PROJECT DELIVERABLE REPORT

NAIADES

Greening the economy in line with the sustainable development goals

D4.7 – AI empowered critical water consumption monitoring toolkit – Mid-term

A holistic water ecosystem for digitisation of urban water sector

SC5-11-2018

Digital solutions for water: linking the physical and digital world for water solutions

"This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 820985"

Document Information

Grant Agreement Number	820985	Acronym	NAIADES
Full Title	A holistic water ecosystem for digitization of urban water sector		
Topic	SC5-11-2018: Digital solutions for water: linking the physical and digital world for water solutions		
Funding scheme	IA - Innovation action		
Start Date	1stJUNE 2019	**Duration**	36 months
Project URL	www.naiades-project.eu		
EU Project Officer	Alexandre VACHER		
Project Coordinator	CENTER FOR RESEARCH AND TECHNOLOGY HELLAS - CERTH		
Deliverable	D4.7 – AI empowered critical water consumption monitoring toolkit – Mid-term		
Work Package	WP5 - NAIADES Smart Framework: AI analytics and predictive services		
Date of Delivery	**Contractual** M18	**Actual**	M18
Nature	R - Report	**Dissemination Level**	PU-PUBLIC
Lead Beneficiary	Jozef Stefan Institute		
Responsible Author	Matej Čerin	Email	matej.cerin@ijs.si
		Phone	
Reviewer(s):			
Keywords	Data Fusion, Machine Learning, Incremental Learning, State Identification		

Revision History

Version	Date	Responsible	Description/Remarks/Reason for changes
1	2/11/2020	Matej Čerin	ToC written.
2	10/11/2020	Matej Čerin	Data sources, Analysis methods
3	13/11/2020	Matej Čerin	Experiments, Conclusion
4	19/11/2020	Matej Posinković, Klemen Kenda	Internal review and additions
5	21/11/2020	Matej Čerin, Matej Posinković	Experiments chapter update.
6	27/11/2020	Julian Bruns, DISY, Cederic Crettaz, MI	Naiades internal review.
7	27/11/2020	Matej Posinković, Klemen Kenda	Internal review feedback incorporation.
8	19/5/2021	Matej Posinković	Full title included on the front page.

Contents

1 Summary

This document describes work done in task T4.4 on artificial intelligence empowered critical water consumption monitoring.

Observation of a complex system via a large set of time series is not an easy task. Artificial intelligence offers us tools to analyze the data that is not comprehensible to a human operator and merge it into a more compact form. To do this, we propose the usage of machine learning clustering algorithms to analyze various typical states of the observed system and Markov chains to model the transitions within that system. Proposed tools produce results on multiple levels, which means that the system can be modeled in a generalized way or explored in more details. The analysis outcome is a visualization of the input data, helping an expert to better understand and interpret the behavior of the observed system.

2 Introduction

This deliverable aims to present the methodologies and tools that will be used in monitoring of water supply. The tools described in this deliverable will allow to monitor data more easily and efficiently. The description of the task from the grant agreement is:

> **T4.4 AI empowered critical water consumption monitoring (from the Grant Agreement)**
>
> Main purpose of the task is to build an analytical model based on a large set of time series (i.e. consumption, alarms, etc.), to understand critical water consumption states. The task will transform each point in time into a typical state of the system (by hierarchical clustering). Furthermore, analytical framework developed will provide hierarchically ordered states (i.e. user behaviour at 20:00 in a particular household on a typical evening could be divided into more states – i.e. washing machine is on, children are in the bathtub or there has been a leakage in the toilet). Based on Markov chains, the models developed will enable assessment of risk for water system to transform into an alarm state (i.e. high water consumption peaks during drought, etc.). In addition, the system will calculate transformation matrices between different states and provide advanced visualization tools. The water system operator will therefore be able to visualize the state of the system, detect the problematic states and design interventions based on the information provided.

The overall objective of this deliverable is to describe the methodologies and tools that will be used in water supply monitoring and present initial experiments on Braila use case. This work is strongly based on our previous work [1].

The document is structured as follows: Section 3 introduces a reader to the task objectives, Section 4 describes the data that has been used for modelling. In Section 5, the reader is acquainted with analysis methods, which is followed by description of conducted experiments in Section 6. At the end conclusions are presented along with the future work.

3 Objectives

The main purpose of the task T4.4 is to create a generic analytical model based on a big amount of time series data, that will help us understand the behaviour of the system on a higher level. Even though we'll be focusing on water flow and pressure data for the city of Braila, district Radunegru, the system will allow any water utility to model its data. A special emphasis will be given to detect the critical states that would warn the stakeholders about the possible upcoming alarm states, such as higher-than-normal consumption of water. The system at a particular time stamp will be represented with a feature vector, based on multiple time-series. Using clustering, the different feature vectors will be dynamically grouped into typical states. Furthermore, Markov chains will be able to produce the hierarchically ordered states. The system will be also able to produce transition matrices that will describe the probability of the system changing from one state to another. Another important feature of the developed system is also the advanced visualization tools that are able to visualize the state of the system, which is helpful in data interpretation.

Visualization is paramount in this system since it enables an domain expert user (e.g. water utility employee) a more detailed introspection into the behaviour. In addition to the graph of the system and to the transition data, the system will provide tools that will enable user to understand the conditions of a particular state. Such tools are, for example, histograms of particular variables in the feature vectors or decision tree models, which represent the conditions on how a system could end in a particular state.

In order to evaluate identified states (e.g. as normal or anomalous) a human input is required. Only after evaluting states with a more comprehensive labels, we are able identify probabilities for a system transition into a, e.g., unwanted state. In such a way is possible to train a machine learning model (or simply estimate probabilities from the Markov chain) which could trigger a warning that the system is headed for an unwanted event.

Furthermore, if the system encounters a state, which has not been identified in the previous data, it could inform a domain expert user, who could investigate the event. For such a purpose, the system would need to be able to operate on the real-time data. To do so, it could be associated with the analytical solutions presented in NAIADES deliverables D5.1 and D5.5, which provide the infrastructure on streaming analytics and incremental learning.

The task T4.4 consist of one use case:

- Braila use case – The main objective for the Braila use case UCB1, described in more detail in D2.5, is to identify various states within the pipeline system. Through state identifications, we aim to identify critical states and those states, that would lead to a critical state.

4 Data sources

4.1 Use case Braila

All the available data is currently available in the form of files (not from a live system). Currently, a JSON file includes a couple of sensors (248 and 249) in JSON format, as presented in Figure 1.

```
{
"timeStamp":"2018-11-23 09:53:00",
"idflowmeter":"MAG8000_024905H318",
"Tot1":49.18,
"Tot2":0.0,
"Analog2":1.1
}
```

Figure 1: Typical data row for Braila dataset.

Parameters in the data set:

- "`timeStamp`": is the date-time of the measure.
- "`Idflowmeter`": is the ID of the meter
- "`tot1`": is the (cumulative) water flow in m3/h the direction of consumers (household); flow that is directed towards Braila district;
- "`tot2`": is the (cumulative) water flow in m3/h in the opposite direction;
- "`analog2`": is the water pressure of the measuring point (Figure 2).

Data is available from: 2018-11-23 – 2020-04-01. The frequency in the dataset is 1 minute, however some portions of the data are missing.

Missing data. There are 1395745 records in the dataset, among those 25 values for `analog2` are missing, 5086 for `tot1` and 100635 for `tot2`. 710925 records belong to sensor 249, and 684820 records to sensor 248. Sensor 248 has start data of the measurements on 2018-12-07. It looks like that sensor 248 misfunctioned through the measurement time, so the analysis should be accomplished on sensor 249. Nevertheless, we were able to use data for whole year of 2019.

Initially, not much feedback has been given regarding the data values, so the exploratory analysis is done in a "blind" mode; data-driven only.

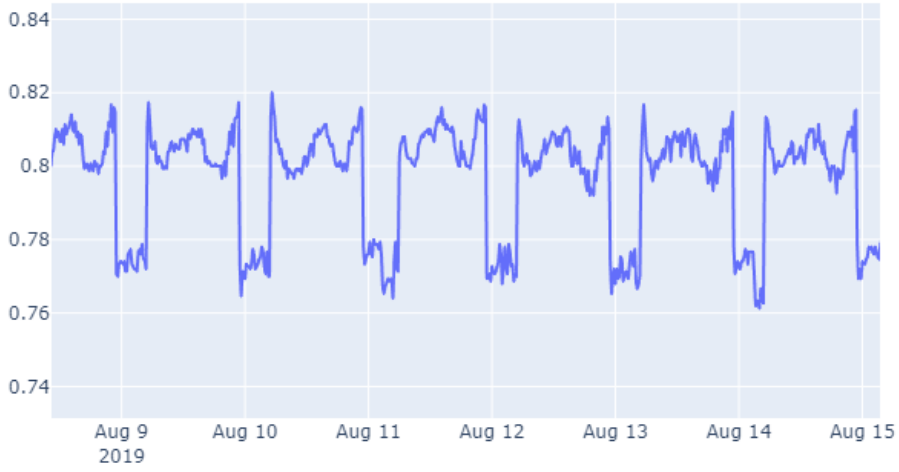

Figure 2: Daily pressure cycles of Braila data (Analog2, sensor 249), excerpt.

For analysis, all the data has later been resampled (by averaging) to an hourly resolution. Results are shown in Figure 3. We can observe that the tot1 and tot2 values include a cumulative consumption of water. In order to help us with the analysis, these values should be converted into a derivative as it can be observed in Figure 4. The conversion of cumulative values into a time differences is compulsory as any supervised machine learning algorithm will probably return unusable results as cumulative value distribution changes constantly.

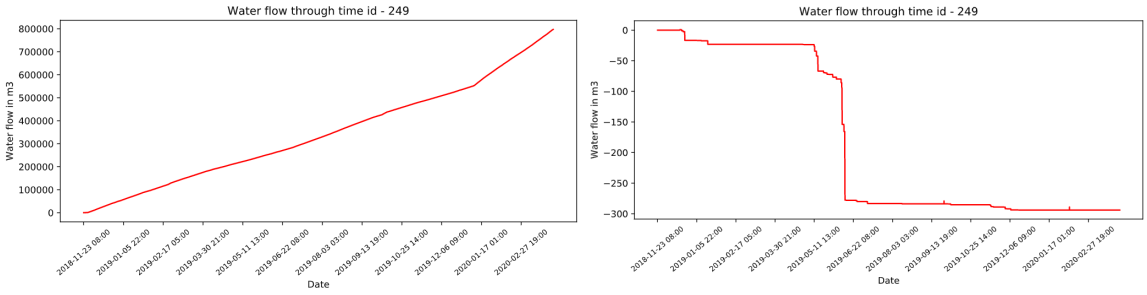

Figure 3: Water flow through time (tot1 and tot2 cumulative values).

In Figure 4, we can observe that the differences generate a timeseries, that makes some sense. In the example below, we can observe a significant jump in the consumption in the beginning of 2020. We can also observe some smaller jumps around 2019-03-01 and 2019-10-01. We can also observe particular days with extremely high consumption. This way we have identified 3 potentially interesting states of the system. Additional parameters that might be interesting are also daily consumption interval as we see that the band representing the measurements is sometimes thicker. Perhaps a moving standard deviation of the measurements might represent a good feature to capture such behaviour.

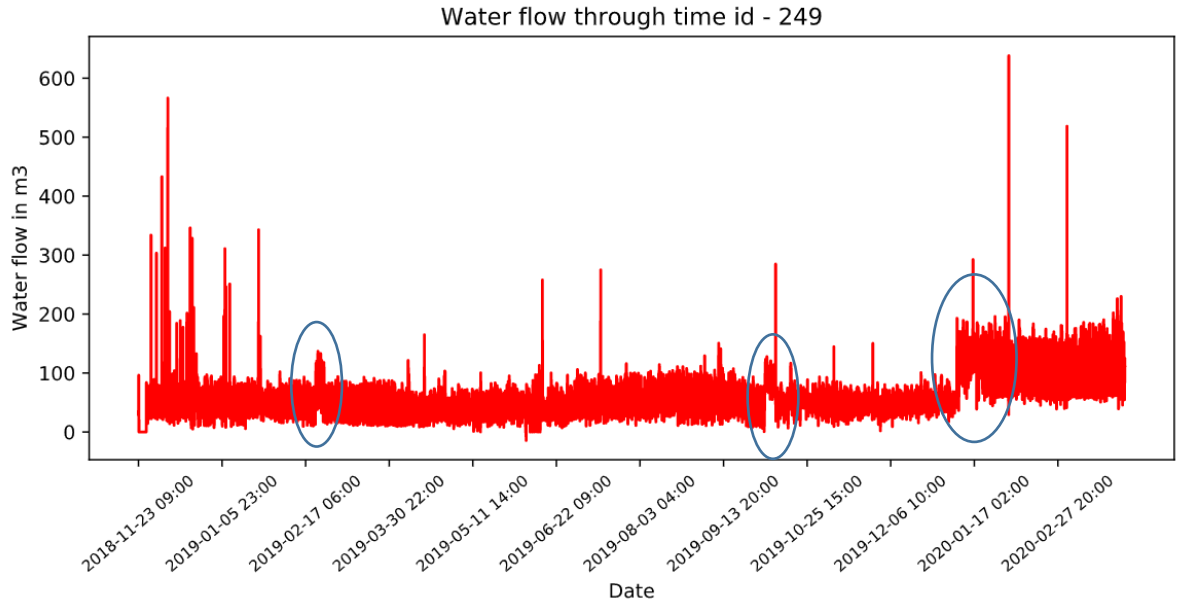

Figure 4: Water through time (for tot1, daily consumption).

In Figure 5 we can observe all 4 different timeseries plotted on the same time scale. The timeseries represent a period of 200 hours. We can observe that there exists a correlation between the pressure values (denoted as analog2) and hourly water flow (denoted as tot1diff), although this does not necessarily show causality. Drop in the pressure might be associated with an operator action (reduce pressure to reduce water loss).

In Figure 6 we wanted to show the correlation between water flow (tot1 diferences) and water flow in opposite direction (tot2 differences). We can notice, that, in order for tot2 to have any non-zero values, tot1 needs to fall to zero. In other words: water flow needs to stop in order to have flow in opposite direction. Obviously, we can't have both, as we are measuring dynamics at one point in one single pipe.

Correlations can be better understood through correlation heatmap matrix in Figure 7. We observe that there is a strong negative correlation between tot1 and tot2, which is understandable, since tot1 appends outgoing water flow and tot2 subtracts incoming water flow. Correlation between tot1 and tot1diff is around 0.45, which is higher than the correlation between analog2 and tot1diff, which we observed from Figure 5. This can be explained with the fact that the overall consumption is rising through time (like tot1), however, pressures stay constant over time.

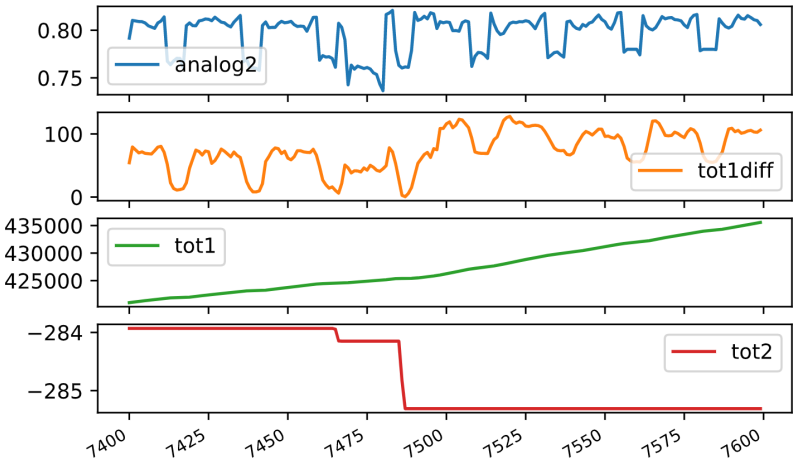

Figure 5: Visualization of various time series from the dataset (analog2 – blue, total consumption tot1 difference - blue, tot1 cumulative - green and tot2 cumulative – red).

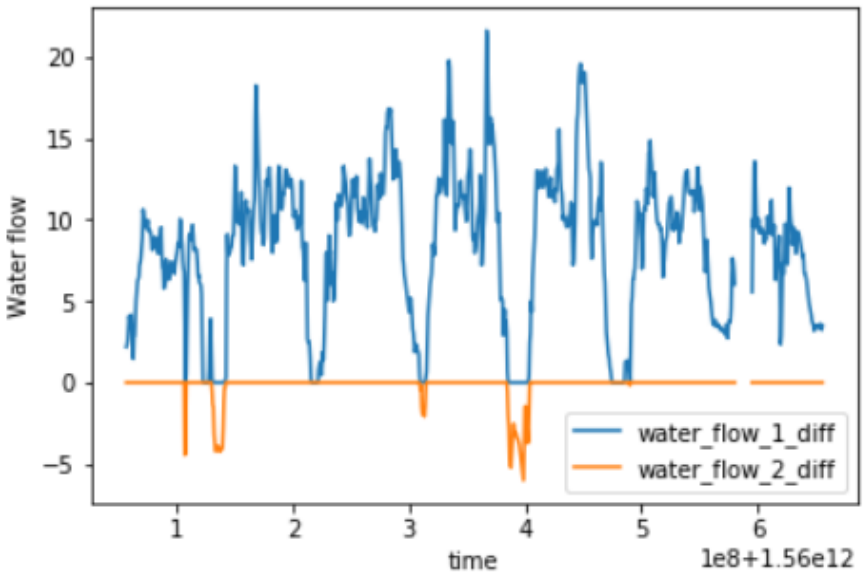

Figure 6: a tot1and tot2 difference. Tot1 data denotes water flow, tot2 data denotes water flow in opposite direction. The entanglement of tot1 and tot2 is clearly visible from the chart: tot2 can have (negative) values only in case when tot1 falls to 0.

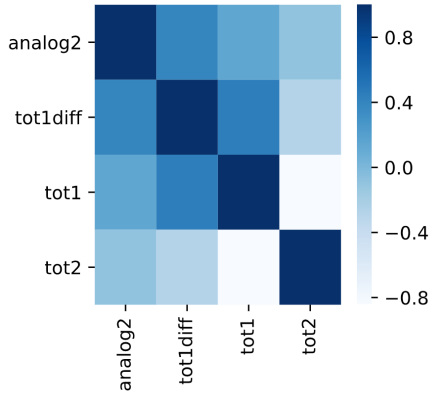

Figure 7: Correlation matrix of the raw features.

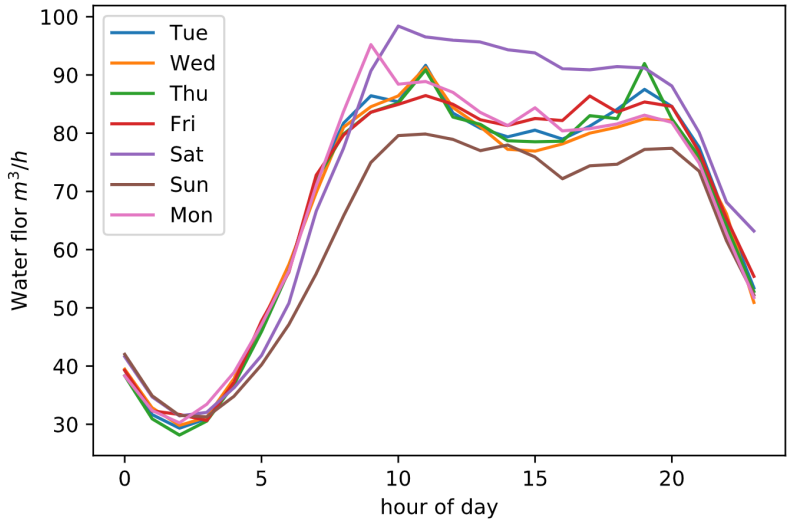

Figure 8: Daily profiles of water flow differences.

In Figure 8 we can observe daily profiles of water consumption. As expected, we observe that Sunday (brown) and Saturday (violet) profiles differ from other daily profiles. Saturday water consumption seems to be higher than on the other days and also more constant during the day. Higher consumption could be attributed to the fact that people usually do not leave their households and perhaps that Saturday is dedicated to the domestic work. Sundays, as expected, exhibit lower water consumption. There is also no visible morning "delay" during the weekend. Standard deviation around the values is pretty high, which can be attributed to the concept drift in the data – the distribution changes through time.

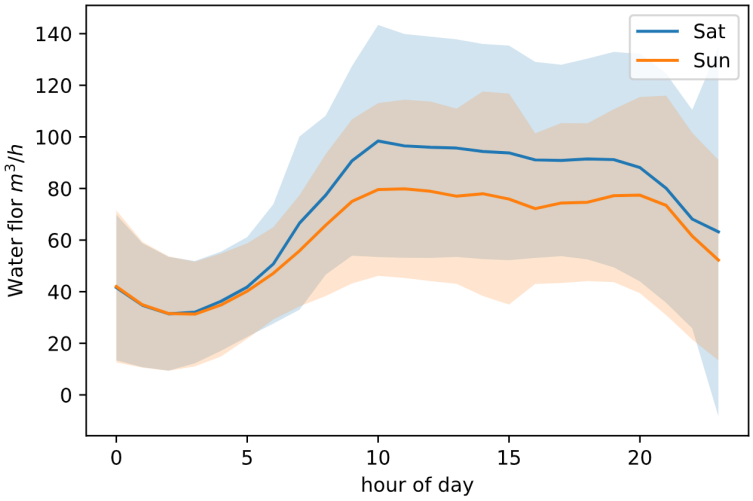

Figure 9: Water flows during the weekend with the standard deviations.

In the figure above, we can observe that concrete daily profiles for a particular day can differ significantly. Also – we can see there are particularly interesting phenomena for example (on Saturday at 23:00 and Sunday at 13:00 and 14:00). Those might be interesting for an expert user to analyse in more detail.

Based on this initial data analysis some of the good features for state detection could be:

- hourly consumption and hourly pressure (to detect day/night dynamics) and peculiarities
- daily average of water consumption and standard deviation (to detect time of day)
- weekday/weekend – to detect two different modes, perhaps even Saturday/Sunday/weekday
- daily trend (difference from day before) – to detect sudden jumps in the data

5 Analysis methods

Within chapter 5, we'll first describe models, behind the state-detection analysis. Then, we'll go through the process of defining various states and end with a visualisation example.

5.1 Feature engineering

Feature engineering is the process used in data mining used to create new features from raw data. The features are often derived from domain knowledge, where experts already know what kind of features typically affect a given situation. Typically, the data features within a machine learning Internet of Things (IoT) based system emerge from various types of sources: IoT based sensor time-series data, external sources (such as current weather conditions data), prediction models (such as weather prediction data) and static properties (such as any anthropogenic data, for example – hour of day, day in a week, etc.).

Furthermore, any of these time series could be aggregated and transformed in various ways in order to obtain relevant features. For example, if we are observing groundwater levels, we would expect that they are dependent on the rainfall, however – there is a delay between the time of precipitation and the time this water reaches the groundwater – so a delay should be introduced. We can also observe that rainfall can be subject to rapid change within a short time period, while the groundwater levels have a more conservative response; this hints on the usage of an average of rainfall over multiple days, shifted for a typical time interval.

In our initial experiments, we have tested various properties of the Braila input time series data. We estimated that the current raw data might not give us enough information about the process. Thus, it is important to calculate the aggregates of features in a time window.

Two types of features have been calculated:

- Shift for n days – The first new feature that we added to the feature vector is the data shifted from the past (meaning for example water flow from one week ago). In experiments, we used data shifted for up to 7 days.
- Mean over last n days – Those features capture the mean value of feature over last *n* days. In our experiments, we used the mean in the 7 days time window.

In both feature cases, we have presumed, that the past data affect the present one. Therefore, we include these types of calculated data to original data instance.

5.2 Clustering

Clustering algorithms belong to the family of machine learning algorithms. More specifically, they belong to the family of unsupervised machine learning techniques. As opposed to supervised learning, unsupervised learning does not require the data to be labelled (for example, any data point does not have the identification of the state it belongs to). It means, that this additional structure, that might exist within the data, should be deduced by the algorithm. Among unsupervised methods we can also find anomaly detection, which must discover unusual states in the data. It is up to the user to define, what is an anomaly [5].

Clustering is the process of combining similar data points into groups where data in the same group are similar. The clustering is not a single algorithm, but a family of different algorithms. Clustering algorithms are unsupervised learning algorithms. Unsupervised algorithms are used in cases where the true label of data instances is not known, therefore we group similar instances of data together. In the work presented in this paper, we use two of the clustering algorithms (K-Means and DP-Means). In the future, there are other options to be considered, especially regarding the evolving nature of data streams. A brief overview of the incremental clustering algorithm is given here as well.

5.2.1 K-Means

k-means [7] is an algorithm that orders *n* instances of data in *k* groups. In this algorithm, we initially chose k random cluster centers (or centroids) in the feature space. We choose the distance measure to determine for each instance of data how close it is to the cluster center. Each group contains instances that are closest to the cluster center. When all instances have their cluster assigned, the algorithm calculates a new center from the average position in that group. The algorithm then repeats the process of assigning each instance to the cluster center. The process is iteratively repeated until a stable solution is found.

In water demand setting, where there are not so many features, the advantages of k-means algorithm are that it is simple and it can adapt to new data points. The possible drawbacks are that we have to choose the number of clusters manually, it is dependent on initial cluster centers and outliers can have a significant effect on the results.

5.2.2 DP-Means

DP-means [2, 4] is a similar method to K-Means. Except it does not use a predefined number of clusters. The algorithm creates a new cluster center when the data point is λ distance away from the nearest the cluster center. The advantage of that method is that the number of clusters is calculated from the data itself, but we still need to determine the λ parameter.

5.2.3 Incremental clustering algorithms

In the real world, concept drift can occur in the data. This means that with the arrival of new data from a data stream it is not reasonable to assume that the distribution of the data will be preserved through time. To adapt to this change, we either need to re-run the clustering algorithms regularly or that we need to stick to an algorithm, that can do this inherently. Of course, distribution change can imply various problems, as – for example – distribution and structure of clusters can change (with this the states of the system change and with this Markov chains are no longer valid in the new model).

There is a plethora of stream clustering algorithms, such as BIRCH, CluStream, ClusTree, D-Stream, DenStream, DGClus, ODAC, Scalable k-means, single-pass k-means, Stream, Stream LSearch, StreamKM++, SWClustering and others [8]. These algorithms rely on different data structures that take care about cluster statistics, which are further used in heuristic for cluster splitting or other operations. Incremental algorithms "see" a particular data point only once and cannot rely on past data, just brief summaries.

Incremental learning algorithms that produce stable results will be examined for a potential usage within the system.

5.3 Markov chain

Markov chain is the mathematical system that describes the transitions between states using certain probabilistic rules. The property of Markov chains is that it does not matter how we get into a certain state the possible future states are the same.

Markov chains are often represented in the graph form, where the nodes represent the state and edges represent the state transitions. The Markov chain can also be represented with the transition matrix **Q**, where an element q_{ij} represents the probability of state i going in the estate j, for all non-diagonal elements.

In our system explained below we use the continuous-time Markow chains. It adds the concept of time into the Markov chain. The idea is that the system spends a random amount of time in each state. To achieve a realistic effect, typically the times spend in each state, are exponentially distributed.

Markov chains could also be replaced with a machine learning algorithm, that could predict the next transition based on the historic data and feature vector representing the current data point.

5.4 Multi-level System's State Analysis Prototype

The system is attempting to solve the problem of visualization and interpretation of multivariate time series data. The goal of the model is to identify different states and identify transitions between those states.

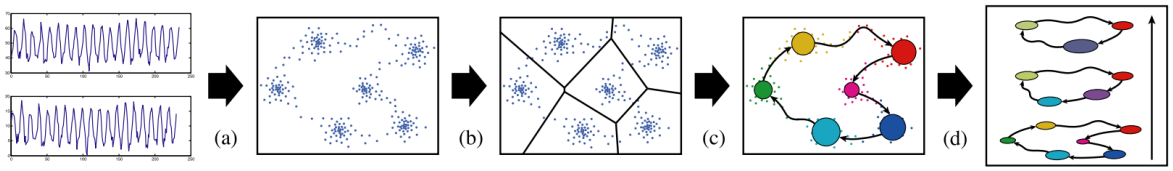

Figure 10: Schematic display of steps. (a) constructing a point cloud, (b) Identifying states by partitioning the ambient space, (c) Modeling the transitions between states and (d) Creating the hierarchy of states and transitions [1].

The methodology of the system consists of four steps (Figure 10):

- Constructing a point cloud to represent time series – This step constructs a cloud of data as points in multi-dimensional space and does not consider the time. This is the approach often used in machine learning when constructing the feature vectors.
- Identifying states by partitioning the ambient space – The model constructs the typical states of data. The states are constructed with clustering algorithms, using predefined metrics. Any metrics can be used, but the most often used metric is Euclidian distance. The current implementation of the system uses k-means and DP-means algorithms. These algorithms are used because they are computationally efficient and robust over the large range of inputs.
- Modeling of the transitions between states – In that step, we try to model the dynamic with transitions between states constructed in the previous step. Each state is represented as the state of the continuous-time Markov chain. The signal model is modeled as the trajectory between those states. The Markov chain is defined with transition rate matrix Q. Its non-diagonal elements define the transition rate from one state to another. Diagonal elements represent the rate of leaving the state. Parameters of the matrix are determined from data, with the use of a maximum likelihood estimator. The diagonal elements of the matrix are definite with the equation:

  $$Q_{ij} = N_{ij}/t_i$$

  The N_{ij} represents the number of transitions from i to j, while t_i represents the time spent in state i. The diagonal elements are the negative sum of non-diagonal elements in each row of the matrix.
- Creating the hierarchy of states and transitions – In the final step we are defining the hierarchy of states and transitions, therefore we can create the multiscale representation of states. That is achieved with the hierarchy of Markov chains. Each Markov chain is then assigned to the level of scale, that can be used in final visualization by the user. At that stage, we have to do two steps. First, we have to decide which states are merged and in the second we aggregate the states of the lowest scale Markov chains to obtain the higher scale.

The final results are visualized with the user interface (Figure 11). Different visualizations of results are possible in the system.

- The first is the graphical representation of the Markov chains at the chosen scale. The states are represented as circles and are connected with arrows that point to likely transitions of states. The user can change the scale at which he observes the results.
  The visualization of states uses five attributes:
  - Radius – It encodes the proportion of time that the observed system spends that state.

- Position – It reflects the position of the state in feature space. The distance between nodes encodes the distance between centroids in clustering. The position can be set manually to fit the user's understanding of data.
- Color – It represents the state's position in the hierarchy
- Label - The system also automatically marks the states with the label with the property that is most remarkable for that state and marks if that attribute is high or low. Labels can be changed to fit the user's interpretation of data.

  The system also creates a simple explanation of which features are important and at what time the observed system is most likely in that state.

- Border – It is used when you choose the state it is highlighted with the blue color. All the transitions from that state are also highlighted.

The likelihood of transitions is visualized as the thickness of arrows and also with the number above it.

- The states are also visualized with histograms that compare the distribution of the state data with all data. If we choose the state by clicking the node on graphical representation, we get an overview of the selected state. The histograms show us the distribution of attributes and compare them to the whole data set. Each histogram also tells us the mean value of data in the chosen state.
- Visualization is also made for an overview of states over time.
- We can visualize each state with an explanation tree. That is decision three that explains the attributes for chosen state.

Figure 11: User interface and graphical visualization of Braila data.

This system allows us to visualize time series data in a way that at least domain experts should be able to interpret it. Even for cases, where data exibit a complex behavior, visualisations are comprehensive enough, that should give domain expert a better understanding of system dynamics.

6 Experiments

In the experiments, we have tested the Multi-level System's State Analysis Prototype on real data. Data used in experiments were supplied from Braila, for district Brailita.

Our experiments have been focused on identifying typical states for pressure data and water flow. In both cases, we have been changing granularity and data structure – all with a goal of filtering out as many typical states as we could identify.

6.1 Identifying raw data states and its derivatives

For the first experiment, we have explored raw data from Braila, district of Brailita. Raw data consists of: timestamp, pressure, accumulated flow, and accumulated flow in the opposite direction. The derivatives include data derivatives for all non-timestamp data. Derivatives come in very handy as, for example, the accumulated flow is, through derivative, transformed into a daily cycle flow (see Figure 5, orange chart).

6.1.1 Results

The very first results of our analysis are shown in Figure 12. When intepreting results, we need to keep in mind, that every state on Figure 12 has been given its proper color and that the results are a visual analysis of data represented on Figure 5.

Figure 12: The overview of all 5 states over time. The colors represent the different states. The x-axis represents time and the y-axis the level of representation. The model has five states of representations. (a) This shows the whole time range. (b) Zoomed figure to see details.

We can notice on Figure 12 (a) 5 stripes, where every stripe is associated with a number at the beginning. The stripe at the top begins with number 100%, second with 13%, third 6%, etc. This numbers represent so called "granularity", or, how much of complexity are we choosing for system's states. For stripe 100%, we are interested in the "big picture", or, in the most important state transitions. On the other hands, when observing a astripe 0%, we usually observe a very complex states overview.

Now, when we have a basic understaning of how to intepret Figure 12, let's first describe it:

- The top stripe, a stripe 100%, has identified two states: a »red« state at the very beginning of the stripe, a »grey« state, for the rest of the stripe, with the exception of little white state between April and May.

The »red« state at the beginning (of every stripe) is due to a faulty data. Despite that's not visible on Figure 5, the initial Brailita data were probably faulty, as the sensors have been very probably in the process of integration. The »grey« state covers all of other states. We'll skip the »white« state for the moment.

- The second stripe, prepended with number 13%, is extending first's stripe insights. We can notice, that »grey« area is divided into two substates: »green« and »purple«. Next to that, we can notice, that the number of »white« states (which in fact are »orange« states, but due to very narrow area those states cover, only white borders are visible) appear within the »green« state.
  The division of »grey« state into »green«, »purple« and »white« (we'll realise a little later, that »white« states are actually »orange«) is obviously due to tot2 values (see Figure 5). However, the appearance of additional »orange« (on current chart »white«) stripes can't be clearly explained from given visualisation. Which is why we need to refer to another tool, helping us intepret less obvious states appearances.

Before getting to that, let's first take a look at Figure 12 (b). Figure 12 (b) is a zoom-in of the results shown on Figure 12 (a). For the sake of clarity, we have zoomed-in at the point, where »green« state turns into a »purple« state.

What we can notice, especially at the stripe 0%, is appearance of daily cycles, which are obviously visible on Figure 5 (see analog2, blue chart).

After the overview of data states analysis, let's introduce a completely new tool, that helps us get a better insight of states for a given granularity. The new tool is visualising transitions between states along with a given state's importance.

Insights for a stripe on Figure 12 (a) (and prepended by a number 100%) is in more detail explaind by Figure 13.

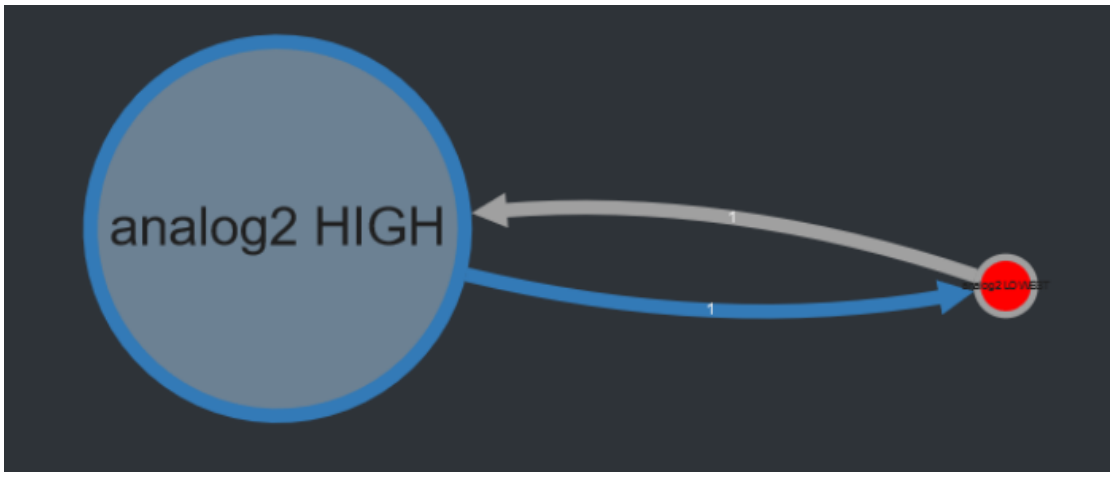

Figure 13: The graphs of states and transitions between them (first level). It shows two states transitioning from one to another.

On Figure 13 we can see two states: »grey« state or »analog2 HIGH« and a »red« state or »analog2 LOWER«. This is the most general picture we can get about Brailita raw data, where two main states prevail: the initial faulty data (low pressure data, for the period, when the system has been introduced) and the rest of the data, where sensors started to provide »normal« data. We see, that the »red« state, with a probability 1 converts into a »grey« states.

The first insight is very obvious. In order to increase complexity, we need to increase granularity. Which is why, we start interpreting a second stripe from Figure 12 (a), prepended with 13%. Insights into the stripe 13% are available on Figure 14.

Figure 14: The graphs of states and transitions between them (second level). Here we see four states and typical transformations.

In Figure 14 we can notice four states:

- »purple« state or »tot1 HIGH«, which consists of high values for water flow;
- »green« state or »tot1 LOW«, which consists of low values for water flow;
- »orange« state or »analog2 LOW«, which consists of low values for pressure;
- and a »red« state or »analog LOWER«, which consists of low values for pressure state.

The red state has been already identified as initial faulty pressure values. The newly appeared states are »purple«, »green« and »orange«. By comparing Figure 13 to Figure 14, we can note, that the »purple«, »green« and »orange« are derived from a »grey« state on Figure 13.

Let us first explain the »purple« state, which transits with (high) probability of 0.83% into the »green« state. The »purple« state encompases, as we have said, high flow values and transits into the state of low flow values (»green« state) – which are nothing but daily cycles visible in Figure 5 (see orange chart).

With the transition between »green« state to »orange« state (and viceversa) we, for the first time, observe transitions between states of two different variables (pressure and state). What we can see in on Figure 14, is, that with probability approximately 96%, a low flow transits into the low pressure and viceversa. Which means nothing but a high correlation: when flow is low, pressure is low as well (and vice versa).

After explaining 2nd level of granularity (the 13% stripe on Figure 12 (a)), we can now move to a 3rd level of granularity (the 6% stripe on Figure 12 (a)). The states on a 3rd granularity level are presented on Figure 15.

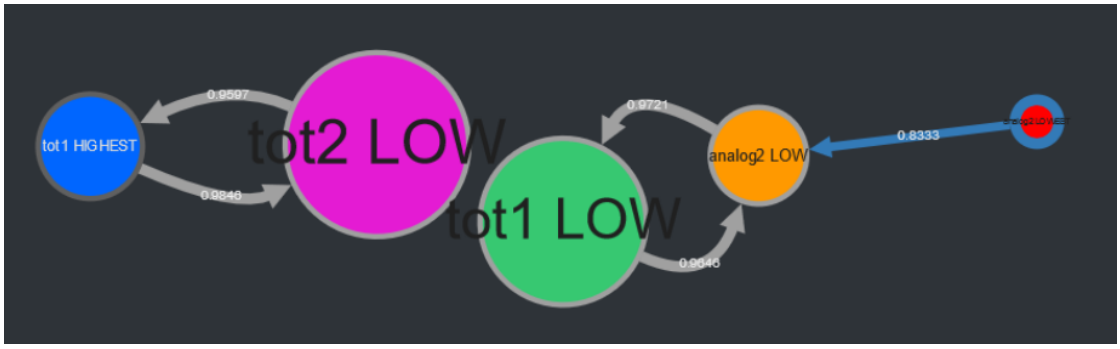

Figure 15: The states and transitions between them (third level). The figure shows five states in two groups. One group represents the daily cycle when water flow is high (right group), and the other when it is low (left roup.)

On Figure 15 we can observe a new phenomena – a split of states:

- the right states (»green«, »orange« and »red«) remain the same as on Figure 14;
- while the »purple« state from Figure 14 splits into two states: »pink« and »blue«.

The right set of states and their transitions have been already explained in 2nd level of states. The newly appeared states are the left set of states and their intepretation is: when water flow is high (»blue« state) the flow in opposite direction is low (»pink« state). Which is an obvious insight: if flow goes in one direction, it surely can't go into the opposite direction at the same time (see Figure 6).

Figure 16 shows the 4th level of granulation. In this figure, we can notice the appearance of a new, »yellow« state (or »tot1 LOW« state).

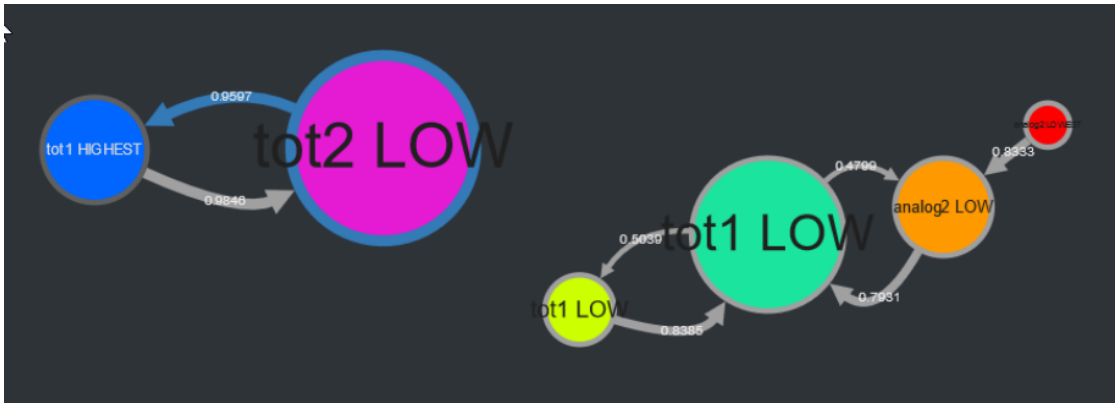

Figure 16: The graphs of states and transitions between them (third level). The figure shows six states in two groups. One group represents the daily cycle when water flow is high (right group), and the other when it is low (left roup.)

Obviously, we have identify the »low« water flow states - states, that appear during the night. What we can see is, that the »green« state transits into the »yellow« state with a 0.5 probability, while »yellow« state transits into the »green« state with 0.8 probability. The interpretation is: during the low water flow period (at night time), we have very low (»yellow«) and low flow (»green«) states. Very low almost always return to low state, while low state only occasionally transit into the very low state.

The fifth and the last level of granularity is shown on Figure 17. We can instantly notice, that the complexity of the states and their connections increases.

Again, as it has been the case already in the previous granularity level (see Figure 16), we can note:

- a separate left states cluster;

- and a right states cluster.

Let us first interpret the left states cluster. Comparing Figure 17 to Figure 16, we can notice the appearance of three new states: 1) a »pink« state or »tot1 AVERAGE« state 2) »purple« state or »tot1 HIGHER« state and 3) a »dark blue« state or »tot1 HIGHEST« state. Next to that, a »light blue« state »tot1 HIGHEST« is remaining from the previous granularity level.

The newly appeared states has been derived from the »pink« state »tot2 LOW« from the previous granularity level (see Figure 16). This is an interesting observation: only when tot2 (water flow in opposite direction) is low, these states appear.

The complexity of left set of states is increased and we are able to identify three possible cycles:

- pink - purple - dark blue – pink
- pink – light-blue – purple – pink
- pink - light blue – pink

Each of those three states represents a possible daily cycle when the tot2 (water flow in opposite direction) is low. The cycles appear at different times depending on the value of »tot1«.

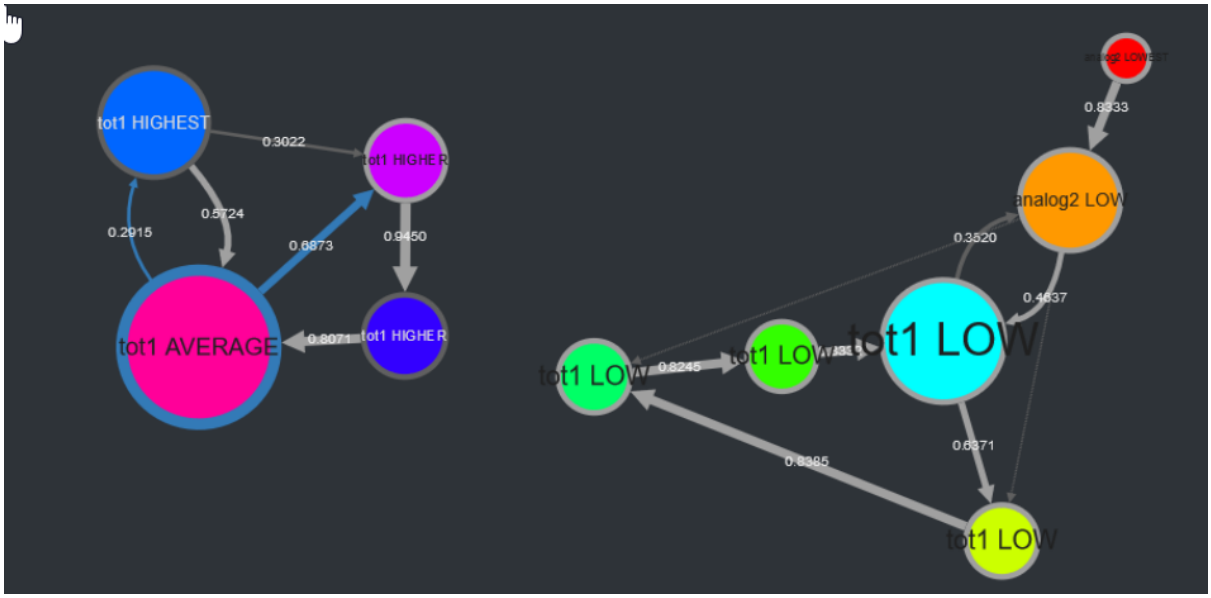

Figure 17: The graphs of states and transitions between them (third level). The states split into two groups. One group represents the daily cycle when water flow is high (right group), and the other when it is low (left group).

The same interpetation goes for a right set of states. At this point we'll skip a more detailed interpretation, however, an important observation can be done at this point. In Figure 17 we can notice a series of high water flow states (»tot1 high« in left set of clusters) as well as a series of low water flow states (tot1 low) in a right series of states.

6.2 Identifying pressure and its derivatives data states

In this experiment, we have narrowed raw data from previous experiments down to timestamp, pressure and pressure derivatives. The idea behind was to explore in a more detail pressure-related data.

6.2.1 Typical pressure states in time series

Let us start exploring pressure states on Figure 18. Figure 18 (a) is giving us an overview of the states over time, showing us most important states, where every color is associated to a given state.

We can notice on Figure 18 (a) states with a longer time span as well as a short-time span states appearing between long time span states. In order to get a better insight into short-time span states we need to zoom in and observe state representations on a shorter timeline interval (Figure 18 (b)).

For most of the time, we can notice the periodical behavior that we would identify as nighttime behaviour followed by a daytime behavior. However, next to that, we can also notice states (e.g. »red« states), that at a first glance, we are unable to explain. As it will turn out later, the »red« states are states with low pressure while the »green« states will be identified to states with unusuall high pressure.

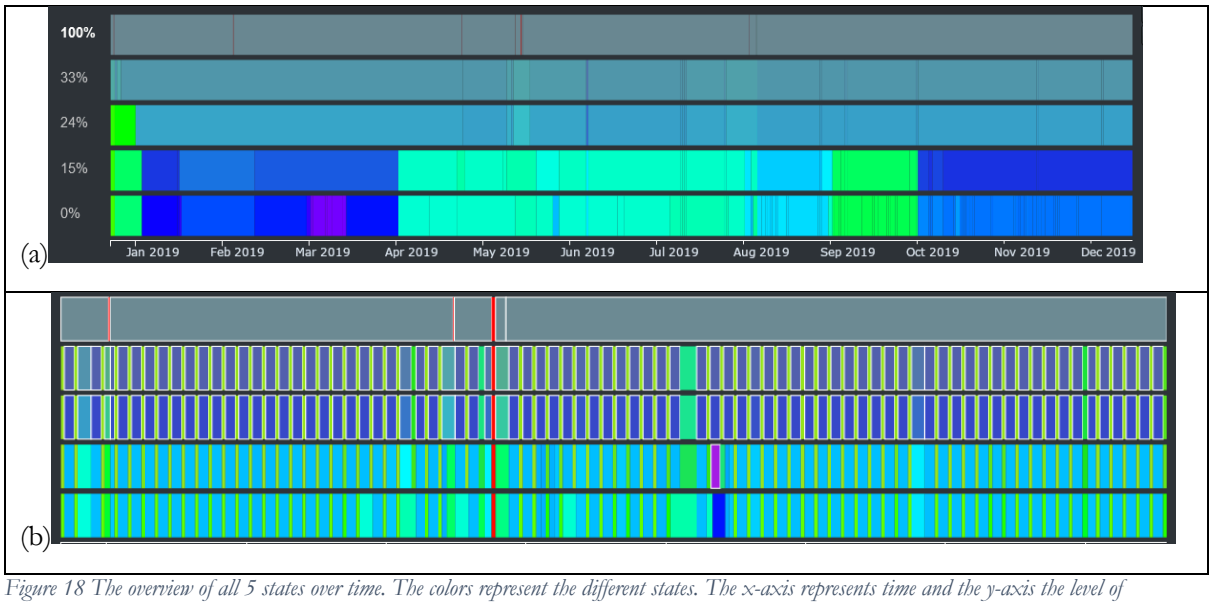

Figure 18 The overview of all 5 states over time. The colors represent the different states. The x-axis represents time and the y-axis the level of representation. The model has five states of representations. (a) This shows the whole time range. (b) Zoomed figure to see details.

Similarly as in the 6.1, we start our analysis with the top stripe from Figure 18 (a). The states are presented on Figure 19.

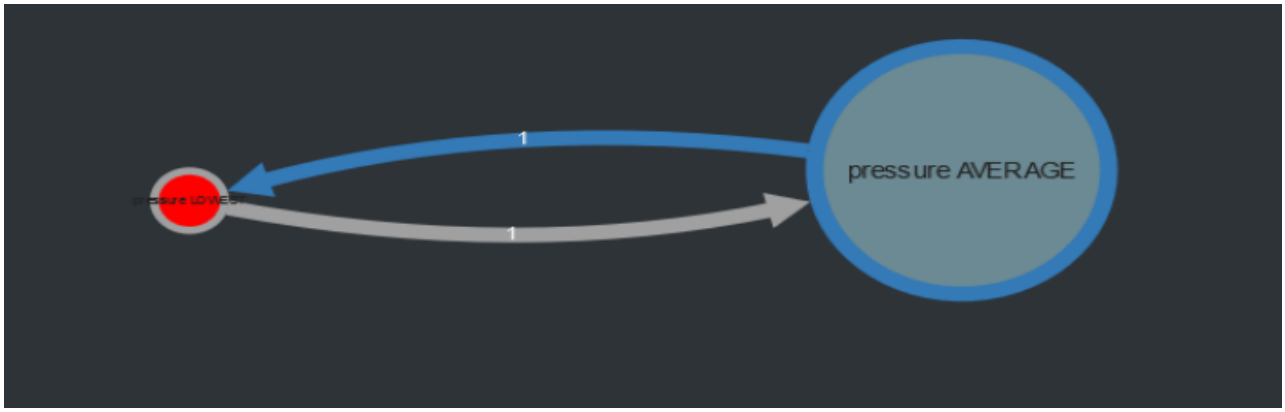

Figure 19: The graphs of states and transitions between them (first level). The figure represents two states.

We see, that on the first granularity level we have identified two states: the »red« state (or »pressure LOWER« state) and the »grey« state (or »pressure AVERAGE«). From the analysis it seems »grey« state covers all of the states except for the very low pressure states. At this point it may be possible, that, already on the highest granulation level, we have identified anomalous (»red«) pressure states. In order to confirm (or reject) our hypothesis, we need to explore states on lower granularity levels.

States on the next granularity level (states overview is provided through second stripe on Figure 18 (a), starting with a number of 33%) are presented on Figure 20.

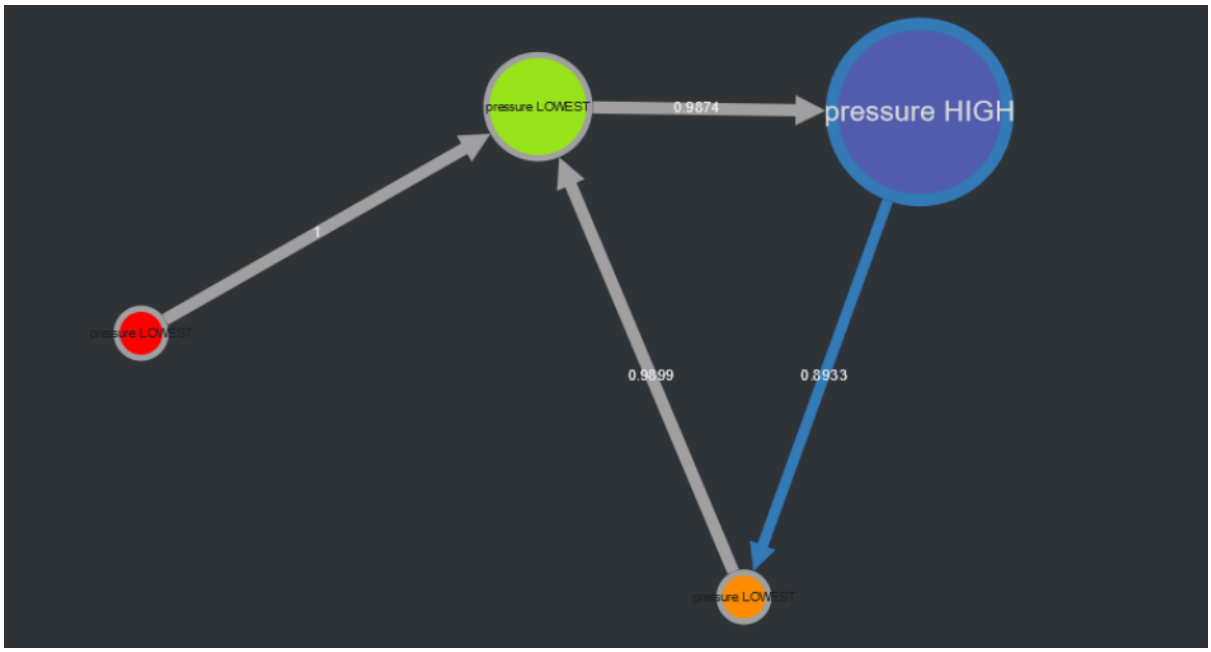

Figure 20 The graphs of states and transitions between them (second level). The figure represents four states. The green, orange and blue represent the typical daily pressure cycle.

We can notice, that the »red« state remains, while the »grey« state from Figure 19 decays into three new separate states. The cycle is intuitively explainable:

- The »blue« state (which is a »high pressure« state) describes a daytime pressure values. The system almost always transit from a »blue« state into an »orange« state.
- The »orange« state seems very similar ot a »green« state, both states describe low pressure values.

It seem like the states have identified daily cycles, where:

- Daytime pressure values are described by a »blue« state.
- Nightime values are desribed by all the other states (»red«, »green«, »orange«).

On the next level of granularity, represented by Figure 21, we can notice the appearance of a new state – »bright green« state.

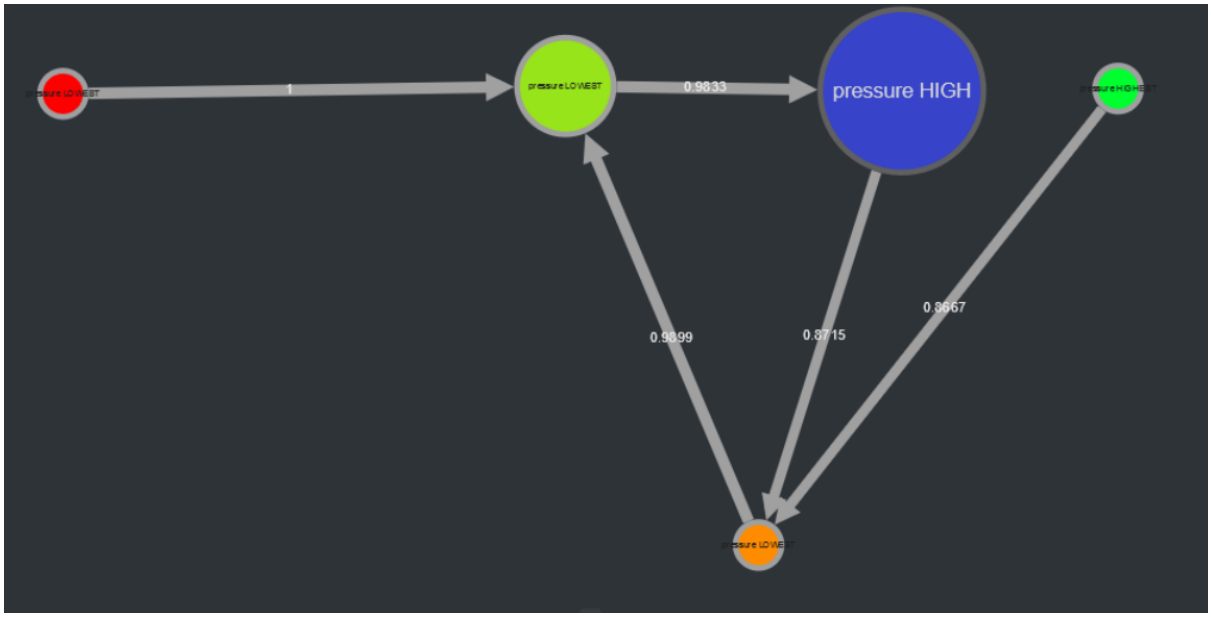

Figure 21 The graphs of states and transitions between them (third level). The figure represents five states. The green, orange and blue represent the typical daily pressure cycle.

One particular dynamics appears on Figure 21. As we have alredy noted on previous chart, the cycle of states composed of »blue«, »orange« and »green« represent daily cyles. However, the two »outsiders«, not part of the triangle cycle, are the »red« and »bright green« states. These two states represent very low and very high pressure. Those states typically do not occur in a daily cycle and always transit in the same way back in the normal dayly cycle states.

An increased granularity is presented on Figure 22. We can notice that new states appear within the daily cycle as well as new states, not part of the daily cycle, appears.

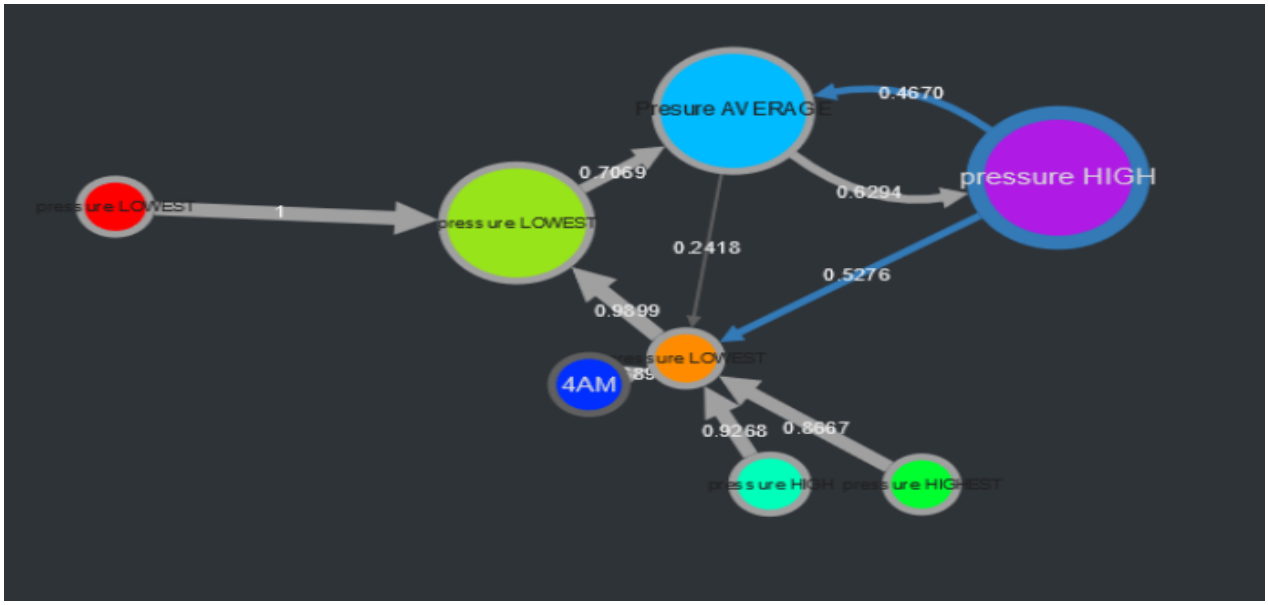

Figure 22 The graphs of states and transitions between them (fourth level). The figure represents pressure states. The connected states (green, light blue, purple and orange) represent the typical daily pressure cycles.

The complexity on the Figure 22 is high enough, that, rather than describing all relations and states, we rather focus on particular aspects of the representation.

First, states transiting to »orange« state seem particularyly interesting for two reasons:

- First, the »dark blue« state appearing at 4am and transiting into the »orange« state. Obviously, we have a characteristical enough behaviour at 4am, which is very probable related to watr utility pressure manipulation.
- Secondly, the two »green« states transiting into the »orange« state. The transitions are interesting as we are witnessing an oscilation in pressure, where pressure from a very hight values drop to very low values – without being a part of the daily cycle.

Second insight is a state exchange between »light blue« and »purple« state within a daily cycle. It seems like we have some kind of oscilation between »normal« pressure values and »high« pressure values.

On the fifth level of granularity, we can notice a very complex state transitions (Figure 23). The level-four granularity is expanded and at this point, the states analysis is limited to particular states, one would want to explore in more detail.

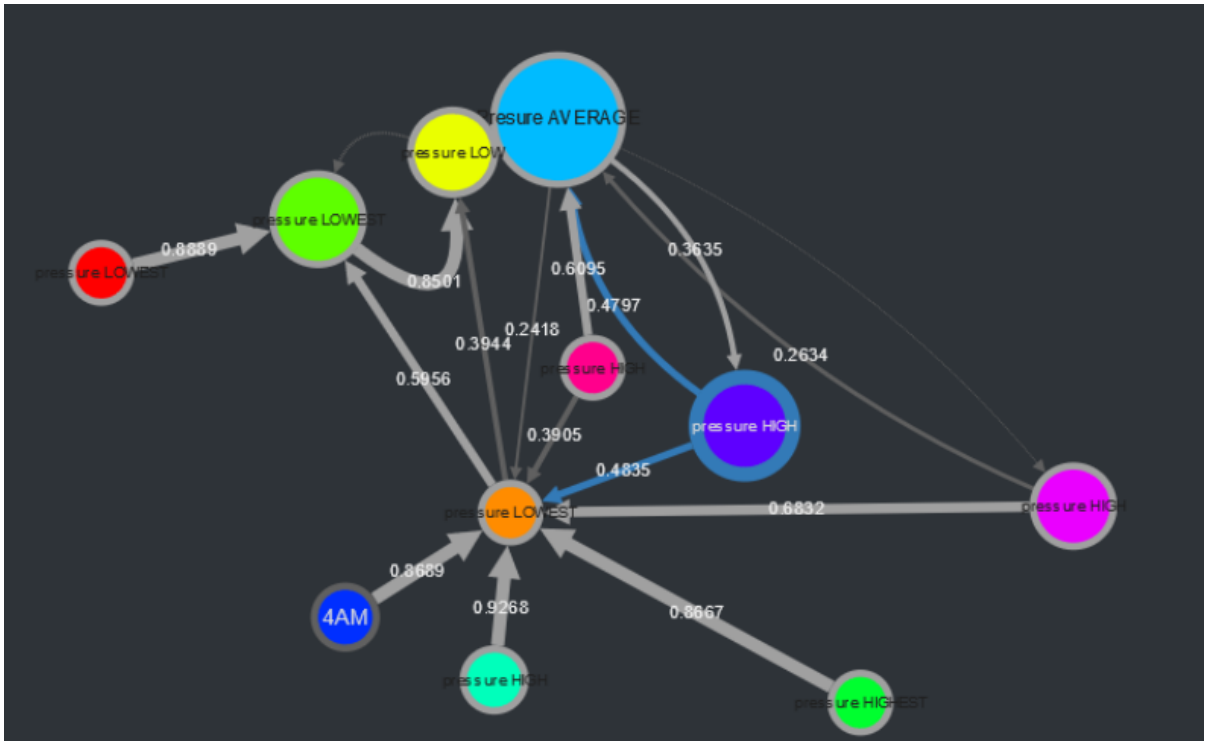

Figure 23 The graphs of states and transitions between them (fifth level).

6.3 Identifying enriched pressure and raw water flow data states

In this experiment, we used the enriched pressure (analog2) data with features explained in 5.1 and raw water flow data (tot1 and tot2). The additional features introduced in this experiment are: (a) pressure mean over the last n days (one, two, three days). By averaging pressure and water flow, we expect to get insight into a generalized behaviour of the Braila data.

6.3.1 Results

Figure 24 shows how in this experiment identified states change over time. On the figure, we see five stripes, each representing the level of granularity. Most of the states do not exibit a daily cycle, which is the immediate effect of newly added features to the dataset. The newly added features (the mean of pressure ower the days) average out daily cycles. As we'll note later, some kind of cycles can be detected only on the fifth granularity level.

We first notice (Figure 24 (a)) that we have three periods over a time span of a year: one at the beginning of the year (winter), one in the middle (summer), and one at the end of the year (winter). We have some states that are present for a small amount of time and we could treat them as anomalies.

Figure 24 (b) is the zoom-in of the results on Figure 24 (a). We zoomed on the point where the purple state change to orange. Again, event with the zoom-in we notice the absence of typical daily cycles. However, what we can identify are yearly cycles with more general lower resolution states.

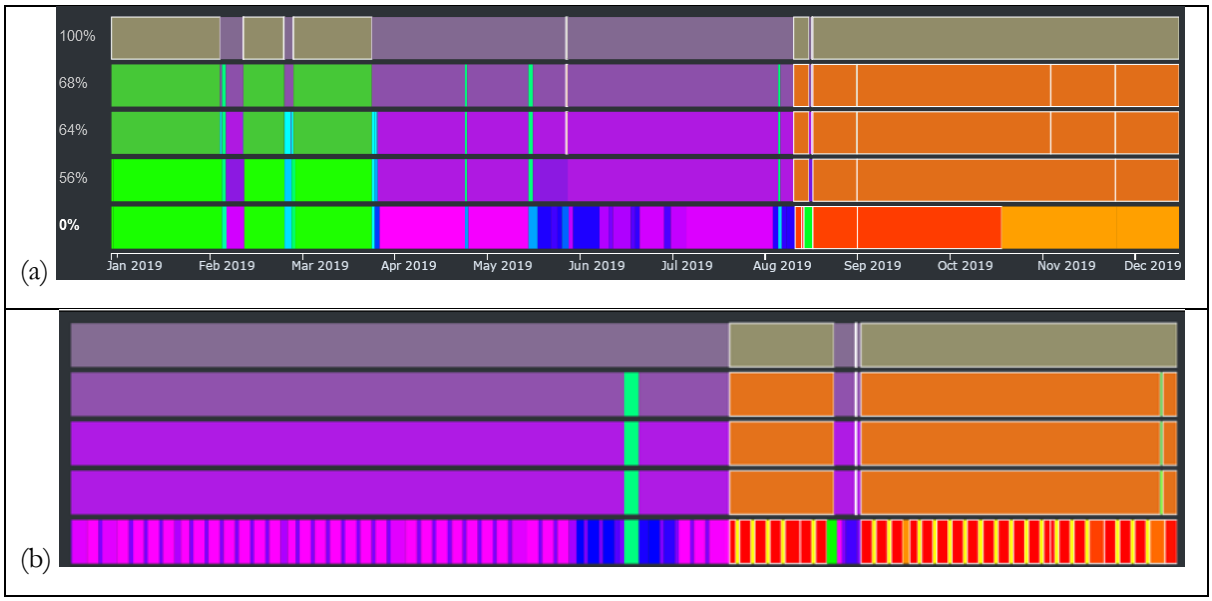

Figure 24: The overview of all 5 states over time. The colors represent the different states. The x-axis represents time and the y-axis the level of representation. The model has five states of representations. (a) This shows the whole time range. (b) Zoomed figure to see details.

Now that we understand the basic overview of states over time let's see how the states and transitions are represented with graphs.

Figure 25 shows the most basic identification of states, where the system was split into two states "pressure_1_day_mean LOW" (purple) and "pressure_2_day mean HIGH" (brown). The purple circle represents the states with low mean pressure, while the brown represents the high mean pressure. We see that both states convert one into the other. Observing those two states in Figure 25, we see that the low pressure is present over the summer and high pressure over the winter.

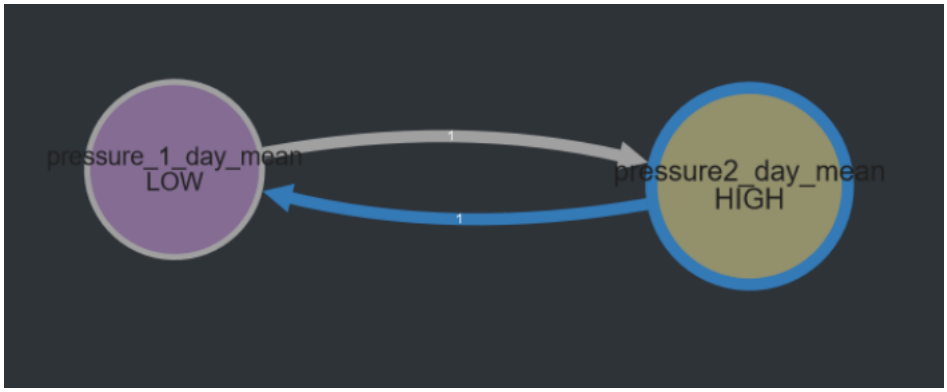

Figure 25: The graphs of states and transitions between them (first level).

By increasing granularity, new states appear (see Figure 26). The state with higher pressure splits into two separate states one with high water flow (tot1 HIGH – "orange") and one with low water flow (tot1 LOWEST – "green"). Both of those two states typically transit into the state with low pressure (purple) already present in Figure 23. The green state mostly returns to a state with low pressure, while the orange state can transit to both, »purple« and »green« state. These transitions represent the change of average pressure that happens over the year.

The »purple« state represents the lower pressure in pipes during summer. While »green« and »orange« states represent higher pressure over the wintertime. The only difference between the »green« and »orange« state is the value of tot1 (water flow). To identify this as a typical yearly cycle, we would need data, that would span over several years. Nevertheless, for year 2019, we can state, that we have identified a (yearly) cycle.

Another state in Figure 23 denotes lower pressure values for daily averages (light green). In Figure 24 (a) we can see that this state has occured only three times in a year 2019, thus this state could be identified as an anomaly.

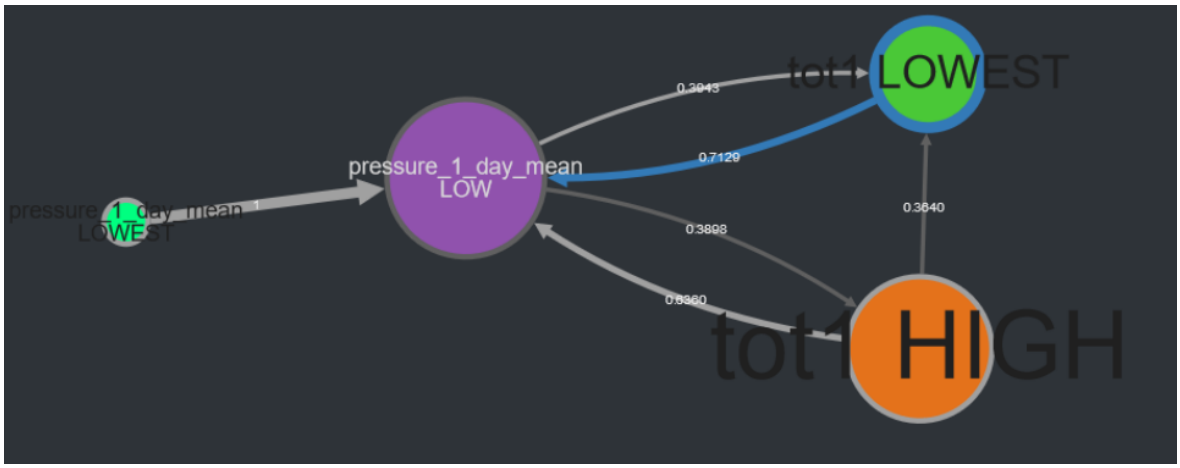

Figure 26: The graphs of states and transitions between them (second level). Figure show four states. Two typical daily cycles purple-green and purple orange.

On the third level of granularity (Figure 27), one new state appears between »purple« and »green« (light blue). This state is not often present over the year and represents low average daily pressure. It occurs when

green and purple states transition between each other or sometimes form a green state and then return to green. All other states stay the same as in second-level granularity.

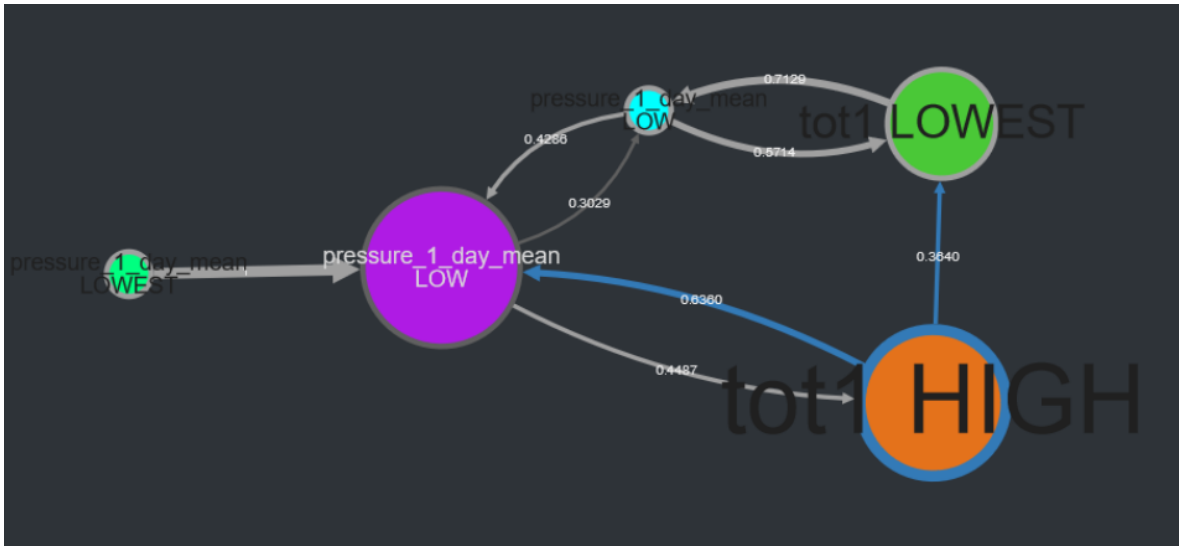

Figure 27: The graphs of states and transitions between them (third level). Figure show four states. Two typical daily cycles - purple-light blue-green and purple-orange.

The fourth granularity level (Figure 29) is almost the same as the third. The only difference is that the state "tot1_lowest" creates a two-state cycle. This new cycle could be explained as the daily cycle of high and low pressure.

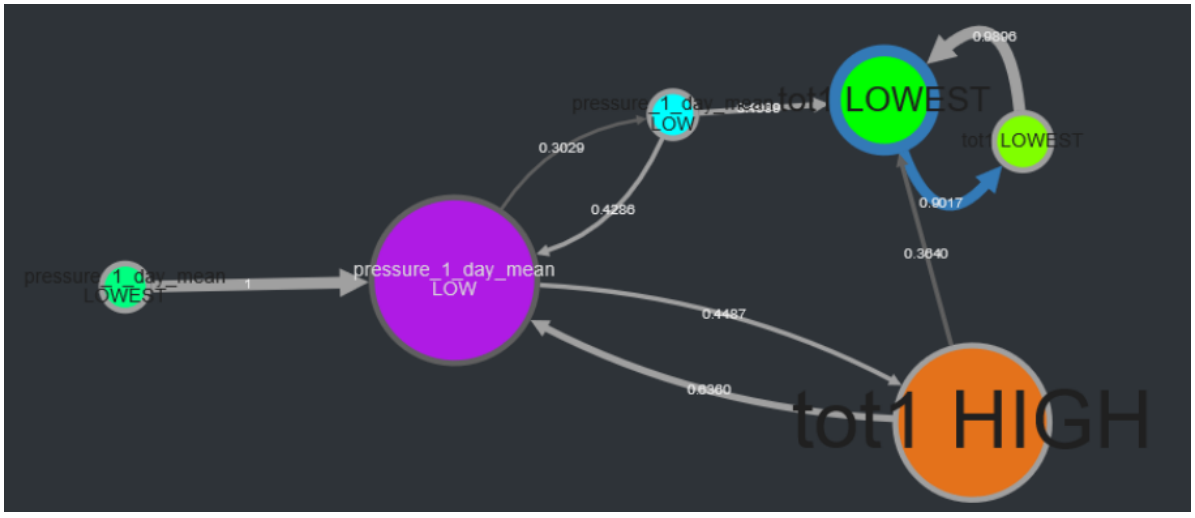

Figure 28: The graphs of states and transitions between them (fourth level).

On the fifth level of granularity (Figure 29) a lot of new states are introduced. First, we can see that basic structure stays the same - change of low and high pressure and split between high and low tot1. But a lot of new cycles appear. These cycles introduce a daily pressure cycle at all states from lower-level granularities. We can see that one group of three states (high pressure, high tot1) gets separated from the rest of the states.

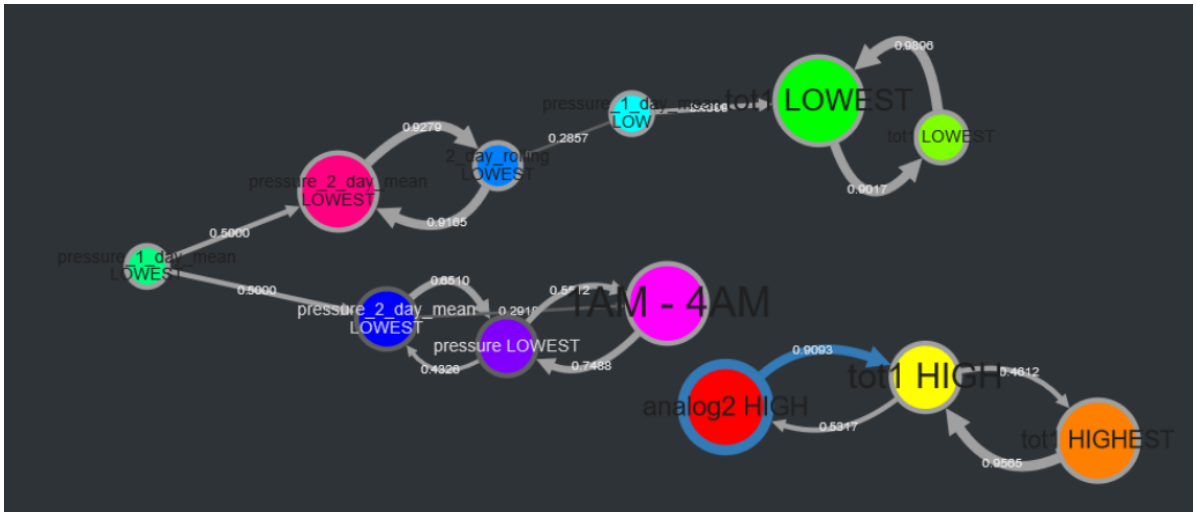

Figure 29 The graphs of states and transitions between them (fifth level). One part with high water flow forms a new group of transitions.

6.4 Identifying enriched pressure and enriched water flow data states

In this experiment we have modelled raw (Brailita) data enriching all of the raw values (water flow and pressure) with shifted features and mean values for various time windows from 1 to 7 days.

To avoid highly correlated parameters, we have included into the dataset only means for time windows of 1 day, 3 days, and 7 days. In such a way, we have reduced the number of parameters without really scarifying potential insights.

6.4.1 Results

The overview of states over time is shown in Figure 30. Again, Figure 30 shows us at what time observed system is in which state - for all levels of representation. We can see the daily cycles, where the observed system passes from one state to another. On all five levels, we can see a significant change of state. The timestamp of the drop corresponds to the time the states change, thus we assume that this is due to a drop in the water flow (tot1) in Figure 5.

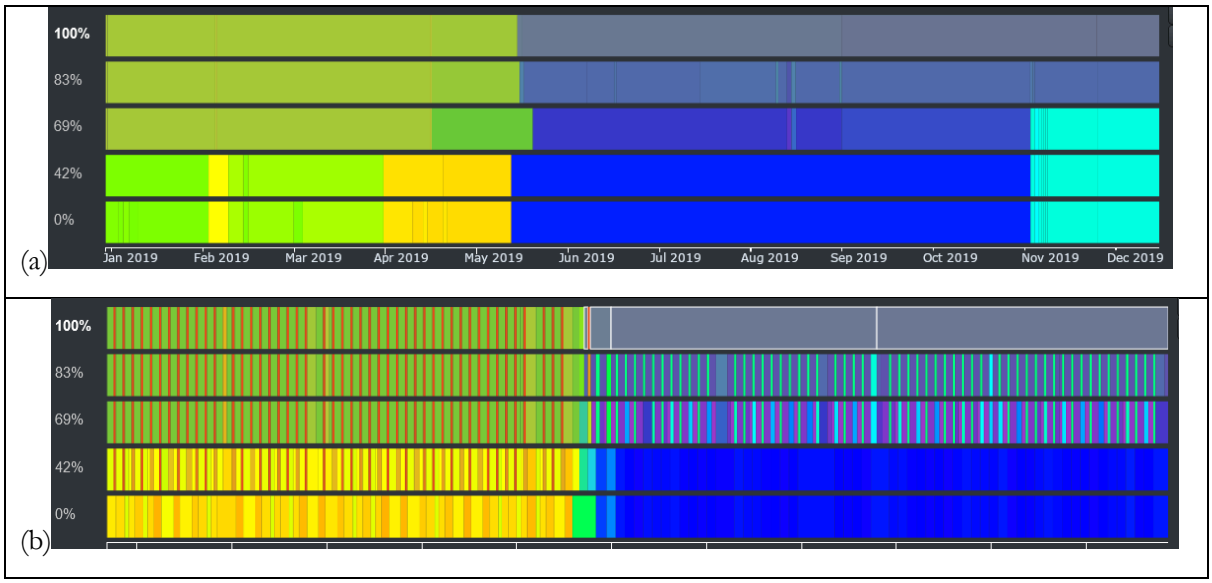

Figure 30: The overview of all 5 states over time. The colors represent the different states. The x-axis represents time and the y-axis the level of representation. The model has five states of representations. (a) This shows the whole time range. (b) Zoomed figure to see details.

The visual representations in Figures 31, 35, 36, 37 and 38 represent different states and transitions between them. Each of those figures represents different levels of representation, thus we have in Figure 31 only three states and each consecutive image more states.

We can see that in Figure 31 we have three states. The observed system is most of the time in the state with high mean water flow (tot1) ("blue" state). The water flow in that state is higher than the average water flow of the observed measurements. That state usually changes in the state with low water flow ("green" state). That state then periodically changes with the state with very low pressure (analog2) ("red" state).

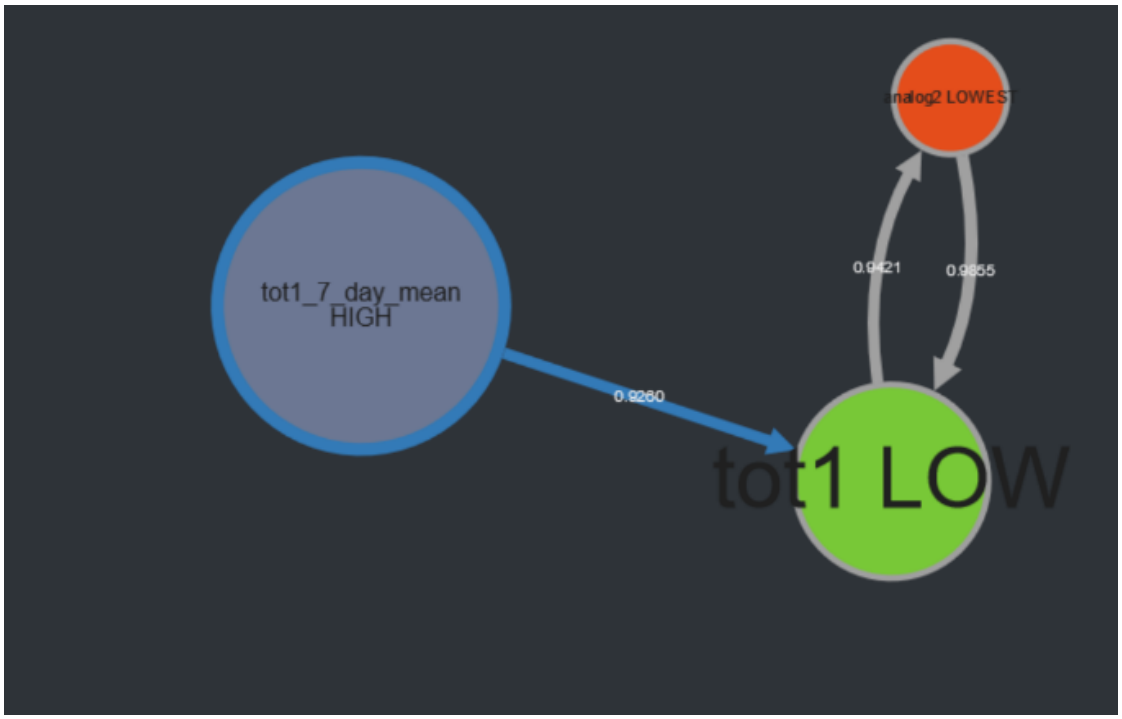

Figure 31: The figure shows three states and transitions between them (first level).

At this point, let us introduce some additional tools, that help us interpreting the data. Figure 32 shows the distributions of raw data for all states in first level representations. Histograms show the comparison of data that belong to that state compared to all data of the observed system. From histograms, we can observe that the most obvious difference between data in each state is the water flow (tot1). It is either higher (a) than a certain threshold or lower (b and c). The states with lower tot2 are further separated by pressure (analog2). Despite Figure 32 shows histograms for raw data only, we are able to access histograms for other features as well.

The histograms interpretation is generated automatically. An example of such auto-generated interpretation is given in Figure 33.

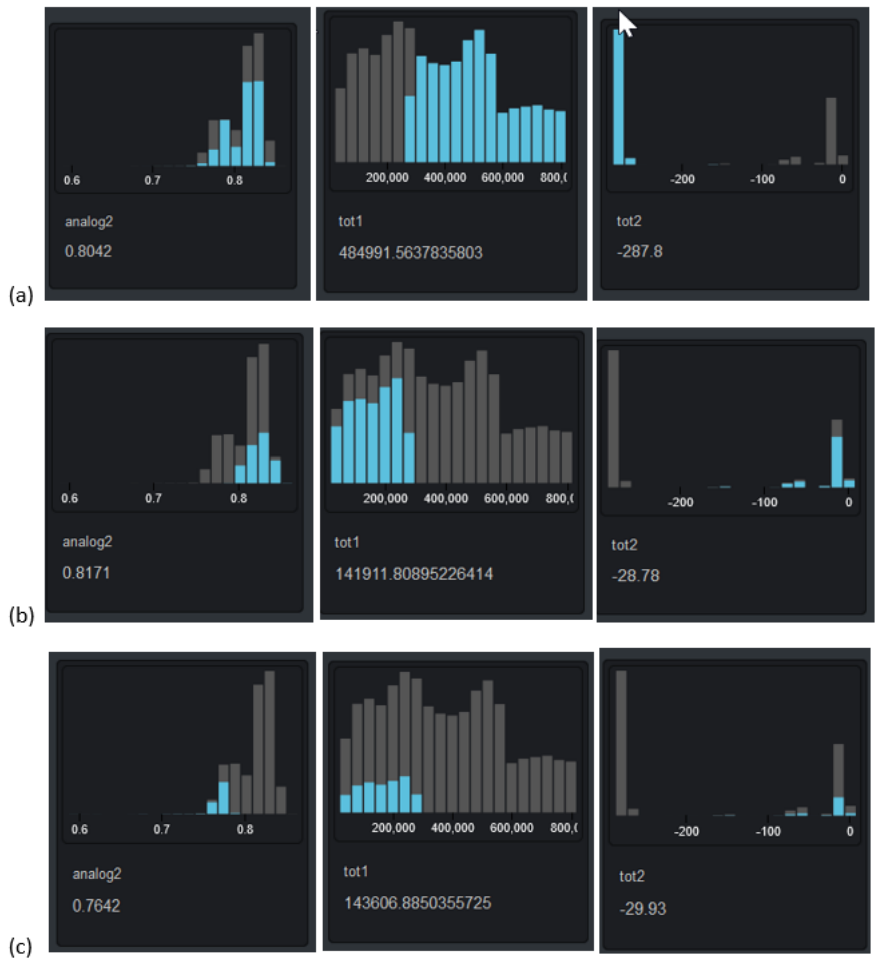

Figure 32: Histograms for first level representation. We represent the histograms only for raw data in states: (a) 'tot1_7_day mean HIGH', (b) 'tot1 LOW' and (c) 'analog2 LOWEST'

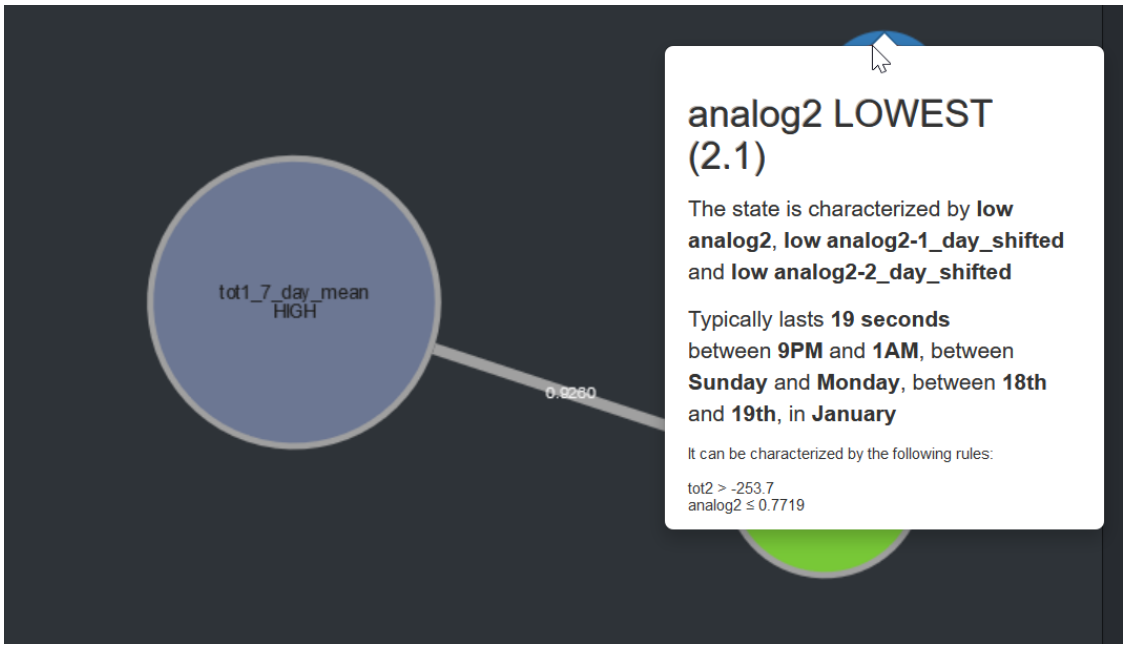

Figure 33: Automatic interpretation of a given state.

A second tool, helping us understand system states better are explanation trees and can tell us the structure of data for a particular state. Explanation trees for the first level of granularity are shown in Figure 34. Every split in the tree has the condition that decides which branch of the tree we follow. Diagrams at each node of the tree, represent the percentage of data belonging into a given state (green) or not belonging to a given state (red). The first tree in Figure 34 represents the node named 'tot1_7_day mean HIGH'. The tree has only one node and that node splits the data based on feature tot1. The data from the state that the tree represents are those for which 259030 < tot1. The second tree represents the state 'analog2 LOWEST'. The interpretation of that tree is if analog2 < 0.772 and -254 < tot2 the data instance is in that state. The last tree represents the third node 'tot1 LOW'. For data in that state apply the rule 259281 < tot1 and analog2 < 0.785. In the higher-level representations, trees have more branches and are therefore harder to explain in words.

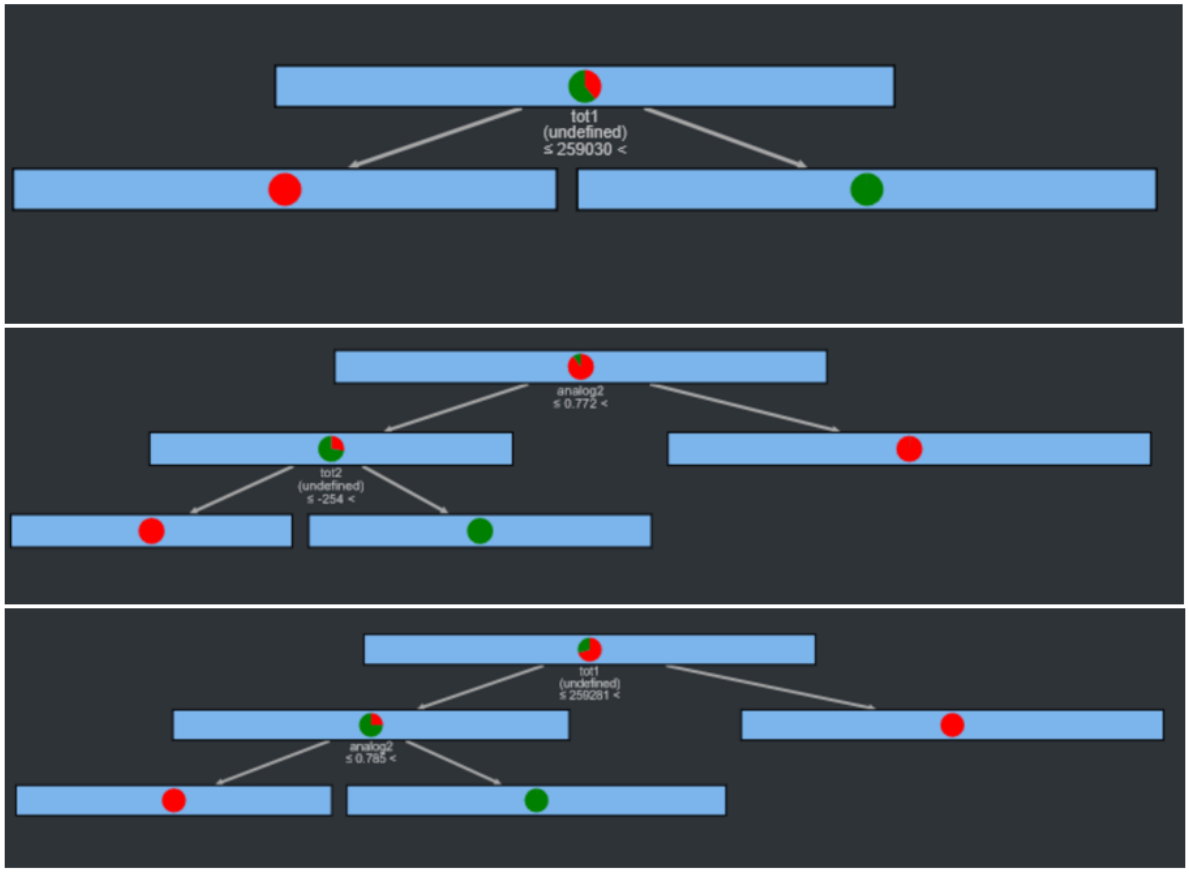

Figure 34: Explanation trees for states in first level representations.

In higher granularisation of the states (Figures 35, 36, 37 and 38), one of the more noticeable properties of higher-level representations is the split of states into two separate clusters. Our interpretation is that it happens due to the jump in the water flow data (tot1). This is a distinct change that we can see in Figure 3. In the representation of states, we can also notice the pattern, where low-pressure states will transform into the states with low water flow.

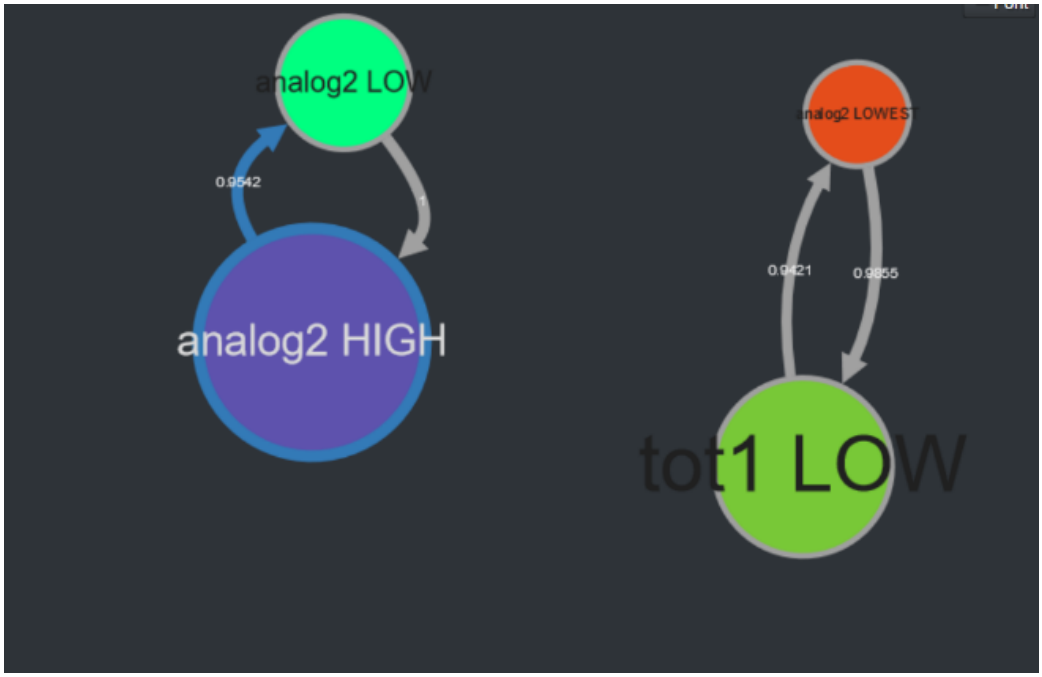

Figure 35: The graphs of states and transitions between them (second level). The figure shows four states that separate into two groups. One group represents the daily cycle when water flow is high (left group), and the other when it is low (right roup.)

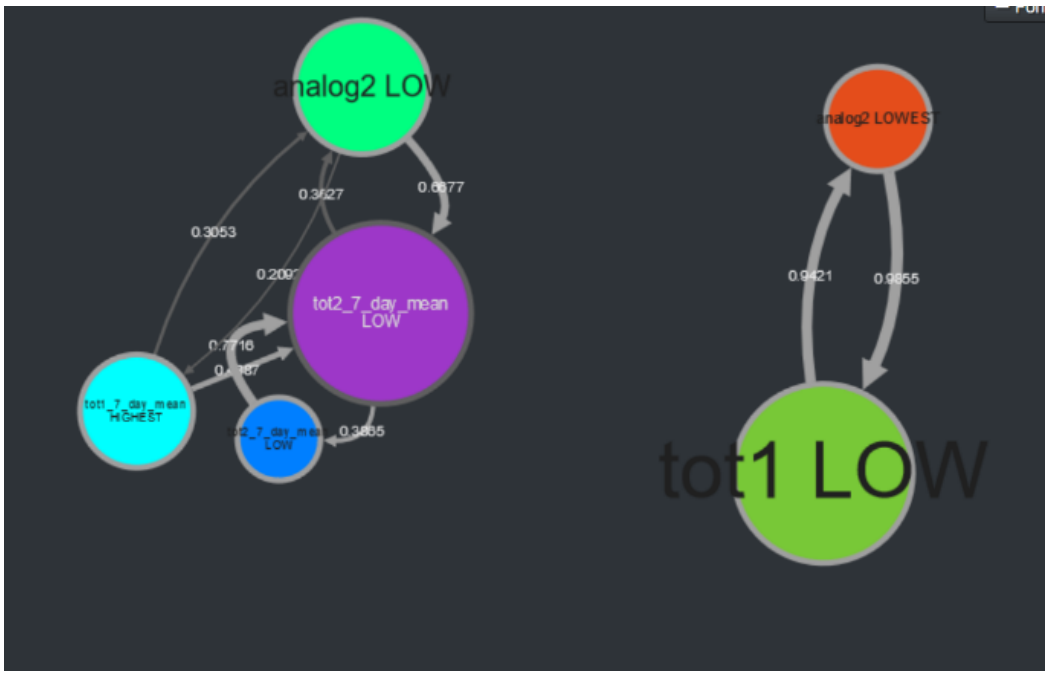

Figure 36: The graphs of states and transitions between them (third level representation). The states are separated into two groups. One group represents the daily cycle when water flow is high (left group), and the other when it is low (right roup.)

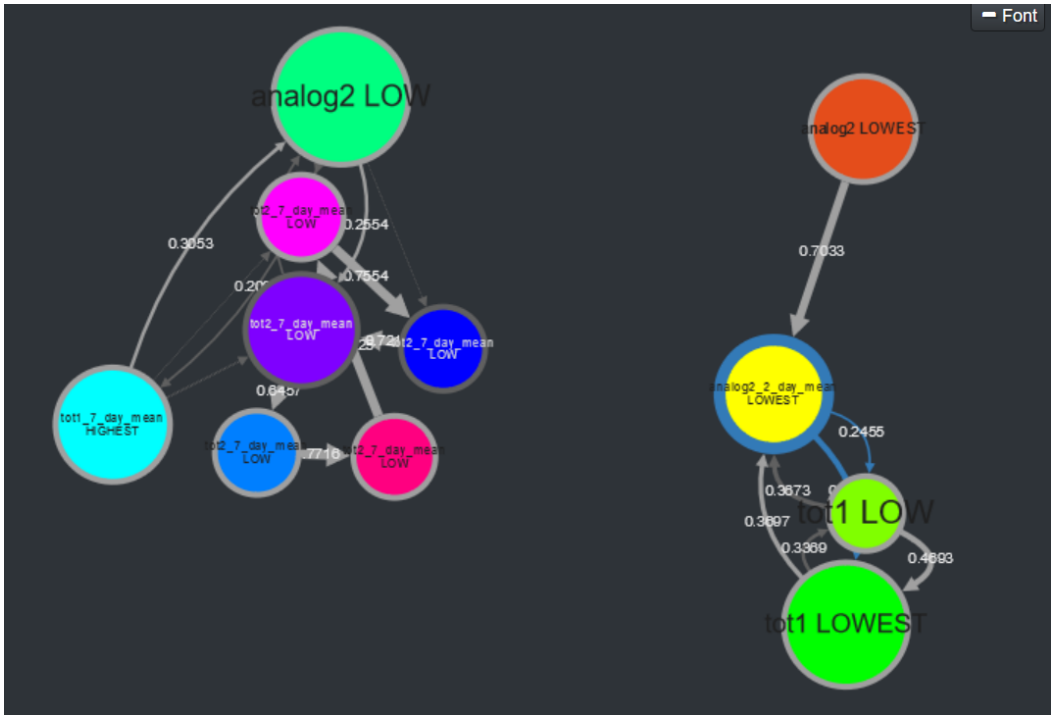

Figure 37: The graphs of states and transitions between them (fourth level representation). The states are separated into two groups. One group represents the daily cycle when water flow is high (left group), and the other when it is low (right roup.)

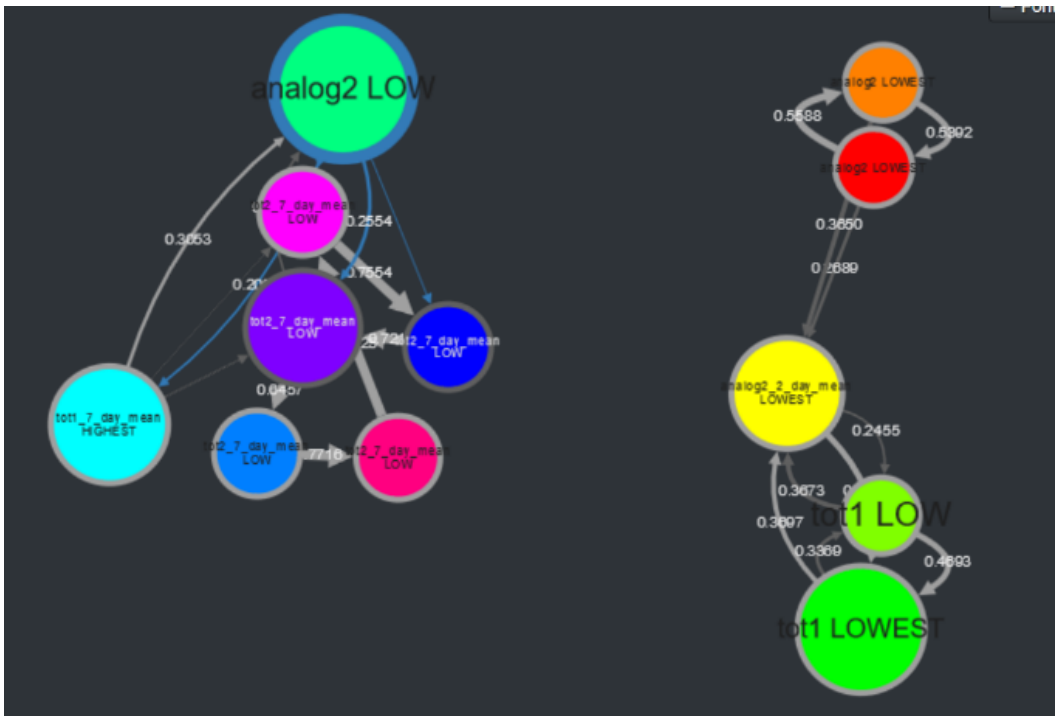

Figure 38: The graphs of states and transitions between them (fifth level representation). The states are separated into two groups. One group represents the daily cycle when water flow is high (left group), and the other when it is low (right roup.)

7 Conclusions and Future Work

It is important that we understand the data that represents our observed system. For a human, it becomes increasingly hard to analyse the state of the system with the number of sensors and the number of observed attributes. Multiple time series graphs are not enough, therefore we proposed the usage of a system that can break up this data into a human-readable form. We translated the multiple time series representation into a dynamic state model.

In this report, we presented the system that can help us define, identify, and interpret states of the system from the time-series data. One of the key features of the system is that it can tell us the possible future states and the likelihood of those states occurring. It is important because that can help us predict likely failures.

The system was tested on the Braila data set and we can see that it successfully creates visualizations that are not too complicated and therefore easy to interpret. We detected the states that we could not see from raw data alone.

The next step in the process of data monitoring is to identify the states that represent the failure and create a warning if the likelihood of that state emerging is high. In such a way, along with human intervention, we may be able to avoid the undesired state(s). To do that, the future work will focus on implementation of real-time system to determine the current state and trigger alarms if the system would be headed to an alarm state.

We believe that we can also optimize the parameter selection for the models. In this early work we were focused on the proof of concept. We showed that our proposed approach works, but the current results do not have optimized parameters. Some of them were chosen arbitrarily, but we believe that it is possible to optimize the model parameters to each use case separately. It will be an important task for the future to get as good results as possible.

Feature construction and selection could also be improved. We will add more aggregated features to the feature set and test different feature selection algorithms to get the optimal set of features. Also, some other features that affect the water system should be added to the model. The features that we plan to add are weather features, such as temperature, precipitation, humidity, pressure, and others.

8 Bibliography

[1] Stopar, Luka & Skraba, Primoz & Grobelnik, Marko & Mladenić, Dunja. (2018). StreamStory: Exploring Multivariate Time Series on Multiple Scales. IEEE Transactions on Visualization and Computer Graphics. PP. 1-1. 10.1109/TVCG.2018.2825424.

[2] Kulis, Brian & Jordan, Michael. (2011). Revisiting k-means: New Algorithms via Bayesian Nonparametrics. 1.

[3] Chen, Taolue & Han, Tingting & Katoen, Joost-Pieter & Mereacre, Alexandru. (2011). Model Checking of Continuous-Time Markov Chains Against Timed Automata Specifications. Logical Methods in Computer Science - LMCS. 7. 10.2168/LMCS-7(1:12)2011.

[4] Kodinariya, Trupti & Makwana, P.R.. (2013). Review on Determining of Cluster in K-means Clustering. International Journal of Advance Research in Computer Science and Management Studies. 1. 90-95.

[5] Kenda, Klemen et al. (2020). NAIADES D5.1 Failure and Leakage prediction – mid-term report, NAIDES project.

[6] Kenda, Klemen et al. (2020). NAIADES D5.5 Water demand prediction toolkit – mid-term report, NAIADES project.

[7] O. Maimon and L. Rokach, Data Mining and Knowledge Discovery Handbook. New York, NY, USA: Springer, 2005.

[8] Rodrigues, P. P., Gama, J., & Pedroso, J. (2008). Hierarchical clustering of time-series data streams. IEEE transactions on knowledge and data engineering, 20(5), 615-627.