Dimensionality reduction and clustering of energy consumption time series in supermarket buildings

Lorenzo Salmina, Justine Stoll

Course of Machine Learning (CS-433), EPFL

I. Introduction

This project focuses on anomaly detection in time series recording among other, the electricity consumption of supermarket buildings. The target is reached by using Machine Learning unsupervised techniques (clustering methods) able to separate weeks with anomalous consumption from those with a standard one.

In this report, we outline the process that brought us from the raw data, as we obtained it, to the final result obtained by clustering the data. We started with a crucial step of data exploration, and data preprocessing. We then, looked into ways to represent the time series as data points. Proceeding with the analysis, we performed a principal component analysis (PCA), and applied clustering methods. All of these steps are performed in the four notebooks provided with this report, one can run them in their indicated order. Note that some of the plots have to be re run by the reader, as they contain sliders. Note also that at times, we will invite the reader to refer to these notebooks.

II. Data set description

The data set we used in this project contains time series recordings of the electricity consumption, as well as other numerical features in supermarkets. The time interval over which we have these recordings spans over approximately 3 years. More precisely from 2017-06-01 to 2020-08-28, which corresponds to 168 weeks, thus 168 data points. Measurements are performed on 35 numerical features.

The Data describes the heating system of a supermarket, the system is composed of two heating sources: the waste heat from the refrigerators and the heat generated by the heating units. The sources are controlled by pumps and valves. The activity of those valves is recorded as well as the temperature and the power consumption at different points of the system.

III. Data exploration

The first step of our project was to gather general information related to our dataset using the jupyter notebook: *Exploration_and_first_processing_of_data_set*. To perform this task we inspected the data using pandas, and identified 35 different features, each one corresponding to a sensor reading of the system. We also found some useful metadata reporting units of each reading and the type of interpolation that was done in case of missing data. Just by looking at the first 50 readings in the dataset we observed that the sampling rate was not constant (probably the data where saved as they become available), this problem is dealt with in the preprocessing section. Next we plotted the distribution of all 35 features identifying 11 binary features and 5 features that had constant value for all the data points. Then, for each feature we made two plots: one with all the datapoints and one with approximately 3 weeks worth of data, from this plots we identified different kinds of periodicity: annual, weekly and daily. An example for this is shown in figure 1.

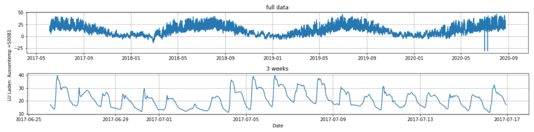

Fig. 1: Example of time series with annual,weekly and daily periodicity

Finally, we had the chance to meet with the building management team, who explained to us the basics of the system. Therefore, we were able to make a schematic representation and locate each feature, as can be seen in figure 2. The building management team also confirmed to us that the 5 features that had constant value where useless for our project and that we could safely discard them.

IV. Preprocessing

The preprocessing of our data is focused on three main tasks. First, we resampled the data in order to obtain a constant sample rate of 1h. The resampling of the non binary feature was done using the pandas method *DataFrame.resample* setting the value of each hour to the mean of the measurements that where done in that hour. For the binary features we counted for each hour the number of ones (i.e the number of times that the valve or pump was on).

The second task was to standardize the data, we choose the MinMax standardization since it is a common standardization for PCA analysis (which we performed subsequently). Then, we verified the results by comparing the initial distribution and the standardizated distribution. This can be seen for each feature in the plots of notebook *Data_visualization_correlation*.

Finally we cut the head and the tail of the dataframe so that the first data point corresponds to the first hour of the

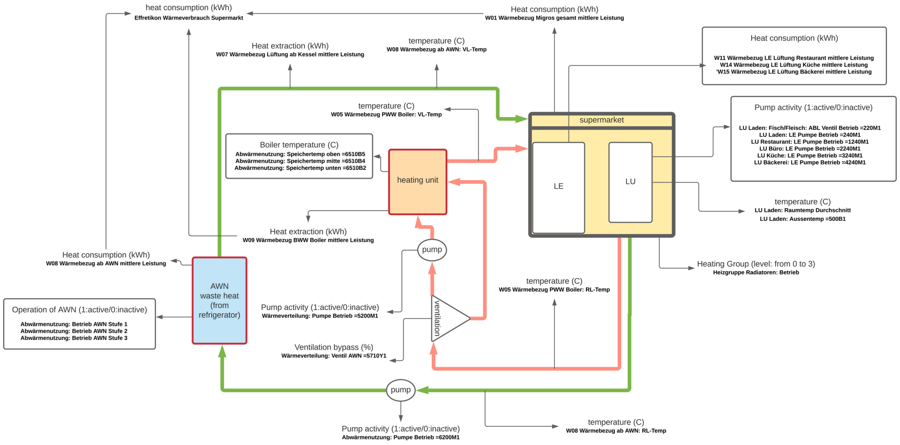
heat consumption (kWh)
Effretikon Wärmeverbrauch Supermarkt
Heat consumption (kWh)
W01 Wärmebezug Migros gesamt mittlere Leistung
Heat extraction (kWh)
W07 Wärmebezug Lüftung ab Kessel mittlere Leistung
temperature (C)
W08 Wärmebezug ab AWN: VL-Temp
Heat consumption (kWh)
W11 Wärmebezug LE Lüftung Restaurant mittlere Leistung
W14 Wärmebezug LE Lüftung Küche mittlere Leistung
'W15 Wärmebezug LE Lüftung Bäckerei mittlere Leistung
temperature (C)
W05 Wärmebezug PWW Boiler: VL-Temp
supermarket
Pump activity (1:active/0:inactive)
LU Laden: Fisch/Fleisch: ABL Ventil Betrieb =220M1
LU Laden: LE Pumpe Betrieb =240M1
LU Restaurant: LE Pumpe Betrieb =1240M1
LU Büro: LE Pumpe Betrieb =2240M1
LU Küche: LE Pumpe Betrieb =3240M1
LU Bäckerei: LE Pumpe Betrieb =4240M1
Boiler temperature (C)
Abwärmenutzung: Speichertemp oben =6510B5
Abwärmenutzung: Speichertemp mitte =6510B4
Abwärmenutzung: Speichertemp unten =6510B2
heating unit
LE
LU
temperature (C)
LU Laden: Raumtemp Durchschnitt
LU Laden: Aussentemp =500B1
Heat extraction (kWh)
W09 Wärmebezug BWW Boiler mittlere Leistung
pump
Heating Group (level: from 0 to 3)
Heizgruppe Radiatoren: Betrieb
Heat consumption (kWh)
W08 Wärmebezug ab AWN mittlere Leistung
temperature (C)
W05 Wärmebezug PWW Boiler: RL-Temp
AWN
waste heat
(from
refrigerator)
Operation of AWN (1:active/0:inactive)
Abwärmenutzung: Betrieb AWN Stufe 1
Abwärmenutzung: Betrieb AWN Stufe 2
Abwärmenutzung: Betrieb AWN Stufe 3
Pump activity (1:active/0:inactive)
Wärmeverteilung: Pumpe Betrieb =5200M1
ventilation
Ventilation bypass (%)
Wärmeverteilung: Ventil AWN =5710Y1
pump
Pump activity (1:active/0:inactive)
Abwärmenutzung: Pumpe Betrieb =6200M1
temperature (C)
W08 Wärmebezug ab AWN: RL-Temp

Fig. 2: Diagram representing the heating system of the supermarket

first Monday and the last data point is the last hour of the last Sunday This step is important as we want to perform a weekly analysis and we can't afford to have shifts.

At the end of the preprocessing, from the 634254 datapoints per feature, we were left with 28225 datapoint per feature.

V. Time series representation

As stated previously, our data unit is a week of measurements. After preprocessing the data, one week corresponds to 168 hourly measurements of 30 features. As the number of data points is relatively low, n = 168, the number of measurements per data point had to be reduced [1]. For this, we explored two possible representations of time series: a simple representation, serving as base line and a more complex representation, making use of polynomial approximation.

a. Simple base line: Mean and Standard deviation representation

With the intention to implement a baseline for time series representation, and subsequent analysis, a weekly resampling of the data was performed, generating for every feature and every week, four values: maximum, minimum, mean and standard deviation. Plots showing this resampling can be seen in the notebook *Simple_data_point_representation*, and two of these plots are shown in figure 3. As we can see in this figure, assuming that reducing the data to four weekly measurements per feature is a good time series representation, reducing it to only two (mean and standard deviation) is expected to be as well. In fact, we see in figure 3 A that the standard deviation captures the information carried by the minimum and maximum, when joined to the measurement of the mean: it varies according to the relative spread of the minimum and maximum around the mean. In the same way, it can be seen in figure 3 B, that when minimum and maximum are at a constant distance from the mean over a time interval, the standard deviation remains constant as well. We obtain a simple representation of time series by which 168 (hours/week) x 30 (features) = 5040 measurements per week, reduce to 2 (measurement of mean and standard deviation) x 30 (features) = 60 measurements per week.

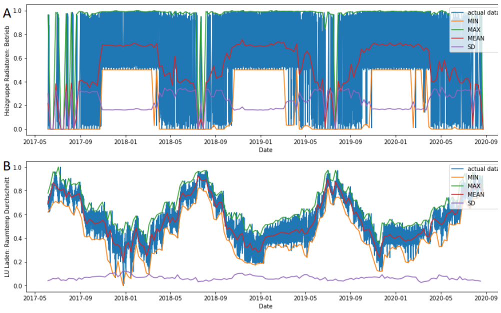

Fig. 3: Plots showing the example of two features over the complete time span (168 weeks). Over this, the minimum, maximum, mean and standard deviation of the values of the features.

b. Polynomial representation

In this section we propose a polynomial representation of each week: each week was associated with a list containing the coefficients of the polynomial that best approximates it.

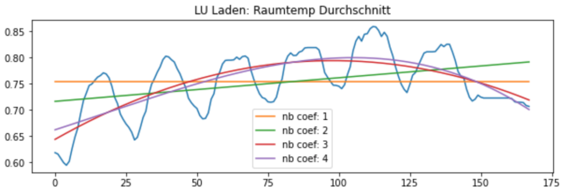

Fig. 4: Example of polynomial approximation in one week for different number of coefficients

We tried from degree 0 to degree 4 as degree of the polynomial, that is, resulting in 1 to 5 coefficients. This ultimately influences the number of features per week, being 30 with 1 coefficient and 5x30 = 180 features for 5 coefficients. The limit was reached with 5 coefficients because the PCA needs to be performed with more datapoints than features [2].

To evaluate the polynomial approximation the R2 was computed for each week comparing the original value to its polynomial representation and reported the results in the table below:

nb. of coef.	nb. of features	R2	points per feature
1	30	0.06	168
3	90	0.25	168
5	150	0.32	168

TABLE 1: INCREASE IN NUMBER OF FEATURES AS THE NUMBER OF COEFFICIENTS RISE

c. Further explorations

Another time series representation that was not explored here, makes use of Fourier analysis. Indeed, it would be interesting to investigate whether the data for every feature and every week can be approximated by some of the first harmonics obtained by the discrete Fourier transform. That is, see if a week can be represented by a limited number of Fourier coefficients, similarly to how it is done with the polynomials.

VI. ANALYSIS

After establishing two appropriate time series representations, we proceeded with a more in-depth analysis of the data. As it is common in machine learning tasks with a great number of numerical features, a principal component analysis was performed, followed by the application of three clustering methods. A correlation matrix was also generated and provided useful information.

a. Correlation matrix

In order to contextualise and interpret the results that are addressed in the next sections, We computed the correlation matrix between the 30 features as well as the covariance matrix. We observed 9 pairs of features with a correlation greater than 0.9, every pair was identified in the supermarket diagram shown in figure 2 where for each pair we found a logical explanation.

In further work on this data set it would be interesting to come back to the correlation matrix and discuss the correlations in relation to what is observed in the plots that show the superposition of each cluster, this operation could lead to a reduction of the number of features and thus change the performance of the model. The mentioned plots appear in figure 7

b. Principal component analysis

A PCA was performed for each of the two time series representations. By looking at the percentage of variance in the data explained by the principal components, we chose the first three for both representations.

In the case of the base line representation, 78.5% of the variance can be explained by the first three principal components, while only 68.6% can be explained by the first two components. Figure 5 shows the data plotted in the space defined by the three principal components. By looking at this figure, one can already distinguish areas with higher density of data points, which will probably result in clusters.

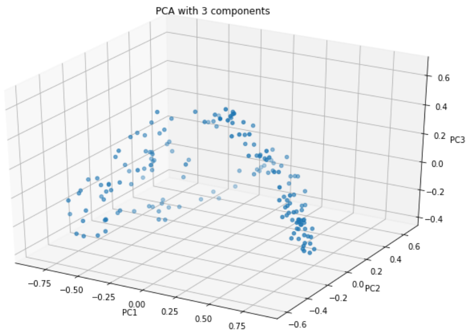

Fig. 5: Plot of the n = 168 data points with simple base line representation of time series. The three dimensional space is determined by the first three principal components obtained by PCA.

For polynomial representation of the time series, we also chose the first three principal components, since depending on the degree of the polynomial, they allow to explain between 77.4% and 82.7% of the variance. More detailed numbers can be found in the notebook *Polynomial_representation_approximation*.

c. Clustering

The main clustering methods that were explored are K-means clustering [3] and density based clustering (DBSCAN) [4]. The main difference between these two is that with K-means clustering, we need to know the number of clusters prior to applying the method, while with DBSCAN this is not needed.

DBSCAN [5] requires the minimum number of points in each cluster, as well as the maximal distance between two data points, such that they can be considered neighbors. This parameter, called epsilon, has to be chosen carefully.

In our case, we chose the minimum number of data points in one cluster to be 6, twice the number of dimensions after the PCA [4]. However, for the choice of the optimal epsilon, we used the k nearest neighbors method [6]. As shown in notebooks *Simple_data_point_representation* and *Polynomial_representation_approximation*, we computed the mean distance between each point and its 6 nearest neighbors and plotted this. We then graphically chose epsilon to be the mean distance at which we identify an elbow, that is, when this mean distance between the 6 nearest neighbors explodes.

Applied to the simple base line representation of time series, we find an optimal value for epsilon of 0.18. Then, DBSCAN yields a division of the data points in 6 clusters, with a total of 13 outliers. This result is visually assessed with figure 6.

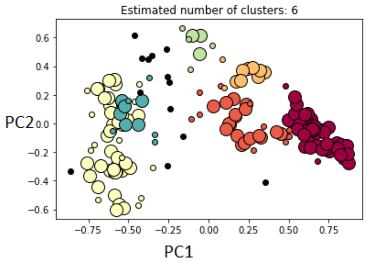

Fig. 6: Result of the application of DBSCAN to the data with simple base line representation. Data points are plotted in a two dimensional space defined by PC1 and PC2 and colored according to their clusters of belonging.

The same steps are applied to the polynomial representation of the time series, and we find here an optimal epsilon of 0.2. DBSCAN results here, respectively for a degree 1, 2, 3, 4, and 5 of the polynomial, in 2, 6, 3, 2 and 4 clusters.

Since K-means clustering requires the number of clusters, it was run with several different values of k. To evaluate the quality of the clustering with each of the values for k, we computed two measures: 1) the average silhouette score [7], a number in the interval [-1; 1] which indicates how distinct in space the clusters are from each other. 2) The mean distance between each point and the center of its cluster, which is also an indication of the density of the clusters. Knowing this, when applied to the simple base line representation of time series, we found that k = 6, seems to be appropriate. This is in coherence with the results found with DBSCAN.

When looking at the polynomial representation, it depended on the degree of the polynomial. For degrees 1 and 2, k = 2 clusters seems to be appropriate, for degree 3 and 4, k = 3 clusters seems the better choice, this choices where made using the average silhouette score that was computed for each polynomial representation and for k spanning from 2 up to 6.

As an additional visualization tool, we plotted each feature over a time span of one week, separating the data points per clusters. These plots can be found in notebooks *Simple_data_point_representation* and *Polynomial_representation_approximation* and figure 7 shows an example. We see similar profiles, indicating the same general behaviour over one week for all data points. However, depending on the feature, there is a clear distinction (a shift in the plots) between points in different clusters. This is a significant validation of our clustering methods.

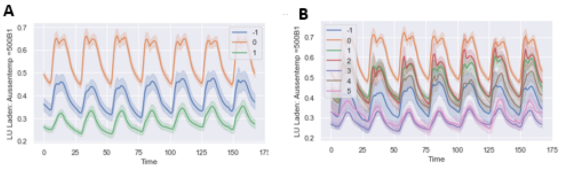

Fig. 7: Plots showing for two features, the superposition of all data points on a time interval of one week. Different colors indicate the different clusters. On these two examples, a clear separation of the data is visible according to the cluster of belonging. Figure A is obtained with the polynomial representation of time series, figure B is obtained with the base line representation of time series.

VII. Conclusion

In conclusion, we were able to apply unsupervised learning methods to our data set, that led to promising results for clustering of this data. Indeed, plots like the ones shown in figure 6 and especially figure 7 give us reason to believe that we are investigating in the right direction. Not only were we able to observe a clear separation between clusters, but we were also able to see a similar behaviour between the two methods of representation of time series.

Although some of our steps such as the correlation matrix or the technique for representing the time series, could be further investigated, we can already cite some encouraging future direction to explore. One of these is creating a sort of calendar, where we would mark the cluster of belonging for every week. In this way, we could try to understand if there is a seasonal dependence intra clusters. We could also investigate in which clusters we find some of the "particular weeks" of a year, that is, those with national holidays for example.

The ultimate step would be of course to understand what makes a week faulty, and try to test our clustering with it.

References

[1] S. Aghabozorgi, A. Seyed Shirkhorshidi, and T. Ying Wah, “Time-series clustering – a decade review,” vol. 53, pp. 16–38. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306437915000733

[2] PCA: Principal components decomposition. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[3] I. Dabbura. K-means clustering: Algorithm, applications, evaluation methods, and drawbacks. [Online]. Available: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa0

[4] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, “DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN,” vol. 42, no. 3, pp. 1–21. [Online]. Available: https://dl.acm.org/doi/10.1145/3068335

[5] sklearn.cluster.DBSCAN — scikit-learn 0.23.2 documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

[6] DBSCAN: Density-based clustering essentials. [Online]. Available: https://www.datanovia.com/en/lessons/dbscan-density-based-clustering-essentials/

[7] sklearn.metrics.silhouette_score — scikit-learn 0.23.2 documentation. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html#examples-using-sklearn-metrics-silhouette-score