Pattern extraction and profiling of historical water network demand patterns

submitted in partial fulfillment for the degree of master of science

Marten Staghouwer
12851868

Master Information Studies
data science
Faculty of Science
University of Amsterdam

Submitted on Fill In The Date In Format DD.MM.YYYY

	UvA Supervisor
Title, Name	Viktoriya Degeler
Affiliation	UvA Supervisor
Email	v.o.degeler@uva.nl

ABSTRACT

Write your abstract here.

KEYWORDS

pattern extraction, clustering, water network, time series

GITHUB REPOSITORY

https://github.com/Mjnstag/IS-thesis-pattern-extraction

1 INTRODUCTION

2 RELATED WORK

3 METHODOLOGY

There have been different uses for clustering in research related to water networks, from using clustering to calibrate models to dividing water networks in different sectors [2, 3]. This paper builds on already existing use-cases of these methods to propose a framework in which clustering methods are used to extract patterns from historical water network data. The goal is to find out if this use-case of clustering if viable and feasible in predicting changes and trends in water networks. For this, multiple different unsupervised clustering algorithms were used in an experimental setup where the model's hyperparameters were automatically optimized and the final results compared against each other and a base model.

3.1 Experimental Setup

3.1.1 Data Gathering.

For this paper one dataset was used. The dataset comes from Vitens[1], a Dutch water network company, and contains data about a subset of the total water network they manage. This dataset is the one primarily used and tested with. The dataset has 17 columns and mainly contains sensor data from a few different points of the network. The measurement interval for the sensors was 1 minute, which when combined with a collection period between 2017 and 2022, results in a total of 3.067.200 rows.

3.1.2 Data Description.

The complete data set consists of 17 different columns which can be divided into 4 different classes.

- Rapportage|Levering|Target is the value of the instantaneous consumption used by an automatic meter reading (AMR) customer in m3/h
- Restpatroon|Demand-patroon|Target is the demand pattern of a certain area or sub-area. It is calculated by dividing the instantaneous delivery to the area minus the AMR measurements by the average annual consumption of the customers present. It is dimensionless.
- Volumestroommeting|meetwaarde|Target is the instantaneous flow at a given time in m3/h
- Drukmeting|Head|Target is the head of the output pressure of a pumping station. The head is calculated by adding the outlet pressure in meters to the height of the pumping station or measurement. It is used for the "Total Head" field in the EPAnet [1] reservoir.

The data describes 9 different locations within the network, both consisting of pumps, reservoirs, and AMR sensors. Not all of these classes fit with each of the locations.

3.1.3 Data transformation.

Even though the dataset has 17 columns, not all of them are useful and contain irrelevant data. This is due to the dataset also being used for other projects and purposes. After filtering out these columns, 14 columns remained.

Due to the data having the Central European Time (CET) timezone, the data was first converted to Coordinated Universal Time (UTC) to avoid conflicts with regards to daylight saving time.

	Amount of missing Values
Column 1	17
Column 2	470
Column 3	572210
Column 4	0
Column 5	0
Column 6	0
Column 7	0
Column 8	3694
Column 9	0
Column 10	0
Column 11	0
Column 12	70475
Column 13	0

Figure 1: Amount of missing values per column.

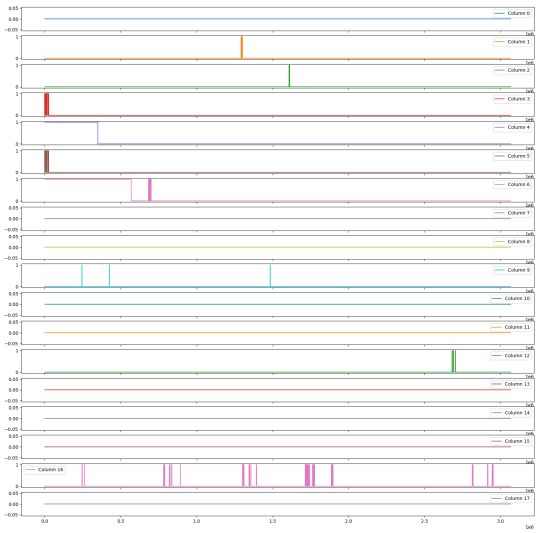

Figure 2: Missing values in the data where a spike means a missing data point.

Throughout the data there are many instances of missing values. Following the EDA, 2 different kinds of missing data were found:

[1] https://www.vitens.nl/

	count	mean	std	min	25%	50%	75%	
GEN-08 Ede\Pompstation Edese bos\Straat 03\Flow	3067183.0	217.412539	93.146601	-46.993860	145.403746	240.364583	296.994087	48
GEN-09-01 Oosterbeek-hoog\Fletcher BV|Supplied	3066730.0	0.797213	0.761748	-49.062112	0.324000	0.700000	1.020000	6
GEN-09-01 Oosterbeek-hoog|Demand-pattern	3050079.0	1.076306	0.533670	-0.166241	0.628070	1.089339	1.409961	
GEN-09-02 Oosterbeek Wolfheze|Demand-pattern	2717894.0	1.370534	1.083315	-36.129348	0.695174	1.320137	1.951666	4
GEN-09-03 Oosterbeek-laag|Demand-pattern	3050088.0	1.277443	0.558586	-1.305478	0.801073	1.325713	1.641878	
GEN-09-04 Renkum-Heelsum\Renkum|Supplied	2494990.0	0.630828	0.976893	-40.779463	0.040000	0.276000	0.590000	6
GEN-09-04 Renkum-Heelsum\Parenco BV|Supplied	3067200.0	3.546196	2.630897	-0.319255	1.720000	2.370000	4.930000	2
GEN-09-04 Renkum-Heelsum\Zorggroep|Supplied	3067200.0	1.062430	0.818336	0.000000	0.560000	0.960000	1.370000	2
GEN-09-04 Renkum-Heelsum|Demand-pattern	3067128.0	0.396411	0.678777	-0.463697	-0.076914	0.000000	0.980895	
GEN-09 Pompstation Oosterbeek\Straat 01\Head	3067200.0	86.532197	1.303967	14.793348	86.467548	86.613930	86.766028	9
GEN-09 Pompstation Oosterbeek\Straat 02\Head	3067200.0	69.392722	0.908778	12.021875	69.026325	69.576654	69.770920	8
GEN-09 Reservoir Doorwerth\Straat 01\Flow	3063506.0	3.620681	14.658989	-1186.451290	-0.233684	0.029438	0.430931	61
GEN-09 Reservoir Doorwerth\Straat 02\Head	3067200.0	54.792932	0.757743	51.986275	54.370048	54.587694	54.779361	6
GEN-09 Reservoir Doorwerth\Straat 03\Flow	3067200.0	96.509647	65.726829	-7.696875	7.457190	100.438377	150.099394	71
GEN-10 Pompstation Wageningseberg\Head	3067200.0	54.714287	30.496846	14.850000	53.789785	54.583858	55.442872	654
GEN-10 Pompstation Wageningseberg\Flow	2996725.0	5.854928	23.618150	-16.796117	0.000000	0.000000	0.000000	35
GEO-06 Pompstation La Cabine\Head	3067200.0	66.445752	1.878207	23.600000	65.366451	66.321320	67.492561	11

Table 1: Data description of all columns

Leading missing data and sporadically missing data throughout the entire length of the time series, see image 2.

In both of these cases an attempt was made to fill the missing values by interpolating them linearly. To not change the behaviour of the data too much, a limit of 20 was set on how many consecutive missing values could be interpolated. In the case that there were more than 20 consecutive missing values, these extra missing values were not interpolated and filled.

To prepare the data for clustering, the columns were split into new data frames. This resulted in 14 new data frames where there was 1 column which contained all that column's rows from the original dataset. To enable clustering, for each of these data frames, the rows were split into segments of 1 day. Each of these segments were added as new columns resulting in a dataframe with 1440 rows of sensor data and an amount of columns based on the amount of data being transformed.

After further splitting the dataframes, all columns were checked if there were any missing values remaining. If this was the case, that column was removed from the dataframe and not used.

Finally, one of the limitations of the models are the potential computational resources needed to use the models. To lessen this concern, the amount of data in the final data frames was reduced by only having a data point 5 minutes instead of every 1 minute. This was done by taking the first of these datapoint while discarding the rest. This limits the data needing processing and thus reducing the computational resources needed for running the different models. After this, the remaining columns were then scaled using a min-max scaler.

3.1.4 Dimensionality Reduction and Clustering Methods.

As mentioned, the goal of this paper was to find out if clustering algorithms can help in extracting patterns in water network data. This was done by using and testing different clustering algorithms like: Principal Component Analysis (PCA), K-means, Self-organising maps (SOM), and DBSCAN. These algorithms were used through python's SKlearn [5], Minisom [8], and Tslearn [7] libraries Due to this, it is not needed to create these algorithms from scratch.

After clustering the data, an analysis was done on the composition and the distribution of the final clusters. This was done by looking at the distribution of columns per cluster and, if needed, filtering cluster which consists of only 1 column due to outliers in the data.

3.1.5 Evaluation.

To evaluate and compare the different clustering algorithms, a base algorithm was chosen: K-means. This algorithm was chosen due to its widespread usage and its simplicity. To evaluate the baseline and other chosen algorithms, multiple different aspects will be used. This firstly includes the silhouette score of the different clusters following P. J. Rousseeuw's method of mathematically scoring clusters, eliminating some biases and human errors [6]. To do this, most of the libraries used include an in-build method for calculating this score. For those libraries who do not have this functionality, an additional python library called SOMperf [4] was used.

To train, test, and validate the models, the data is split in 80% of the data in the training set, 10% in the validation set, and finally 10% used for the test set. The data being split is comprised of the new columns created from the complete dataset.

In addition, other methods of forecasting have been compared to using clustering such as R's timeseries regression models.

4 RESULTS

5 DISCUSSION

6 CONCLUSION

REFERENCES

[1] [n. d.]. Vitens/epynet: Object-oriented wrapper for EPANET 2.1. https://github.com/Vitens/epynet. (Accessed on 02/08/2023).

[2] Xinran Chen, Xiao Zhou, Kunlun Xin, Ziyuan Liao, Hexiang Yan, Jiaying Wang, and Tao Tao. 2022. Sensitivity-Oriented Clustering Method for Parameter Grouping in Water Network Model Calibration. *Water Resources Research* 58, 5 (2022), e2021WR031206.

[3] Armando Di Nardo, Michele Di Natale, Carlo Giudicianni, Dino Musmarra, Giovanni Francesco Santonastaso, and Antonietta Simone. 2015. Water distribution system clustering and partitioning based on social network algorithms. *Procedia Engineering* 119 (2015), 196–205.

[4] Florent Forest, Mustapha Lebbah, Hanane Azzag, and Jérôme Lacaille. 2020. A Survey and Implementation of Performance Metrics for Self-Organized Maps. arXiv:2011.05847 [cs.NE]

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[6] Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

[7] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and Eli Woods. 2020. Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research* 21, 118 (2020), 1–6. http://jmlr.org/papers/v21/20-091.html

[8] Giuseppe Vettigli. 2018. MiniSom: minimalistic and NumPy-based implementation of the Self Organizing Map. https://github.com/JustGlowing/minisom/

Appendix A FIRST APPENDIX

Put your appendices here.