates data structures as trees, and each node in it represents a data points. The anomaly is easy to spot because in the tree-architecture it is a leaf that, differently from its sibling, doesn't have children nodes.

In order to create a random cut forest, we had to specify how many:

- **number of trees**: the different number of trees not correlated between each other in the random forest, and
- **the sample per each tree**: so, in an indicative way, how many data point there have to be for each outlier encountered.

For this approach, we exploited Amazon Sage Maker (3.3.1), a SaaS service belonging to Amazon Web Service (AWS) (3.3).

This technology produce a new vector for the data set, a score vector, that tells for each data point passed how much it differs from the rest of the data points.

The score vector has to be based on the target column of the data set. The target column has to be a numeric column of real values (the values of consumption in this case) where higher the values of the scores indicates the more probable anomaly of the data point associated to that score.

Amazon Sage Maker procedure and results We exploited an instance for running faster our algorithm, passing our data set and the variables number of trees and number of sample per tree. This is the initialization of our random cut forest [23].

This random cut forest generates an inference in order to predict the values of the scores.

It's from these values that we took a column called 'scores' made of indicative values related on the level of consumption and computed the anomalies, taking the mean and the standard deviation of these scores. Outside a window of 3 times the standard deviation, we cut off the scores that were, then, considered as outliers.

Changing in an effective way the values of number of trees and sample of data points per outlier, we can obtain the best combination of parameters for predicting

the outliers. Using our data we calculated the ratio between the number of highest values of consumption over the number of all the non-negative data points, obtaining 500. It was a right number for the sample tree. We conveyed 50 as the number of tree because the buildings we had were 27. 50 is also the minimum number of tree that the Random Cut Forest Algorithm takes in input. Further, we thought that it's a good approach to not to generate too many trees, due to the fact that 50 is already enough for addressing the water behavior.

And this was the best case in all the study of our model for Random Cut Forest in AWS.

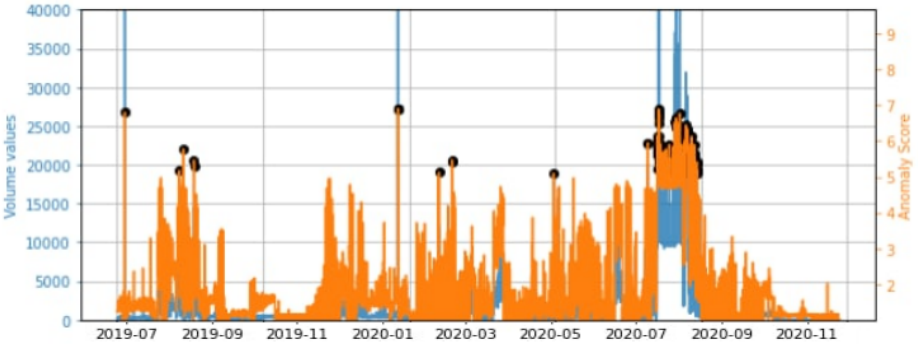


Figure 5.7. Random cut forest using Amazon Sage Maker

5.3 Periods of water anomalies

Based on the scores per data point given from Amazon Sage Maker (3.3.1), we were able to take the top of those anomalies that were more permanent over the time (Figure 5.2).

As we see, the period of anomalies are mostly located during 2019 (especially March, April and May), August and September 2019, and on March 2020.

Year	Month	Building	Occurrence
2019	02	Building 17	2
	03	Building 17	13
	05	Building 17	7
	06	Building 4	8
	07	Building 5	3
		Building 4	7
	08	Building 13	2
		Building 4	6
		Building 2	1
	09	Building 17	1
		Building 4	9
	10	Building 17	2
		Building 4	5
		Building 12	1
	11	Building 17	4
		Building 4	5
	12	Building 17	7
		Building 4	3
2020	01	Building 17	6
		Building 4	11
	02	Building 4	15
	03	Building 17	14
		Building 4	13
	04	Building 16	4
		Building 17	8
		Building 4	5
	05	Building 17	3
		Building 2	2
	06	Building 4	5
	07	Building 19	2
		Building 4	10
		Building 2	2
	08	Building 4	7
		Building 12	1
	09	Building 4	13
	10	Building 4	14
	11	Building 16	2
		Building 4	8
	12	Building 4	1

Table 5.2. Anomalies Periods and Buildings detected by the Random Cut Forest

Besides, we can see there are two buildings involved: the 4^{th} and the 17^{th} (Figure 5.85.9).

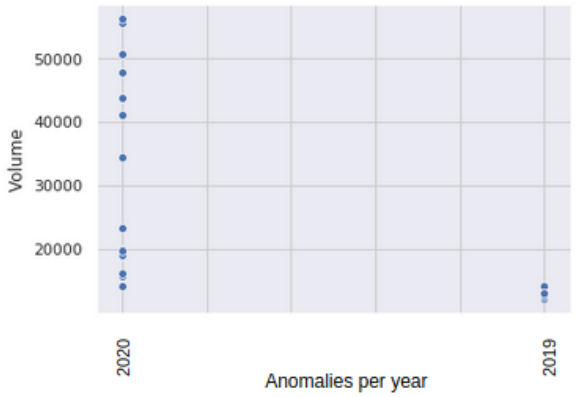


Figure 5.8. Anomalies volumes plot Building 4

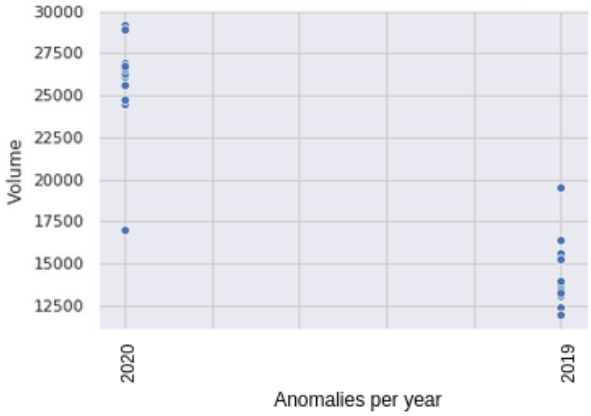


Figure 5.9. Anomalies volumes plots Building 17

5.4 Isolation Forest

The Random Cut Forest is nearly precise but time-consuming when it constructs trees and forests, thus we move to another methodology called Isolation Forest. The Isolation Forest is an efficient algorithm based on the mechanism of isolation and is fundamentally different from all existing statistical-based, distance-based, density-based, and clustering-based methods. This method achieves abnormal detection function by isolating instances instead of constructing a profile of normal data, which greatly improves the computational performance.[24]

This need of speed arises from what we already mention in 5.

An Isolation Forest works similarly to the Random Forest. In fact, both are ensemble methods, meaning that they produce strong classifiers using few resources, but with the difference that Isolation Forest spots directly the outliers values instead of profiling the normal data and differ them to the anomalies.[20]

As the Random Cut Forest, the Isolation Forest it's pretty fast and requires less memory in respect to other algorithms. In fact, the general approaches are trained for comprehending the normal data, not to detect the anomalies. This reflect in a bunch of false alarms or few spot of anomalies[25].

It has to be considered that the procedure was faster also due to the fact that we preprocessed our data in an earlier step, taking the data related to the volume, extracting only the (non-negative) consumption column, taking the per-night consumption.

Both IF and RF create a set of decision trees, but the Isolation Forest finds the path that brings to the anomaly in the tree. In the Isolation tree the anomalies are 'isolated' and close to its root node, thus are ranked at the top of the list; in disregard, the normal data points are located in the rest of the Isolation tree.The anomaly score of an instance i is computed sub sampling the data set taking n data points. The sub sample can be represented as X = $i_0, i_1, ..., i_n$. For each sub sample created we take the maximum and minimum value of the feature selected. Doing so recursively in the data set, we are able to get the path length for an isolated node.

Averaging this path length over all the random tree generated in the isolation forest, we are able to detect which are the abnormalities in the data set.

Isolation Forest Parameters In our implementation of the Isolation Forest we had to specify few parameters:

- **number of estimators**: 50 in our case, the number of base estimators in the ensemble (i. e. the number of trees we want to generate in our forest).
- **maximum of samples**: The number of samples to draw from X to train each base estimator.
- **contamination**: as in the Random Cut forest algorithm it is similar to the number of samples per trees, the number of outliers we can find in a bunch of data points.
- **max features**: the number of features used in order to define the outliers; in our case, the water consumption it's enough for our aim.

Once determined these arguments, the algorithm makes the average between of all the isolation tree distances between the roots and the data points, finding the best approximation of the bound that divides the points to be classified as anomaly or not. [25]

This is feasible because of the random partitioning of the data set, allowing the re-sampling of the data: more trees are generated, more the path length to the

anomalies and the path length of the normal data are differentiated from each other, converging to their values. But we have to pay attention: if the number of trees is extremely increased, the algorithm tends to converge too precisely to the boundaries that detect the anomaly, furthermore we lose in terms of speed of this technique.

The peculiarity of the Isolation forest algorithm is that anomalies are 'few and different', few because we spot the significant ones, without the presence of false anomalies, and different because the location of the anomalies are visibly in other points of the plots, we will see them in 5.12.

Isolation Forest for all the data points

As we see in the image, the plot is really similar to the one originated by the Random Cut Forest (Figure 5.10), telling us that the results are analogous for both the techniques when we pass the same data set.

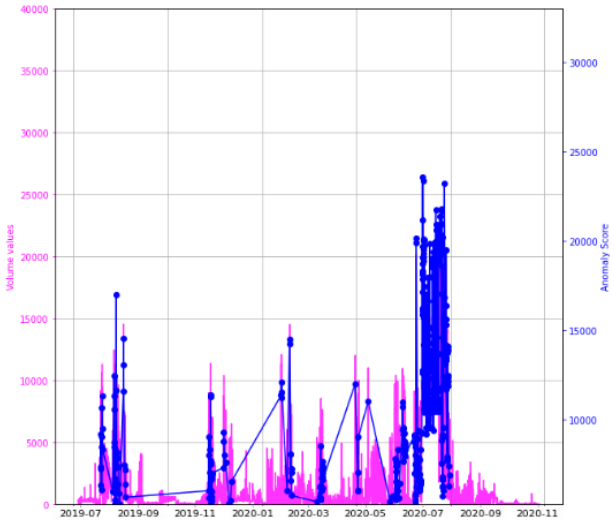


Figure 5.10. Isolation Forest for all the night data points, number of estimators 100 and contamination 0.05

5.4.1 Grid Search Cross Validation

We carried on our research, and to not 'manually' insert the hyper-parameters for the smartest combination of them 5.4, in technical terms we found the optimization between parameters in the Isolation Forest (Figure 5.4.1).

The Grid Search CV method is been exploited, where CV stands for Cross-Validation. This method splits the data set in k parts, using k-1 splits for training the data and the last portion is used for the test. Hence, we are able to evaluate the model having different k accuracies of it.

```
from sklearn.model_selection import GridSearchCV

x = night_flow.loc[:,['value','year','month','day']]
# y = night_flow.loc[:, 'value']

def scorer_f(estimator, X):   #your own scorer
      return np.mean(estimator.score_samples(X))

param_grid = {'n_estimators': [50, 100],
              'max_samples': ['auto', 30],
              'contamination': [0.005, 0.01, 0.05, 0.1],
              'max_features': [1],
              # 'bootstrap': [True, False],
              'n_jobs': [-1]}

# f1sc = make_scorer(f1_score(average='micro'), y_test, y_pred_test)

#or you could use a lambda aexpression as shown below
#scorer = lambda est, data: np.mean(est.score_samples(data))
clf = IsolationForest(random_state=47)
```

```
isolation_forest = GridSearchCV(clf, param_grid, scoring=scorer_f)
```

```
model = isolation_forest.fit(x)
```

```
print(model.best_params_)
```

```
{'contamination': 0.005, 'max_features': 1, 'max_samples': 'auto', 'n_estimators': 50, 'n_jobs': -1}
```

Figure 5.11. Grid Search CV

The best parameters obtained are: {contamination: 0.005, max_features: 1, max_samples: auto, n_estimators: 50, n_jobs: -1}.

For usability reasons, we are going to show only 5 Isolation forests obtained from each building, compared in the first and in the second case.

Starting from the same data, we can reach different conclusions: In the case of having 100 estimators and 0.05 contamination, the Isolation Forest caught a considerable amount of anomalies (for example, if we think in the first plot of the Building 4, 15 suspicious anomalies happened during two years). We can easily spot that some of these are not actually anomalies, so we are facing false alarms. In addition, we set a contamination quite high, 0.05: in a data set of almost 10k points, it would mean that we would have around 500 total water losses in the two-study years. Instead, looking also to the 4.1.7, the method got 50 outliers over a total number of 10k points. Thus, we can state that the value of our contamination should be lowered and to definitely be around 0.005, as shown in 5.4.1.

5.4.2 Isolation Forest Results

Year	Month	Building	Occurrence
2019	02	Building 17	1
	03	Building 17	9
	05	Building 17	7
	06	Building 4	3
	08	Building 4	2
	09	Building 4	6
	10	Building 17	2
	11	Building 17	2
2020	01	Building 17	1
	03	Building 17	1
	04	Building 4	1
	07	Building 19	2
		Building 4	1
	09	Building 4	6
	10	Building 4	4
	11	Building 4	2

Table 5.3. Anomalies detected in the Isolation Forest case: periods and buildings involved

5.4.3 Isolation Forest best parameters

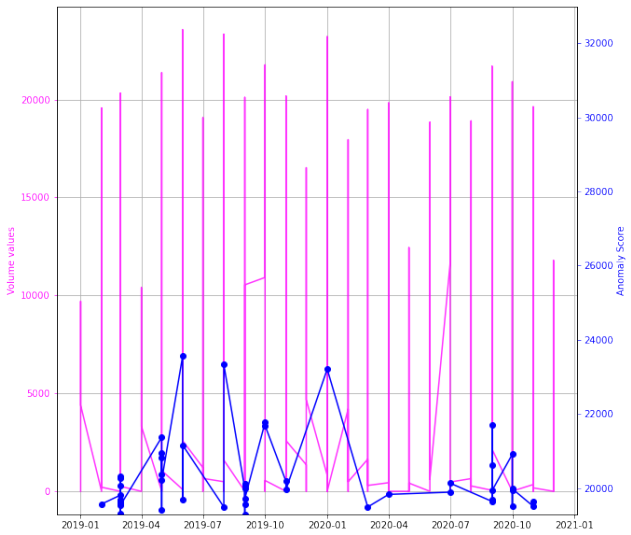


Figure 5.12. Best parameters plot over time: Showing the isolation forest in the case of n estimator 50 and contamination 0.005: we got less points detected as anomalies in comparison with the first Isolation Forest

5.4.4 Best parameters plot per building

Once we ran the algorithm for the whole dataset and got the results (5.3), we want to show all the anomalies divided by building.

The images are shown ordered by the maximum consumption per building 4.1.4.

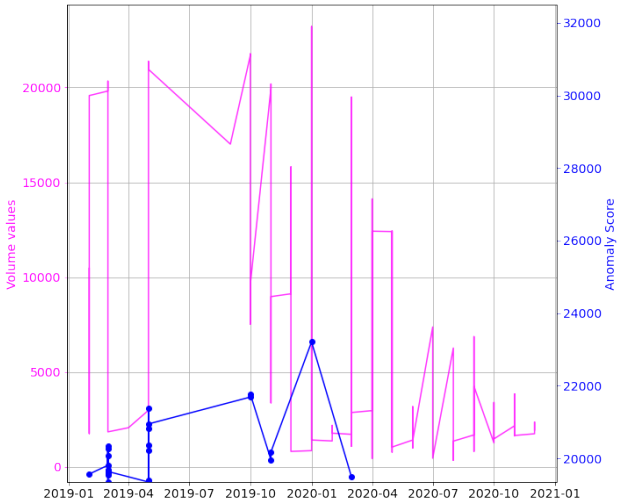


Figure 5.13. Showing the anomalies in the Building 17 according to the IF algorithm ran for the whole dataset

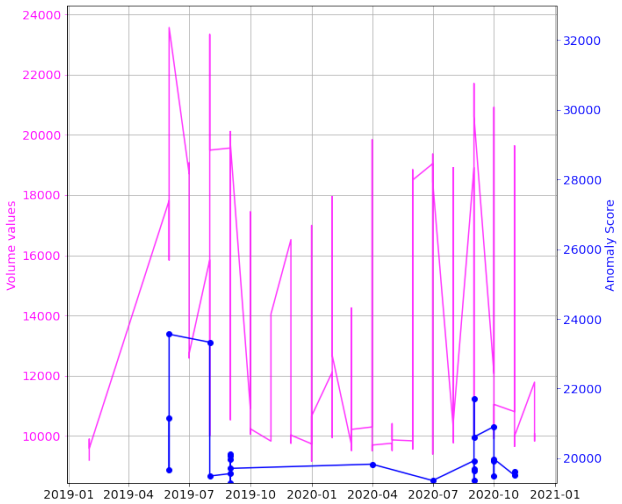


Figure 5.14. Showing the anomalies in the Building 4 according to the IF algorithm ran for the whole dataset

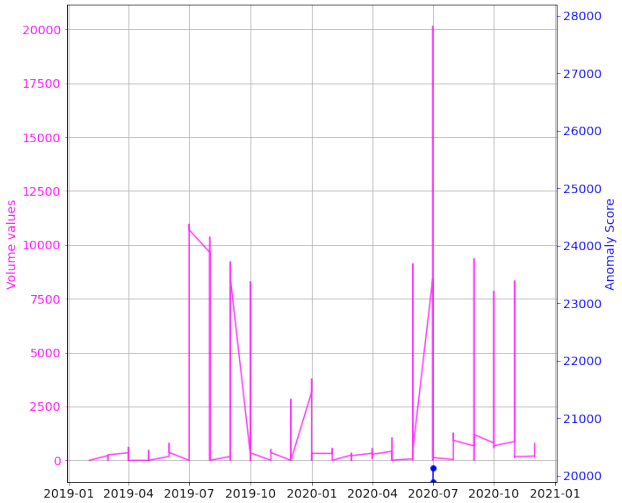


Figure 5.15. Showing the anomalies in the Building 19 according to the IF algorithm ran for the whole dataset

5.4.5 Best parameters Isolation Forests, Top 5

We ran the algorithm per each building alone using the best parameters: either with the high water usages in the buildings, the isolation forest identified very less anomalies when we ran the algorithm in comparison with the RCF algorithm, making the Isolation Forest more reliable.

It is more acceptable to have one or two disruptions per building in two years.

We can also ascertain that few ruptures took place for any possible reason. Furthermore, basing the study also on our exploratory analysis 4.6, water and ambient temperature don't reach freezing values, hence there is less possibility to experience a rupture (then, a water loss) in this sense.

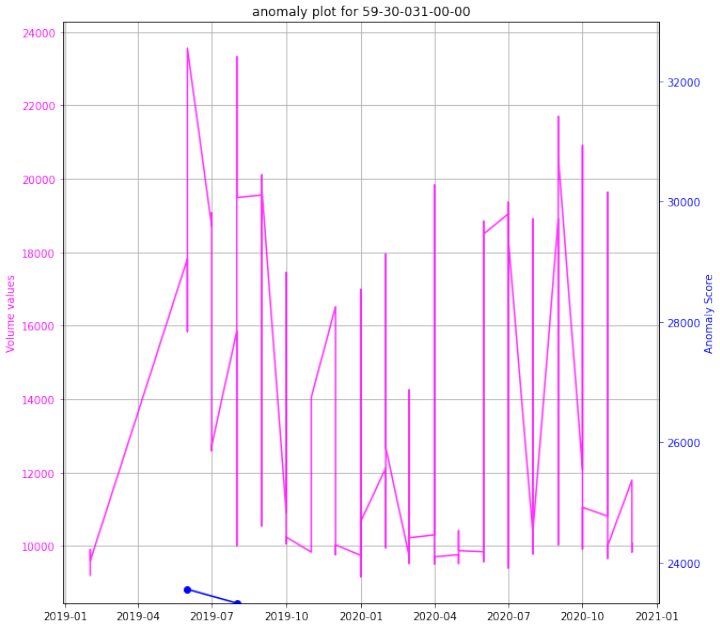


Figure 5.16. Isolation Forest Building 4

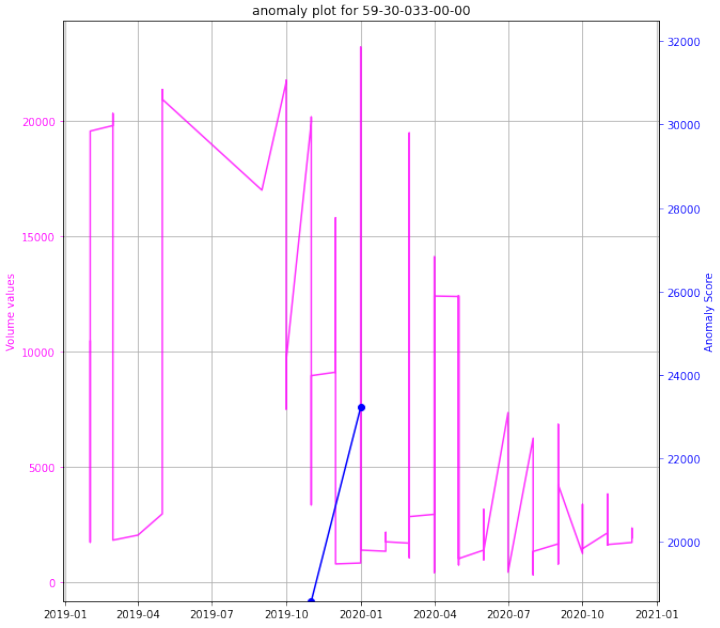


Figure 5.17. Isolation Forest Building 17

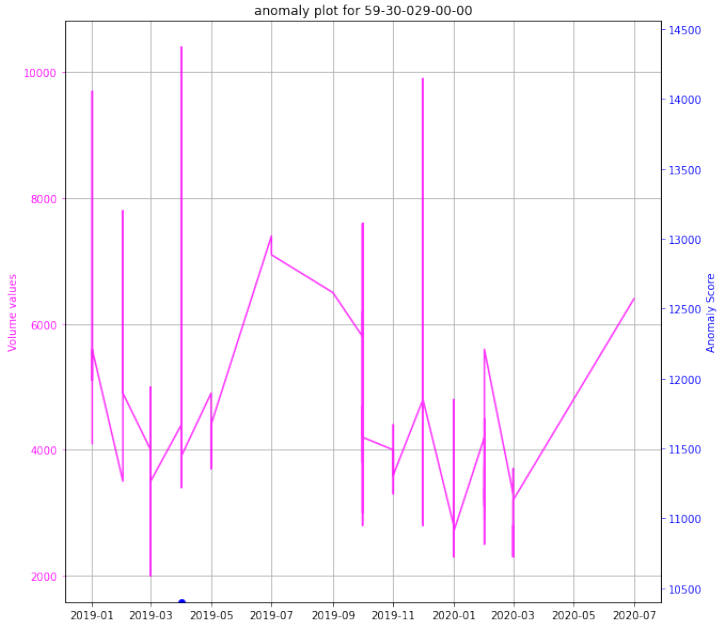


Figure 5.18. Isolation Forest Building 14

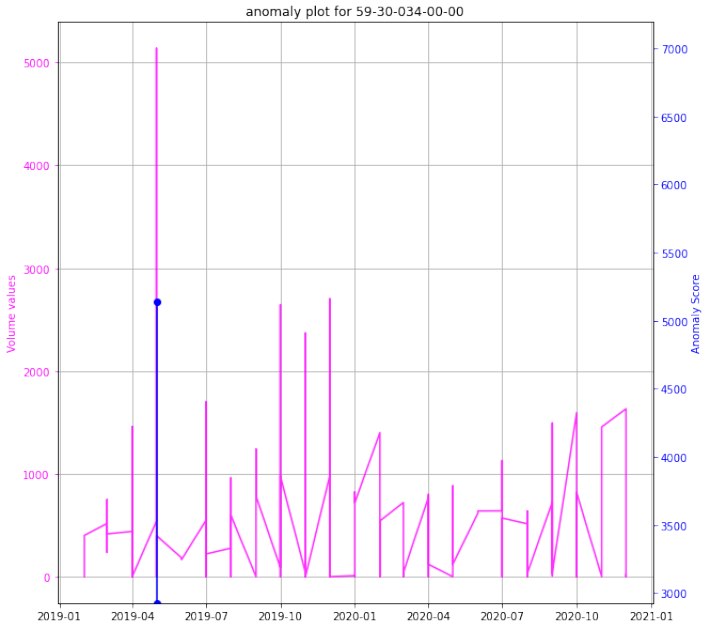


Figure 5.19. Isolation Forest Building 11

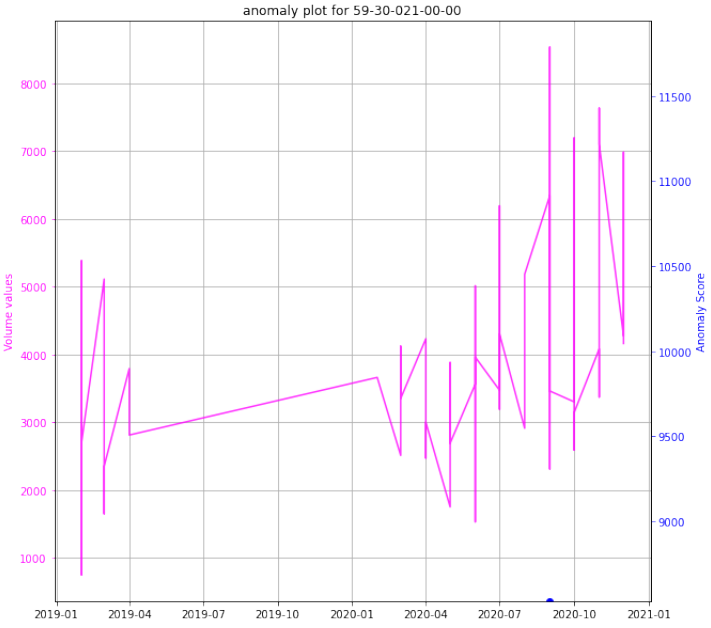


Figure 5.20. Isolation Forest Building 7

Other examples for the Isolation forest with 0.005 contamination

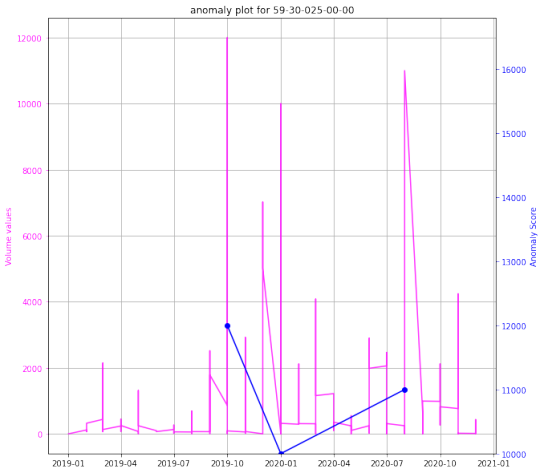


Figure 5.21. Isolation Forest Building 12

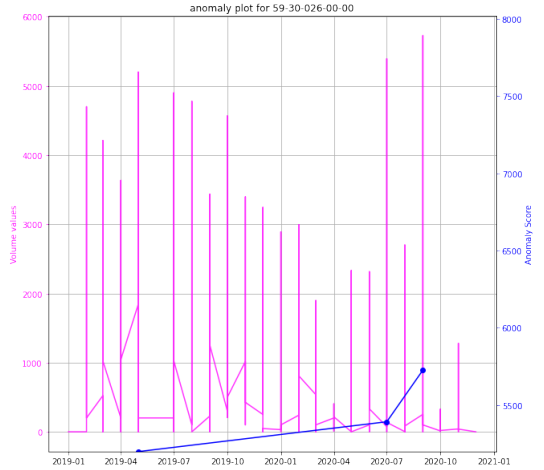


Figure 5.22. Isolation Forest Building 3

5.5 K-Means

Additionally, we used another unsupervised algorithm, the K-Means clustering.

Initialized the number of clusters we want to see as classes, the algorithm minimizes the mean squared distance from each data point to its nearest center.

We wanted two types of classes, respectively for those data points that could be classified as normal or as anomaly.

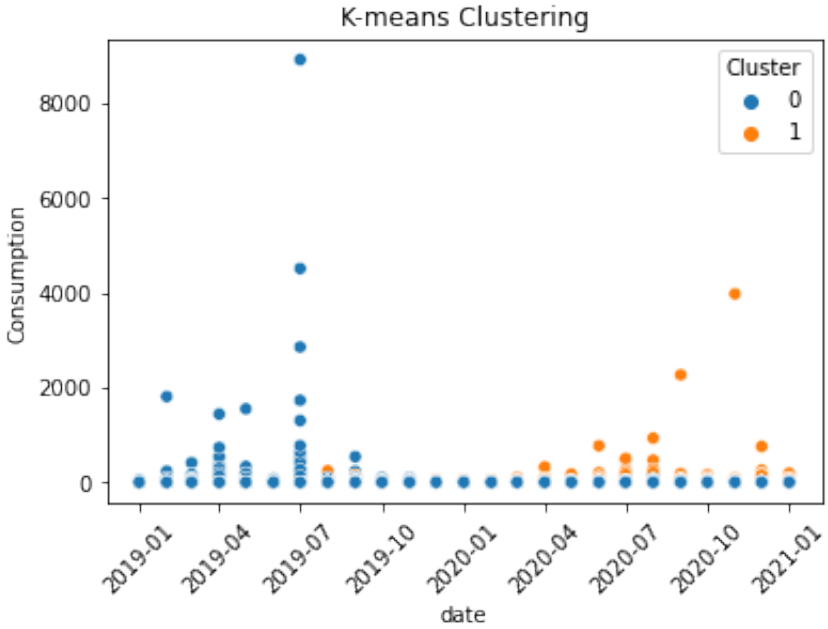


Figure 5.23. At a first glance, we see the first data points are classified as normal, and, over the time, more data points have been classified as anomaly

For a clearer view, the k-means clusterization is been screened only for the night flow data points (Figure below).

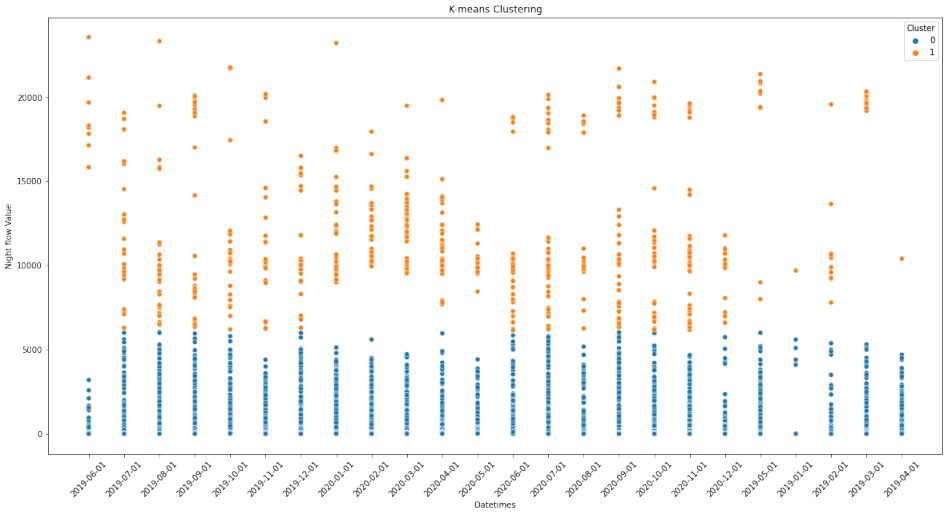


Figure 5.24. Night flow K-Means Clustering over time

5.6 Comparison of Results

At a first glance, we operated over the data set exploiting different parameters in our Isolation Forest.

The first example 5.10, using 100 estimators, and a high value of contamination that is 0.05. This one may seems low, but having around 10.000 data points referred to the night water consumption in our case, it would mean to have **500 losses** for the whole duration of measurement. Hence it looks quite hard to happen, as we can see in 5.2.

In fact, comparing this plot with the one from Amazon Sage Maker (Figure 5.7): The anomalies found in the Isolation Forest look many more than in the case of the Random Cut Forest.

This can be seen as unusual behavior of our isolation forest, predicting too many anomalies that are actually not true.

The data points behind the graph tells us we are working on the same data. Summarizing our results:

General overview		
	% of anomalies	Computation time
Random Cut Forest (AWS)	2.38	$\approx$ 15 mins
K-Means	6.18	$\approx$ 2 mins
Isolation Forest	0.51	few seconds
AMNF (accuracy of 97%)	0.78	few mins

The Automated Minimum Night Flow is been already deepen in [4]. It is necessary to spend few words on these other methodologies: as we saw, the Random Cut Forest generates many more anomalies. But some anomalies can result in false alarms, and also false alarms are expensive.

Hence, we want to detect at least the minimum amount of anomalies in the smart water grid, in order to save water.

This is why we can use two types of detection, the Automated Minimum Night Flow and Isolation Forest, respectively a sequential statistical analysis and an ensemble method.

The amount of anomalies of these two methods doesn't reach the amount of

anomalies detected by the Random Cut Forest (we want to remind that, over a dataset of 10k data points, the Random Cut Forest was detecting 2.38% of anomalies and the AMNF together with the Isolation Forest, detect an overall of anomalies of 1.21%). And apart from this, the two methods together are able to detect the same anomalies that also the Random Cut Forest can detect.

Some of the common buildings detected were: the Building 4, Building 17, the Building 19, the Building 16, the Building 2 and the Building 13.

Anomalies values obtained per each algorithm

For each algorithm used we predicted all the anomalies, here the maximum and minimum values.

Isolation Forest detected the highest values, we are sure that the Isolation Forest picked the strongest water losses. The AMNF has a little minimum value of anomaly: because the algorithm create a new column called Moving Average, it focuses on the last 100 rows of the dataset, thus we can deduct that it has detected a water loss (and not the only one, this is the minimum value) in a period in which the buildings were not frequently visited.

	Min value of anomaly ($\frac{m^3}{h}$)	Max value of anomaly ($\frac{m^3}{h}$)
Random Cut Forest (AWS)	11000	23210
K-Means	5200	23210
Isolation Forest	19290	23210
AMNF (accuracy of 97%)	61	14500

Comparisons of the technologies

We can focus on RCF and the Isolation Forest technologies, in order to see the pros and cons.

RCF	Isolation Forest
requires some time for the computation	fast
requires memory	require almost no memory
effective	effective
expensive	cheap

Chapter 6

Conclusions and future works

In this trend, we can detect spikes of water consumption. As we already said, we can detect the anomalies in our smart water grid combining two algorithms, the Automated Minimum Night Flow (AMNF), and the Isolation Forest. The Isolation Forest detects the biggest spikes, meantime the AMNF take those point which are not necessarily high, but in between a normal behavior and the strong anomaly behavior.

Thus, the Isolation Forest algorithm detected the biggest water losses in the smart water grid, without leaving any possible false alarm.

The AMNF procedure detected:

- water leakages that were not detected by the others algorithms;
- really low water leakages due to the fact that in that period the water was not used at all, but still some leakages were taking place;
- in the case of Building 19, Building 2, Building 13 and Building 12, the AMNF was able to detect in advance the leakages;
- in the case of Building 16, AMNF started to detect strange behaviors already in February and March 2019, one year before the Random Cut Forest would detect the anomalies in April 2020.

Although, we can ascertain that the detection is been conducted but all of the losses got already fixed: we didn't make it on time for intervening.

From the numbers we got, the maximum water lost can be compared from 7 to 9 Olympic swimming pools.

For this reason, it's important to apply these machine learning algorithms in real-time and with a quick response.

Interacting with the real world and in a fast way can be unsafe, impractical or band-width limited.

Hence, this challenge puts us in front of several performance requirements: low latency, fault tolerance, debuggability and profiling [26].

The solution proposed can help in addressing the problem not only in terms of responsive fixing, but also for an ethical and environmental account.

Bibliography

[1] Debra G Coy. Looking at water: a view from wall street. *Water Resources Impact*, 4:14–18, 2002.

[2] ENTREPRENEUR STAFF. Water begins trading on wall street in the futures market for fear of shortages. https://www.entrepreneur.com/article/361135, december 2020.

[3] Seung Won Lee, Sarper Sarp, Dong Jin Jeon, and Joon Ha Kim. Smart water grid: the future water management platform. *Desalination and Water Treatment*, 55(2):339–346, 2015.

[4] Elias Farah and Isam Shahrour. Leakage detection using smart water system: combination of water balance and automated minimum night flow. *Water Resources Management*, 31(15):4821–4833, 2017.

[5] Wikipedia contributors. Internet of things — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Internet_of_things&oldid=1007483235, 2021. [Online; accessed 20-February-2021].

[6] Wikipedia contributors. Exploratory data analysis — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Exploratory_data_analysis&oldid=1007327334, 2021. [Online; accessed 20-February-2021].

[7] EC Amazon. Amazon web services. *Available in: http://aws. amazon. com/es/ec2/(November 2012)*, 2015.

[8] Dimitrios Amaxilatis, Ioannis Chatzigiannakis, Christos Tselios, Nikolaos Tsironis, Nikos Niakas, and Simos Papadogeorgos. A smart water metering deployment based on the fog computing paradigm. *Applied Sciences*, 10(6):1965, 2020.

[9] Ioannis Chatzigiannakis, Luca Maiano, Panagiotis Trakadas, Aris Anagnostopoulos, Federico Bacci, Panagiotis Karkazis, Paul G Spirakis, and Theodore Zahariadis. Data-driven intrusion detection for ambient intelligence. In *European Conference on Ambient Intelligence*, pages 235–251. Springer, 2019.

[10] Eryk Lewinson. Outlier detection with hampel filter. https://towardsdatascience.com/outlier-detection-with-hampel-filter-85ddf523c73d?gi=8329281ea215, september 2019.

[11] Stuart Lloyd. Least squares quantization in pcm 25. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[12] David Sculley. Web-scale k-means clustering 40. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[13] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering 11. *SIAM Journal on Computing*, 49(3):601–657, 2020.

[14] Sudipto Guha, Adam Meyerson, Nina Mishra, Rajeev Motwani, and Liadan O'Callaghan. Clustering data streams: Theory and practice 13. *IEEE transactions on knowledge and data engineering*, 15(3):515–528, 2003.

[15] Edo Liberty, Zohar Karnin, Bing Xiang, Laurence Rouesnel, Baris Coskun, Ramesh Nallapati, Julio Delgado, Amir Sadoughi, Yury Astashonok, Piali Das, et al. Elastic machine learning algorithms in amazon sagemaker. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 731–737, 2020.

[16] Samar Mirghani and Hassan Hajjdiab. Comparison between amazon s3 and google cloud drive. In *Proceedings of the 2017 2nd International Conference on Communication and Information Systems*, pages 250–255, 2017.

[17] World Health Organization et al. *Water safety in buildings*. World Health Organization, 2011.

[18] Suraj Kumar Bhagat, Wakjira Welde, Olana Tesfaye, Tran Minh Tung, Nadhir Al-Ansari, Sinan Q Salih, Zaher Mundher Yaseen, et al. Evaluating physical and fiscal water leakage in water distribution system. *Water*, 11(10):2091, 2019.

[19] Tom Nørgaard Jensen, Vicenç Puig, Juli Romera, Carsten Skovmose Kallesøe, Rafał Wisniewski, and Jan Dimon Bendtsen. Leakage localization in water distribution using data-driven models and sensitivity analysis. *Ifac-papersonline*, 51(24):736–741, 2018.

[20] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2712–2721, New York, New York, USA, 20–22 Jun 2016. PMLR.

[21] Richard J McAlexander and Lucas Mentch. Predictive inference with random forests: A new perspective on classical analyses. *Research & Politics*, 7(1):2053168020905487, 2020.

[22] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[23] Lauren Yu. https://sagemaker.readthedocs.io/en/stable/algorithms/randomcutforest.html, july 2020.

[24] Shiyuan Hu, Jinliang Gao, Dan Zhong, Liqun Deng, Chenhao Ou, and Ping Xin. An innovative hourly water demand forecasting preprocessing framework with local outlier correction and adaptive decomposition techniques. *Water*, 13(5):582, 2021.

[25] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[26] Robert Nishihara, Philipp Moritz, Stephanie Wang, Alexey Tumanov, William Paul, Johann Schleier-Smith, Richard Liaw, Mehrdad Niknami, Michael I Jordan, and Ion Stoica. Real-time machine learning: The missing pieces. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems*, pages 106–110, 2017.