NTNU
Kunnskap for en bedre verden

TDT4259 Anvendt data science

Group project

Water leakage detection in Verdal municipality.

Aneeq Ahsan - aneeqa@stud.ntnu.no - 546865
Emin Houseb - eminh@stud.ntnu.no - 481920
Jone Vassbø - jonev@stud.ntnu.no - 475335
Peder Ward - pederew@stud.ntnu.no - 479689
Torbjørn Wilson - torbjorw@ntnu.no - 479816

Autumn - 2021

Summary

In the last decade water distribution system has got more attention in the industry. The main cause for this is the lack of maintenance which results in massive water leakages in the network. The leakages are often difficult to find which further results in many labor hours, high leakage volume and a risk to contaminate the clean water. To overcome this issue many different methods have been researched in the last years. This report investigates the possibilities to use sensordata that already is obtainable from the water distribution systems as input data in anomaly detection algorithms. By using an autoencoder, which is an unsupervised neural network, this report shows that it is able to detect sudden and gradual leakage before the main leakage. This is done by training the model on normal flow measurement and testing the model on real historical leakages. This result shows that machine learning models with sensordata that already exist can be used as predictive maintenance in water distribution system.

Contents

List of Figures

List of Tables

1 Introduction and problem definition

1.1 Maintenance cost of water distribution systems

A water distribution system consists of a network with links such as pipes, pumps and valves that supply or distribute water to end-users (also called nodes). The main cost of maintenance comes from pipes [1]. It is a well-known fact that distribution infrastructures like these are deteriorating systems. The water industry is therefore facing a problem of managing these networks in an efficient manner to maintain service levels. Maintenance is done on these systems to prevent system failures, as well as restoration in case of an occurrence of failure to maintain predetermined service levels. Therefore, the main objective of maintenance is to improve and maintain system reliability and operation regularity [2].

Water leakage from pipes is nearly inevitable. However, it is certainly possible to reduce the leakage rate. Treated water supply, like in Norway, is a product formed after factors of production have been consumed on raw water. The reduction of leakage thus directly impacts the economic value in terms of chemicals, labor hours and energy of treatment processes [3]. According to research done by G. Venkatesh in 2020, total savings by a reduction of water leakage in terms of values in 2011 is estimated to be 9.6 MNOK [3]. According to a report published by The National Institute of Public Health in Norway (FHI) in 2017, the likelihood of contamination of drinking water in Norway is likely to increase [4], which further strengthens the need for a robust maintenance of water distribution systems.

1.2 Problem description

This project is a mandatory part of the course Anvendt Data Science (TDT4267). On that occasion, a municipality has provided us with data collected by sensors measuring the flow rate of water within their network of pipes. The municipality wishes to optimize cleanliness and up-time of their water distribution network by applying data scientific techniques on their data. The goal of this project will be to provide the municipality with tangible and applicable information to reduce cost due to downtime, water-waste, and polluted water, while maintaining their customers satisfaction.

More specifically, with the use of in-depth data analysis, we aim to solve the following challenges:

- Predicting leakage.
- Reducing maintenance cost and cost due to water waste.
- Providing clean drinking water.
- Reducing downtime of water supply to households.

1.3 Team description

The team working on this project consists of five NTNU students with both shared and unique areas of expertise. Midway through the semester one of the group members decided not to pursue TDT4259 any longer and withdrew from the project. Reducing the group size from six to five. Having members from several faculties and departments, such as Department of Computer Science, Engineering Cybernetics and Industrial economy & Technology management, makes the group robust and diverse. Furthermore, several members bring valuable and relevant experience from the industry, having worked with data science- and economy-related jobs.

Relevant competency/background/roles of team members can be seen in Table 1.

Members	**Background/roles**
Torbjørn Wilson	Bachelors degrees in electronics / Computer science method and video editing
Peder Ward	Bachelors degrees in automation / Machine learning
Jone Vassbø	Bachelors degrees in computer science / Model testing
Emin Houseb	Bachelors degree in mechanical engineering / Features extraction
Aneeq Ahsan	Bachelors degree in mechanical engineering / Features extraction and video

Table 1: Team members and background/roles.

Since not all members were experienced with this project's main topic, we kept the roles and task assignments open during the start-up phase. The initial phase was used to brainstorm ideas and to familiarize with the data. Eventually, the team delegated tasks among the team members based on their experience and desires. After delegating task assignments, the group was further split into three smaller groups.

2 Background

2.1 Objective and achievements

With the problem described in Chapter 1.2 the objective in this project can be defines as:

- Detect minor water leakage.
  - This will reduce some water waste.
  - This can prevent a bigger or a major leakage.
- Detect an intermediate water leakage.
  - This will reduce water waste.
  - This may reduce the risk of water getting polluted.
  - This can prevent a major leakage.
- Detecting a major leakage.
  - This will reduce water waste significantly.
  - The major leakage can affect the water supplies of households and companies.
  - This will reduce chances that the water leakage has an effect on the water reservoirs.

It is obvious that detecting any kind of leakage and then making data driven decisions on what to do about that information is the most beneficial. The problem is that small leakages can be hard to detect and can be mistaken for other water usages. What this project group aims to achieve is to use machine learning techniques on historical data to pinpoint where there is likely to be a leakage. This can later be used to predict where there may be a future leakage and then take preventive measures, both by reducing total water loss and reducing maintenance cost. Other positive effects by taking these measures is that it can reduce the chances of the drinking water being polluted while also making sure that the water supply is stable. By applying more automatic control bodies to detect leakage, the need for manual inspection and surveillance will be reduced.

2.2 Cross industry standard process for data mining: CRISP DM

CRISP-DM is a methodical structure for applying data science on big data. This is the method used in this project. The standard consists of six stages, see figure 1.

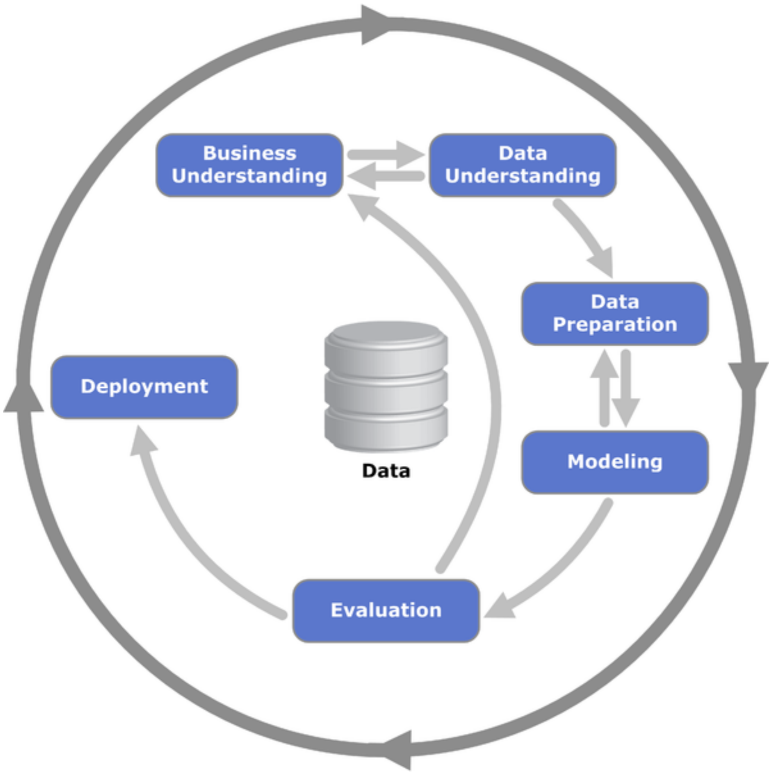


Figure 1: The six stages of CRISP-DM, from Wikimedia [5].

The **Business understanding** step is about getting to know the business, what their goals are and what they want to achieve.

Data understanding is the second step and includes getting familiarized with the data. What kind of data is it, what format is the data, what can it tell us, how precise it is and so on.

The thirds step, **Data preparation**, is how the data is organized and prepared. This includes removing uninteresting, empty, or corrupt data. In this third step the format of the data may be changed to be more optimal for the fourth step, **Modeling**. In the modeling step different types of modeling techniques should be considered and used. If a model needs a specific kind of dataset or format, the data should be prepared in the third step again. The modeling step is the step where the data is computed.

The results from the fourth step is then getting evaluated in the **Evaluation** step. In this fifth step it is important to go back to the first step to see if the business has reached its goals and achievement, and then choose the best model. If none of the models reach the objective, the process should start over in step one.

The sixth and final step, **Deployment**, is the step after the objectives are met in the evaluation step. This step should clarify how the modeling/-calculations should be implemented in the real system. In this step it is important that the business understands what this model can do and how it can be used. [6]

2.3 Design Thinking

Design Thinking is an iterative process that aims to give the designer or teams guidelines to understand the user, challenge assumptions, and reframe issues in order to find new methods and answers that are not immediately obvious, based on their current level of understanding. The phases in the design thinking process are straightforward at a high level: first, fully grasp the problem; second, investigate a wide range of viable solutions; third, iterate widely through prototyping and testing; and finally, implementation. For the purposes of this report these are divided into the six steps shown in figure 2, although there exist some variation of these based on the source. These six steps are discussed further in the following paragraphs. Although the steps are shown in a linear manner, this is a highly iterative process where every step yields insight, that shines light onto potential improvements, and necessitates reiterating the previous steps to improve results.

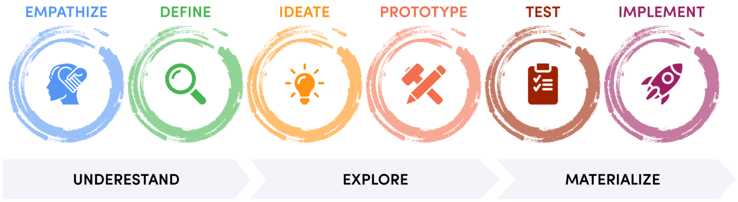


Figure 2: The six steps of design thinking [7].

Empathize: By the means of user research, it gains an understanding of (potential) users of the product. Design Thinking is a human-centered design process. Empathy is essential in this design process to understand the problem from the user's perspective and lets the designer's own understanding of the problem aside. This lets the developer or designer find a solution to the actual problem, instead of a solution to a problem the designer thought was the problem. It is therefore of utmost importance to be open minded so that confirmation bias does not affect the results. [8]

Define: Analyze the observations and synthesize them to define the core identified problems, called problem statements. It is common to create personas to help keep the efforts human-centered before proceeding to ideation.

Ideate: The next step is to generate ideas. With a strong foundation of information from the prior two phases, you can begin to "think outside the box", aim to seek different perspectives, and come up with creative solutions to the problem statements. In this stage brainstorming is common practice.

Prototype: This is the start of an experimental period. The goal is to find the best solution for each problem encountered. To test the concepts created, the team should create several low-cost, scaled-down copies of the product (or particular functionality present inside the product). This may be as simple as prototyping on paper.

Test: The prototypes are extensively tested by evaluators. Despite the fact that this is the end of the process, the process is iterative, so that the new insight from testing will assist in redoing the previous steps to make more iterations, changes, and improvements, or to discard some ideas.

Implementation: This step is not necessarily a part of the design process, and depending on the context and product, further iteration might not be necessary after implementation. Although it is in many cases, the design loop might go on almost indefinitely, with improvements and adaptations to changing environments.

2.4 Relevant examples

A significant body research is available in the subject of Leak detection in water distribution networks. A paper from El-Zahab & Zayed 2019 [9] identifies 941 distinct such papers written between 1980 and 2017 with a criteria of containing certain keywords in the title. The paper chooses 31 of these papers at random, and manually classifies them based on main technology. The result of this is seen in figure 3. Although the amount of data used to produce the figure is very low, it seems likely that using flow rate through the pipes, like proposed in this report, has not been a historically predominant strategy. This might be caused by several factors, such as concerns regarding location precision and sensitivity. The advances in anomaly detection in time-series analysis, can make this more viable, and might present a cheaper solution that requires less manual work and oversight than the other technologies.

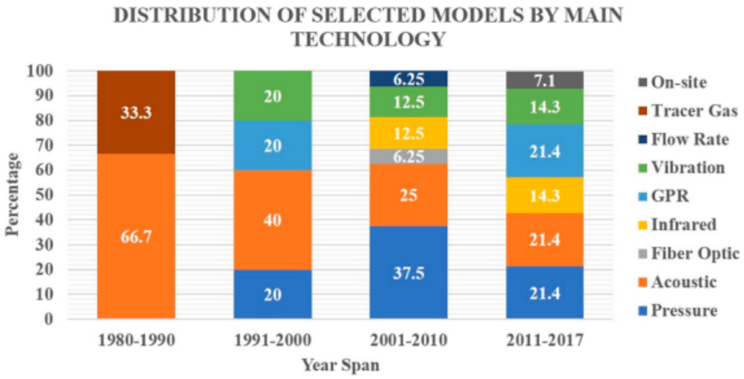


Figure 3: An overview of the frequencies of different methods of technologies for detecting water leakages [9].

3 Method

3.1 Data understanding - Description of dataset

The dataset contains 165 measurements, one data point each hour, for 6 years, in a distribution network of drinking water. A small subset of the network is shown in figure 4. The drinking water is located e.g., in a reservoir at a higher altitude than the houses it is supplying. As the water is flowing to the houses, at certain points, the flow is measured, in e.g. liters per second. The circle with a cross inside represents where these measurements can be located. There is no systematic order for the locations of these measurements.

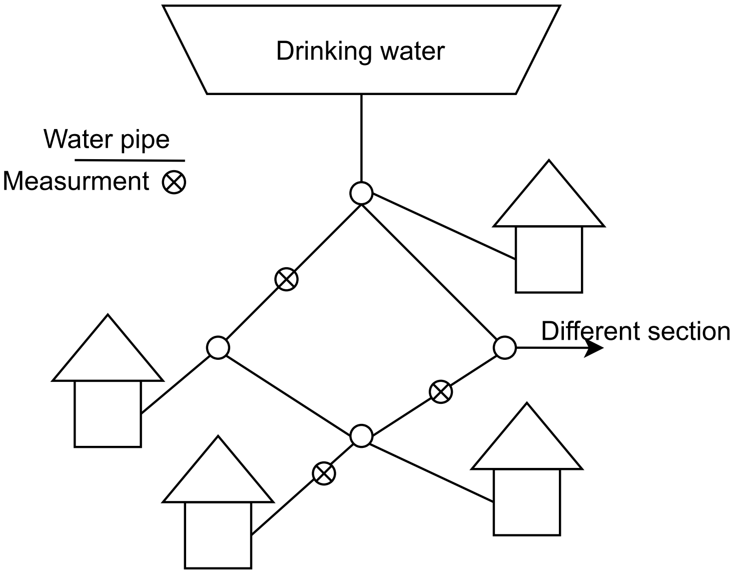


Figure 4: Description of dataset.

Figure 5 shows an example of the provided dataset. It is one week of data (149 points), for one of the 165 different measurements the dataset contains.

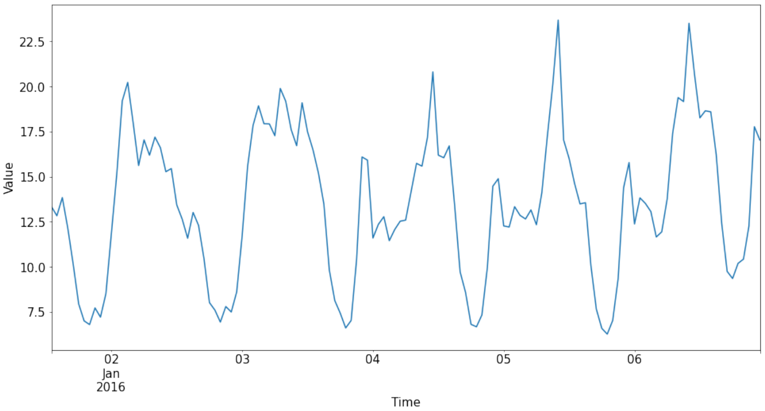


Figure 5: Example of 1 week of data from one measurement.

Table 2 shows example of 4 data points from one measurement. For each hour, there is one value, and there are values for 6 years. Adding another measurement will add another column to the table.

Timestamp	Value (l/s)
2016-01-01 01:00:00+00:00	13.133328
2016-01-01 02:00:00+00:00	15.717773
2016-01-01 03:00:00+00:00	16.877650
2016-01-01 04:00:00+00:00	16.984114

Table 2: Example of 4 points of data from one measurement.

As shown in the table above, the dataset contains hourly average values. This means that the sensors can, and will have, different sample times (number of measurements per hour), but these have been aggregated into one average value. This was done before the dataset was provided. The hourly averages are a sensible simplification, since it is reasonable to believe that it takes more than an hour to detect a leakage, and it is also sufficiently early enough. The simplification of the dataset helps running algorithms on it. Instead of having e.g. 240 data points in a day, there is only 24. This means doing calculations on the dataset will be much faster, and require less resources.

3.2 Tools

To analyze the dataset, Python and Jupyter Notebook have been used. Including in Python the libraries Pandas and Numpy are the main tools to analyze the dataset. In addition, matplotlib has been used to illustrate graphs and plots. To find abnormal patterns and possibly leakages in the dataset several methods were considered. Sklearn is an important library in

Python for this task. Sklearn contains several machine learning algorithms that can possibly be used to find anomalies in the dataset. Scikit-learn is a free machine learning library used for Python programming [10].

Because five students are working together on this project, a Jupyter server was made to collaborate. In this server everyone has the opportunity to create their own Python files to analyze the dataset, which is also located on the server. This makes it easier for the group to collaborate and one can easily import the dataset and see the work done by other group members. Figure 6 illustrates the work environment.

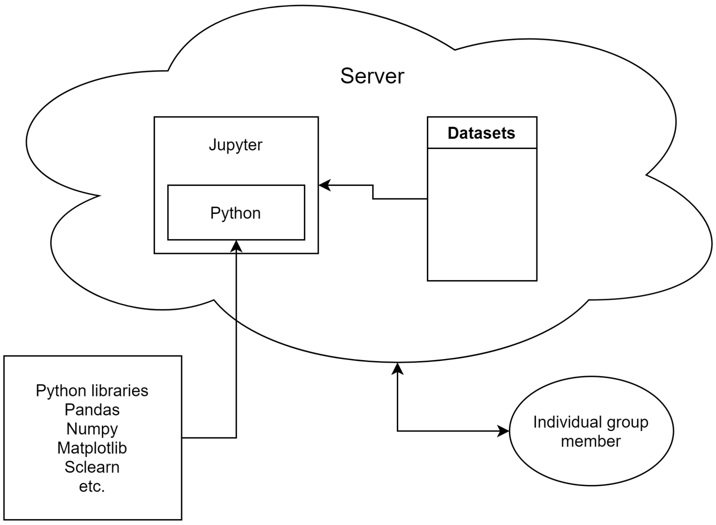


Figure 6: Topology of the work environment.

3.3 Data preparation - Prepossessing and cleaning

Some of the datasets are not complete. Figure 7 shows an example of a dataset with some signal loss. The reason for this could be error in the sensor, system error, maintenance work or lost connection. Also, some of the sensors only measure flow in one direction and are not bidirectional. Since water flows the easiest way through the network this can result in zero values if the water flows opposite direction of actual direction of the meter. For the sake of simplicity, complete datasets will be used in the analysis. For the purpose of cleaning the data and extracting the most valuable information, the sensors with a lot of zero values or not a number (NaN) values were excluded. Leaving sensors with complete or almost complete datasets for the analysis.

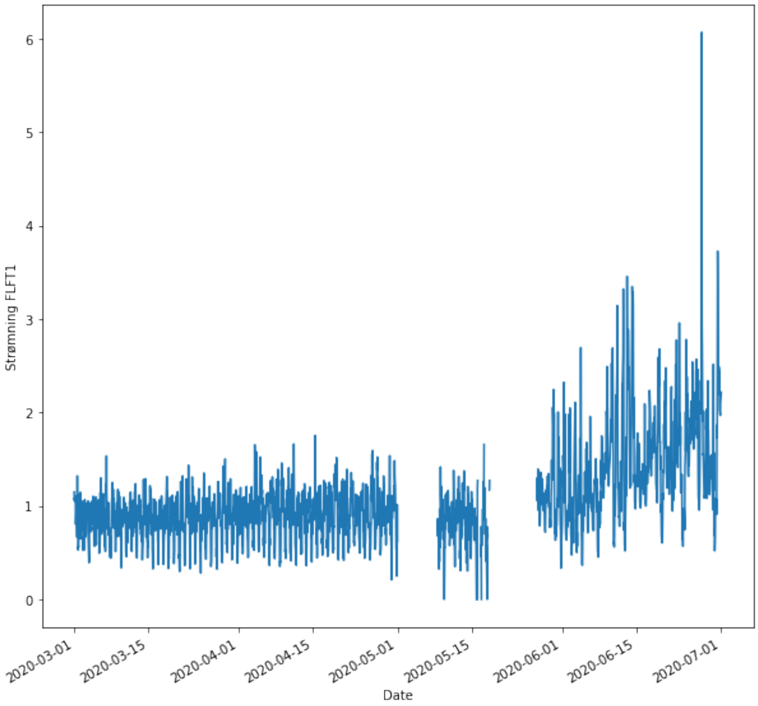


Figure 7: Example of an incomplete dataset.

Since the analysis will focus on single time-series data, in this case, the values from one sensor, and not multiple sensors combined, only three sensors have been chosen to be used. These sensors have been chosen since they are known to contain leakage on given dates. The three different leakages are represented in figure 8 , figure 9 and figure 10.

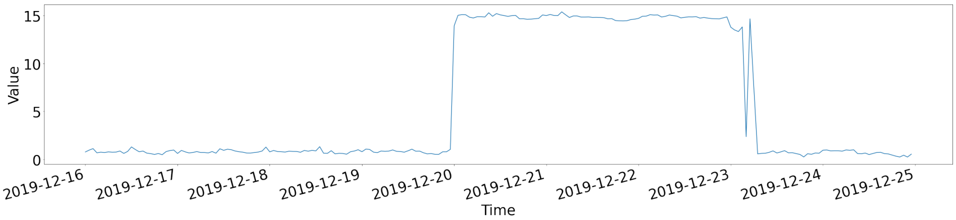


Figure 8: FLFT1 sudden leakage.

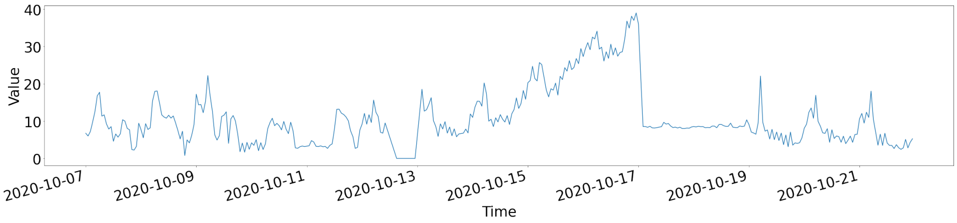


Figure 9: FT01 fast gradual leakage.

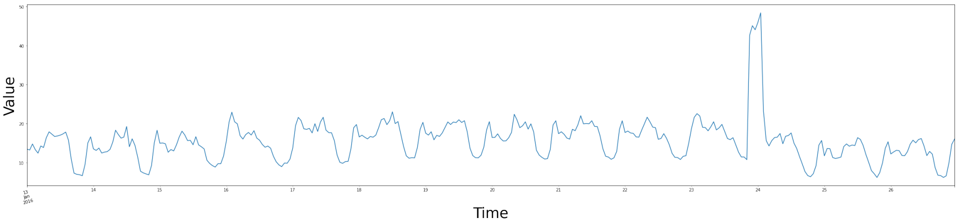


Figure 10: FT1_122 slow gradual and sudden leakage.

The reason sensors with known leakage are chosen is that the results are known, and it is then easy to compare machine learning results to the real world. The leakages used are different types of leakages where every one of them is interesting. The below list describes the different interesting leakages that the algorithm should detect:

- Slow gradual leakage.
- Fast gradual leakage.
- Sudden leakage.

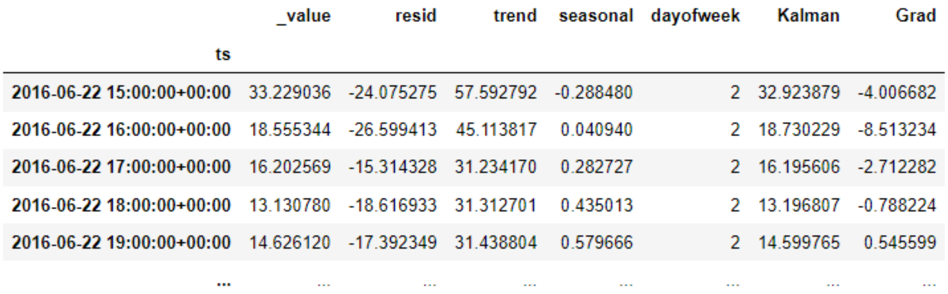
3.4 Feature extraction

Having only one-dimensional time series data, the group had to carry out feature engineering, in order to improve the performance of the model. The feature extraction started out broad by extracting basic features from the dataset itself and weather data from external sources. Some of the features that were extracted are day of week, holiday, minimum and maximum flow values, temperature, humidity, and rainfall etc.

Kalman and gradient features were extracted for all the features in the dataset individually. According to [11] Kalman filtering is an algorithm that provides estimates of some unknown variables given the measurements observed over time. Since the goal is to find the outliers and possible leakages in the system, Kalman filter provides a good measure of abnormalities in the sensor values. Gradients were found along the time steps and provide information such as change in sensor values, which has same effect as differencing a time series and thus makes the time series stationary. A stationary time series is easier to analyze by various algorithms and thus generalizes well to the dataset[12].

By inspecting figure 5, one can see that there are hourly variations and an increasing trend in the distribution network. Using seasonal decompose from the statsmodels library, which is a statistical Python module [13], the data was decomposed to find the features trend, seasonality and residual. Where trend is the average change in water consumption, seasonal is the recurring short-term cycle in the time series, resid is the time series after trend and seasonal components have been removed, and dayofweek is the enumeration representation indicating day of the week. After testing different combinations of features, it was noticed that using too many features increased the CPU usage and run time drastically, without improving the model noticeably. It was therefore decided to pick a handful of features which balanced the run time and model performance.

In figure 11, all the applied features can be seen. _value is the measured and flow, resid, seasonal, dayofweek, Kalman, and Grad are as explained above.



	_value	resid	trend	seasonal	dayofweek	Kalman	Grad
ts							
2016-06-22 15:00:00+00:00	33.229036	-24.075275	57.592792	-0.288480	2	32.923879	-4.006682
2016-06-22 16:00:00+00:00	18.555344	-26.599413	45.113817	0.040940	2	18.730229	-8.513234
2016-06-22 17:00:00+00:00	16.202569	-15.314328	31.234170	0.282727	2	16.195606	-2.712282
2016-06-22 18:00:00+00:00	13.130780	-18.616933	31.312701	0.435013	2	13.196807	-0.788224
2016-06-22 19:00:00+00:00	14.626120	-17.392349	31.438804	0.579666	2	14.599765	0.545599
...	...	...	...	...	...	...	...

Figure 11: New dataframe including added features.

3.5 Anomaly detection - Autoencoders

To find water leakage the group decided to look into methods of anomaly detection algorithms and one of them was autoencoder. Autoencoder has proven to be good with many different problems; facial recognition, feature detection and anomaly detection in time series data [14]. Autoencoders is an artificial neural network that uses unsupervised learning to train the model. Figure 12 shows the three main parts in an autoencoder; encoder, decoder and a bottleneck. The encoders job is to read and compress the data so it is able to go through the bottleneck. Decoders job is to take the data from the bottleneck and try to reconstruct the original data. The model is trained to minimize the reconstruct error, that is the error between the input data and the reconstructed data (output data). When the model is trained right, and the input data is approximately similar to training data, the model may be able to reconstruct the data with low error. But if the input data is very different from the test data, the model may have problem reconstructing the data, and therefore gets a bigger reconstruct error. By monitoring the reconstruct error it is possible to build an anomaly detection algorithm. High reconstruct error indicates unusual data passed through the model, also called anomaly.

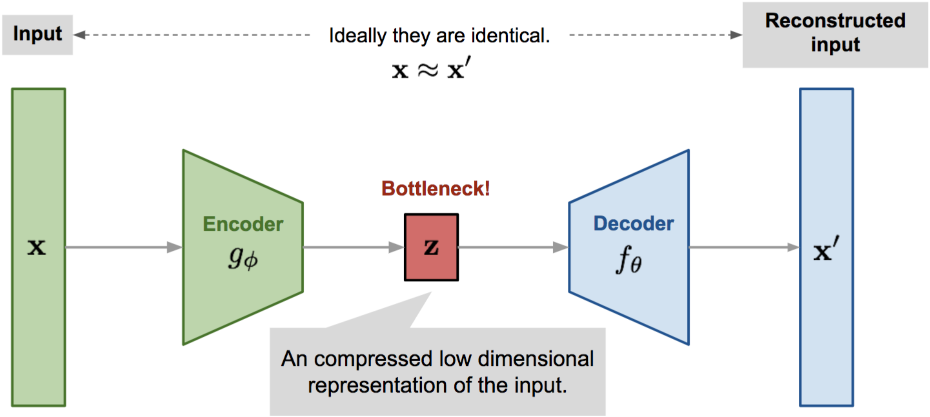

Input
Ideally they are identical.
$\mathbf{x} \approx \mathbf{x}'$
Reconstructed input
$\mathbf{x}$
Encoder
g_ϕ
Bottleneck!
$\mathbf{z}$
Decoder
f_θ
$\mathbf{x}'$
An compressed low dimensional representation of the input.

Figure 12: Representation of an autoencoder model [15].

There are many different configurations to build the encoder and decoder in an autoencoder. One way is to use LSTM layers as they have proven to work well in such problems (anomaly detection in timeseries). LSTM (Long Short-term Memory) is a type off RNN (recurrent neural network). One of RNN's main tasks is to remember and find the connection with previous data, and this works well if the dataframe is not too big. Too big dataframe could result in the RNN starting to forget important information from later data samples and could also result in the vanishing/exploding gradient problem. These problems are solved by using LSTM. LSTM is able to remember information over long time periods by continuously finding which information is important to remember and which information the model can forget. [16]

4 Analysis

This section deals with the construction of the model that is used to detect leakages in the water supply network. At the end of the chapter, the model is tested on real historical water leakage data.

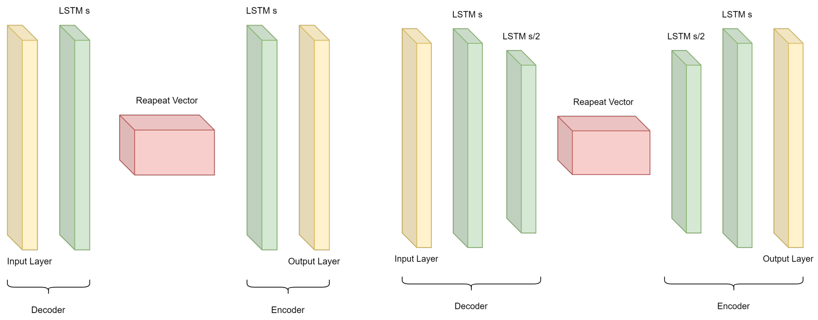
4.1 Modeling

To find the best suitable model for this task a cost function is needed. A cost function is a function which the model tries to minimize or maximize and will after doing so optimize the model itself. For autoencoder the mean absolute error (MAE) function was chosen as a cost function. Formula 1 shows how the MAE is found. The function uses the mean of all absolute errors in the prediction, where the error is predicted value (y_i) subtracted by input value (x_i).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| = \frac{1}{n}\sum_{i=1}^{n} e_i \tag{1}$$

The MAE is found on a separate dataset, the validation set, not the train- or test dataset. The reason for this is to not influence the model by optimizing it on the test dataset. The validation set, as it is called, is in this case 20% of train dataset. When a cost function is chosen, the right hyperparameter is needed to optimize the function, in this case minimize the error and therefore minimize the MAE.

As discussed in chapter 3.5 the encoder and decoder are assembled by LSTM layer(s). To find number of layers two models were made, first one model with one LSTM layer in the encoder and one LSTM layer in the decoder, total of two LSTM layers. Second model had two layers in the decoder and encoder, total of four layers. Figure 13 shows the two different models. When the number of layers is chosen, the size of each layer needs to be chosen. When there is more than one layer in the encoder, it is normal for the next layer to have half the size of previous layer [17]. Same for decoder but in opposite direction. Figure 13 also visualizes this.



(a) Two LSTM layers

(b) Four LSTM layers

Figure 13: Different number of layers [18].

The batch size is the size of each batch of data going through the model. Lastly different types of optimizers are evaluated. All these parameters were tried with a grid search that try all combinations of the different parameters and make a new model for each combination. Total of 50 models were made by training on data that was considering normal water usage. Table 3 shows the 12 best models and the parameters chosen to minimize the MAE. Figure 14 shows MAE for each epoch throughout the training.

Models	**Layers**	**Size**	**Batchsize**	**Optimizer**	**MAE**
Model 1	2	4	20	Adam	0.456
Model 2	2	4	50	Adam	0.478
Model 3	2	8	20	Adam	0.448
Model 4	2	8	20	Adamax	0.461
Model 5	2	16	20	Adam	0.434
Model 6	2	16	20	Adamax	0.449
Model 7	4	8/4	20	Adam	0.450
Model 8	4	8/4	20	Adamax	0.463
Model 9	4	16/8	20	Adam	0.442
Model 10	4	16/8	20	Adamax	0.461
Model 11	4	32/16	20	Adam	0.444
Model 12	4	32/16	20	Adamax	0.444

Table 3: The 12/50 best models with the parameters and MAE.

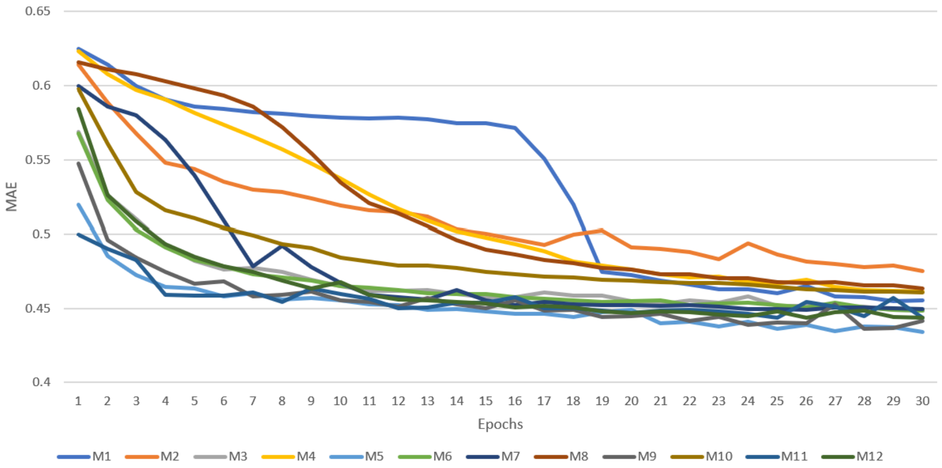


Figure 14: Validation loss (MAE) for 12 best models in 30 epochs.

The model with the lowest MAE was chosen as the best model; **Model 5** with total two LSTM layers with the size 16 on each layer. When the model is built, next step is to test the model on real leakage data. The group that provided the data also provided dates where there has been real leakages. The tests were done by training the model on data considering normal water usage, and then test the model on known water leakage data. Since Autoencoder is an anomaly detection algorithm it can find anomalies on every sample if the data is abnormal. In a water leakage that is not the case. A water leakage will result in multiple anomalies after each other (in series). Therefore, a filter was added to the algorithm. The filter needs 24 consecutive samples to be anomalies before it is considered as a water leakage anomaly.

4.2 Test results

Each test shows a plot with four subplots. The first subplot shows the traindata used to train the model. The second subplot shows the testdata used to test the model. The third subplot shows where the autoencoder finds anomalies and the last subplot shows the 24h algorithm explained previously. In the autoencoder subplots, one (1) indicates anomaly data and zero indicates normal data. The time-frame is identical on the last three subplots.

Test 1:
Trainingdata: 2017.01 – 2018.01. Testdata: 2019.11 – 2020.02. Sudden leakage 2019.12.22. Figure 15 shows the model being tested on a sudden leakage.

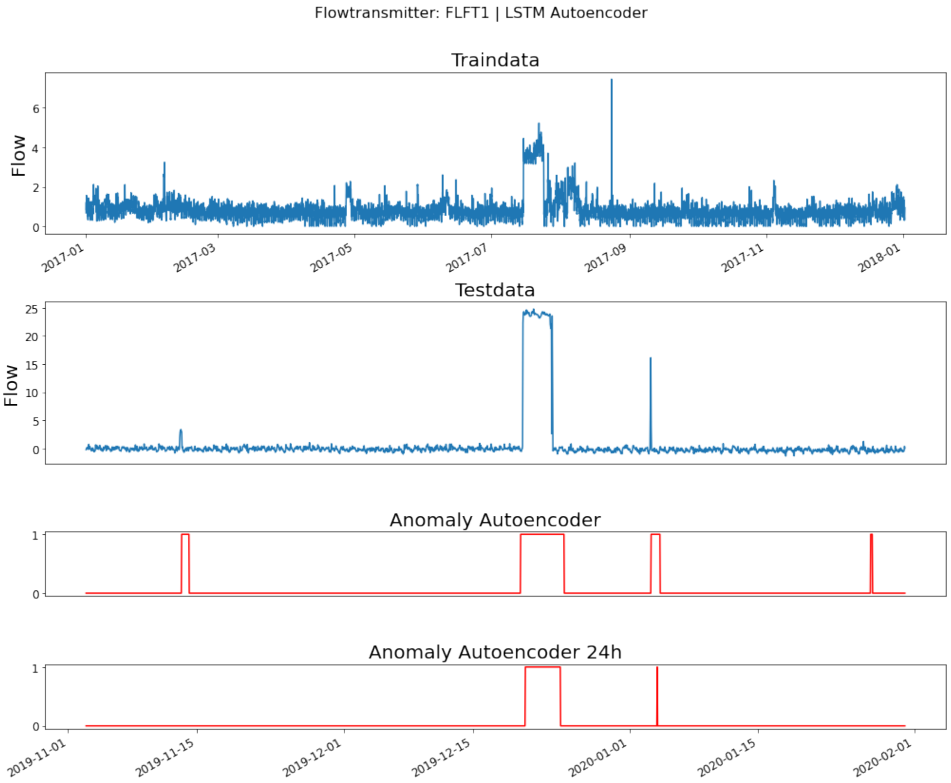


Figure 15: Results from test 1.

The autoencoder finds the main leakage. The autoencoder also finds error in some spikes in the test data before the leakage. It is possible that the spikes are interference. This shows why a 24-hour filter may be necessary.

Test 2:
Trainingdata: 2019.12 – 2020.05. Testdata: 2020.09 – 2020.12. Gradual leakage 2020.10.14. Figure 16 shows the model being tested on a gradually leakage.

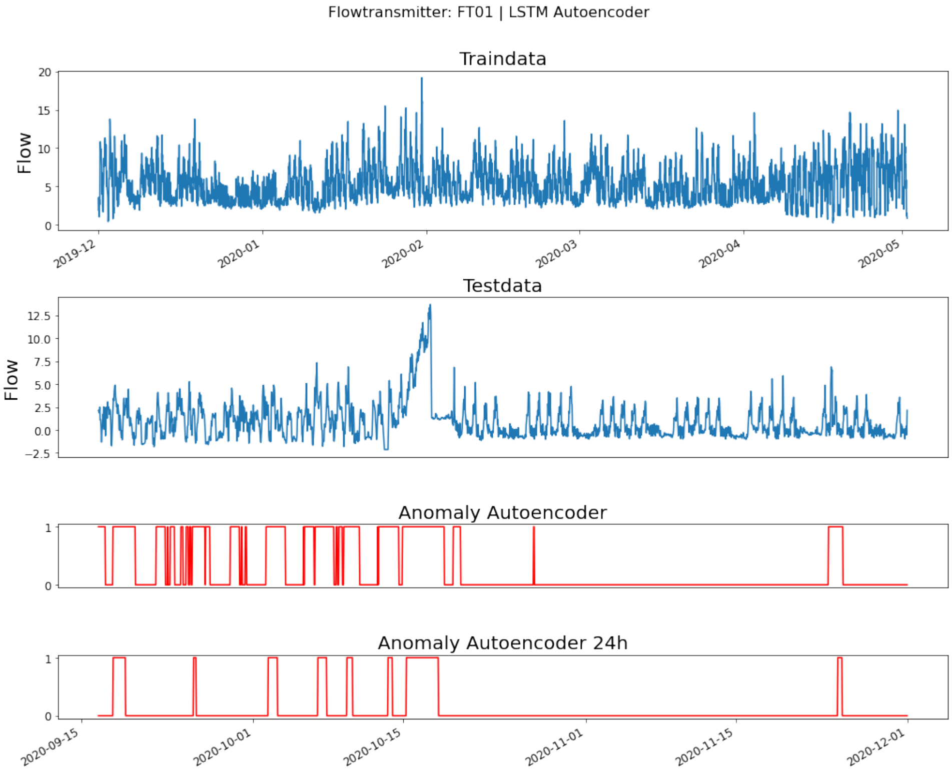


Figure 16: Results from test 2.

In test 2 the model tries to find anomalies on a gradual leakage. Already many weeks before the final leakage, the model finds anomalies. The plot shows some unstable flow before the leakage if compared to after the leakage. This may be a result from a small leakage that makes the flow unstable before it bursts into a more gradual leakage. After the leakage is repaired the autoencoder doesn't detect any significant anomalies. This is an example where the model could be used in predictive maintenance.

Test 3:
Trainingdata: 2016.02 – 2016.03. Testdata: 2016.01 – 2020.02. Sudden leakage 2016.01.24. Figure 17 shows the model being tested on a sudden and gradually leakage.

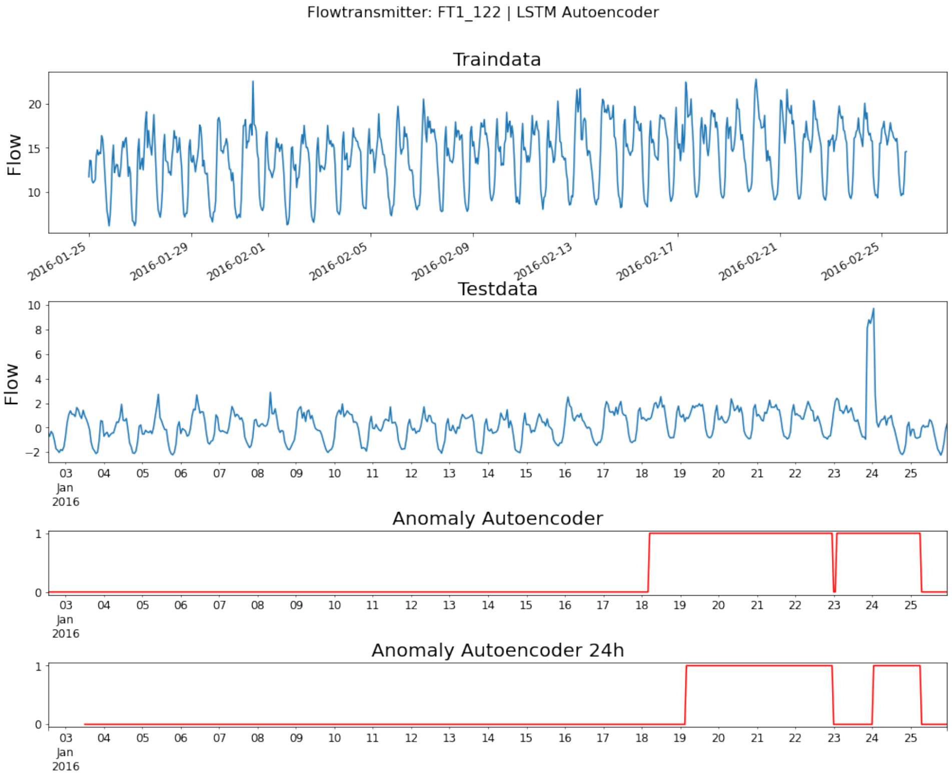


Figure 17: Results from test 3.

Test 3 is trying to find anomalies on a sudden leakage. Again, the autoencoder finds the leakage and also gives information on something abnormal a week before the leakage itself. It's hard for the human eye to see something abnormal, but by taking a closer look it is possible to see that the total volume is increasing before the leakage. This is something the model notices and therefore labels it as anomaly. Again, this is a situation where the model could be used in predictive maintenance.

4.3 Evaluation

Every project should have a goal or a desired achievement for it to have a purpose. By using CRISM-DM methodology we ensure that evaluation is done on business objectives and we stick to business understanding to produce a desired outcome. To ensure that the project is heading towards the desired outcome and the achievements are met, an evaluation of the project is needed. The CRISP-DM methodology has this as a stage for the purpose of achieving what the project aims to achieve. Some of the challenges in this project were to predict leakage, reduce maintenance cost and provide clean drinking water, all by detecting and predicting everything from a minor to a major leakages.

From test 1 and 3 it is clear that the system detected sudden major leakages fast. This is important since the major leakages do the most damage, and can prevent households and other buildings from having water supply, but is also the easiest to detect.

From test 2 it is clear that the system detects a gradual leakage at an early stage. Indication of a leakage is generated approximately a month before the leakage flourishes. This might be positive or negative. If the system generates too many false positives, an alarm might not be taken seriously. Anomaly Autoencoder is sensitive and gives some false positives, but with the 24h filter, the anomaly detection is more reliable.

4.4 Deployment

There are many ways the result from this project can be used. By using the model created, there may be several critical situations that should be avoided. If the situations have already occurred, it will likely be detected earlier than with the current system. The model will detect unusual water flow and warn the user that something is suspicious. By using a smarter warning system instead of only using warning based on a threshold, minor and medium water leakages may be spotted, and false positives can be avoided. Gradual leakages may be fixed before the leakage evolves to a major leakage. Some advantages and disadvantages from implementing that may occur are:

- Advantage:
  - Reduce water waste.
  - Reduce the risk of polluted drinking water.
  - Find and maintain pipes with gradual leakage.
  - Reduce the need of manual surveillance on the sensors.
  - Reduce workload on finding leakages.

- Disadvantage:
  - More false warnings.
  - More visual controls.
  - Maintaining and developing the model.

To have full control of the development this may be implemented on only a few sensors at first, then if the system is working fine, it may be expanded. The final goal would be that the system does calculation on every sensor in real time.

4.5 Key results from the analysis

Three tests were carried out on the dataset using Autoencoder assembled by an LSTM architecture. The Autoencoder detects anomalies in the dataset. However, this can be problematic, as interference from measurements or other faulty outliers may also be detected as anomalies. To solve this, we implemented a 24hr filter which requires 24 abnormal, consecutive data-points for the algorithm to label it as a water leakage.

The results from the tests can be summarized as following:

- The model captures sudden leakages well. However, without the 24h filter, interference in the measurements and other faulty data-points may also trigger anomaly detection.
- Gradual leakages are detected at an early stage. Several weeks in advance, the model captures minor leakages which potentially could turn into major leakages. These minor, gradual leakages are hard for the human eye to detect. Due to the early notifications, the model may be used as a tool in predictive maintenance, if implemented correctly.

5 Interpretation and recommendations

5.1 Implementation plan

In this section, there is developed an implementation plan. The plan in table 4 specifies actions on concrete time-frames after a leakage has been detected. The success criteria for all the recommendations is to fix the leakage and prevent further loss of drinking water.

Time-frame	**Stakeholders**	**Recommendation/Action**
Within 4 hours	Municipal employees	- Visual control - Reduce pressure
From 4-24 hours	Nearest inhabitants	- Notify of leakage and order to boil the water - Ask for location tip
From 1 day to 7 days	Municipality residents	- Notify of leakage and order to boil the water - Ask for location tip
From 7 days to 30 days	Municipal employees	- Mount several sensors in the leak area

Table 4: Implementation plan, time-frame and recommendation.

Some of these recommendations/actions need some elaboration. A visual control is where the municipal employees travel out to where the sensor indicating a leakage is located, and perform a visual control, looking for water splashing from the ground. To reduce the leakage, pressure may be dropped. Putting less pressure on the leakage area will reduce the outflow of water and reduce the leakage as a whole [19].

When a leakage occurs, there is a risk that polluted water can get mixed with the drinking water. Often, the drinking water and the waste water is located closely, this makes it possible for the waste water to penetrate into the drinking water if a leakage occurs. To ensure that nobody gets sick, the nearest inhabitants of the leakage may be recommended to boil their drinking water. Boiling the water kills germs, an prevents people getting sick [20] [21].

Some leakages are very visible. The municipal employees are recommended to ask the inhabitants for information, of where the leakage can be located.

If a leakage is hard to find, and it remains prolonged, it will be recommended to mount several sensors in the leakage area. One sensor can cover a large network of the distribution, and the leakage can be located anywhere in that distribution. With several sensors, the distribution network per sensor is smaller, and the leakage is easier to find.

5.2 Recommendations and limitations

For the dataset, there are multiple limitations and recommendations. To get the total advantage of the dataset, for all of the 164 sensors, the owner should provide a list of dates with registered leakages. This will make it possible to exclude data around these dates from the training data automatically. This will also make it possible to test the prediction to a greater extent, to make prediction at these dates and checking that the prediction is registering the leakage.

A second recommendation is to provide data about the piping in the distribution and where the sensors are located on the pipes. This will make it possible to use volume comparison for detecting leakage. In connection with this it is also recommended to mount several sensors in the distribution. By increasing the number of sensors, each sensor will cover a smaller area of pipes, which means that a volume comparison will be more accurate, and at the same time, the leakage will be easier to locate.

A third recommendation regards the dataset it to avoid null values, or a zero value when there is flowing water. As mentioned in chapter 3.3 this report does not handle these kinds of values because sensors which do not have this challenge have been chosen. The analysis would be overall more complete if all sensors and all of their data was utilized. This could be remedied by e.g., change one way sensors with bidirectional sensors. In this way, even if the water flows the opposite way, there will be a measured value other than zero or null.

Big amount of false positives may increase the workload with manual and visual inspections. This would possibly lessen the reliability of the system and the model. There is also some limitation with the model itself. While building and testing the autoencoder it was hard to find a model that would able to be generalized. This leads to the need of training a new model per sensor. If a new sensor is installed it would not have any historical data. As this model is trained on historical data the sensor needs to capture the normal flow for some time before an anomaly detection model can be made for that sensor.

6 Conclusion

Initially the objectives and achievements are listed in section 2.1. Detecting a variation of leakages was the main objective, and that will result in reducing water loss and reducing the risk of water getting polluted, among other benefits.

Autoencoder with LSTM layers was chosen as the main model to find anomalies. A large number of models with different hyperparameters were tested in section 4, where the 12 best results were visualized. All these models had the main objective of detecting different variations of leakage. As these models were evaluated in section 4.3, it is with certainty that we can confirm the models presented will detect a variation of different leakages. To deploy the models on all of the 164 sensors, and detect leakage in real-time, some improvements need to be made, these are explained in section 5.2. Despite this, the analysis is considered successful and the objectives achievable.

6.1 Further Development

In this section, some ideas for further development are listed. As there is a large number of opportunities, only the most relevant are mentioned.

The analysis is done with the limited resources of a personal computer. On further development it would be interesting to run even more exhausting testing, with different hyperparameters and different numbers of layers and neurons. This will demand more resources and is therefore more suited to be executed in a cloud environment. In a cloud environment it is possible to pay for computational usage, and therefore use more computer resources when the model is being trained. This will decrease the training time from days to minutes, and makes it practically possible.

The analysis focuses on detecting leakage, but in a practical scenario it will be an important aspect to not produce false positives. False positives means that the system alerts that there should be a leakage, when there is not. In further development, a more exhaustive testing should be done, to produce some statistics on false positives. These results should be evaluated, and taken into another iteration of CRISP-DM, where they could be used to adjust hyperparameters etc.

In contrary to more exhausting testing on different models, further development also might include testing different models, and different types of models. In particular, statistical models, not containing neural networks. For example autoregressive models [22], and extension of this model, like autoregressive integration moving average [23] and seasonal autoregressive integrated moving average exogenous [24]. In addition to analysing how these models perform, it would also be interesting to benchmark these models against the autoencoder, in matters of timing and cost, to find out which model produces the best result in least amount of time, and which model uses the least resources.

As more resources introduces a higher cost, it is beneficial to produce a usable result with the least necessary resources.

References

[1] Tim Watson et al. “Maintenance of water distribution systems.” In: *The 36th Annual Conference of American Water Works Association.* 2001.

[2] Marvin Rausand and Arnljot Hoyland. *System reliability theory: models, statistical methods, and applications.* Vol. 396. John Wiley & Sons, 2003.

[3] G Venkatesh. “Cost-benefit analysis–leakage reduction by rehabilitating old water pipelines: Case study of Oslo (Norway).” In: *Urban Water Journal* 9.4 (2012), pp. 277–286.

[4] NIPH. *Drinking water in Norway.* 2017. URL: `https://www.fhi.no/en/op/hin/infectious-diseases/drinking-water-in-Norway` (visited on 09/27/2021).

[5] *CRISP-DM Process Diagram.* Accessed: 2021-SEP-09. URL: `https://commons.wikimedia.org/wiki/File:CRISP-DM_Process_Diagram.png`.

[6] Pete Chapman et al. *CRISP-DM 1.0 Step-by-step data mining guide.* Tech. rep. The CRISP-DM consortium, Aug. 2000. URL: `https://docplayer.net/202628-Crisp-dm-1-0-step-by-step-data-mining-guide.html#show_full_text`.

[7] Desmond Brisbin. *Data Leakage in Machine Learning.* Accessed: 2021-nov-17. July 2019. URL: `https://freshworks.io/design-thinking-process/`.

[8] *Design Thinking: Getting Started with Empathy.* Accessed: 2021-OCT-27. URL: `https://www.interaction-design.org/literature/article/design-thinking-getting-started-with-empathy`.

[9] Samer El-Zahab and Tarek Zayed. “Leak detection in water distribution networks: an introductory overview.” eng. In: *Smart Water* 4.1 (2019), pp. 1–23.

[10] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python.” In: *Journal of Machine Learning Research* 12.85 (2011), pp. 2825–2830. URL: `http://jmlr.org/papers/v12/pedregosa11a.html`.

[11] Youngjoo Kim and Hyochoong Bang. “Introduction to Kalman Filter and Its Applications.” In: *Introduction and Implementations of the Kalman Filter.* Ed. by Felix Govaers. Rijeka: IntechOpen, 2019. Chap. 2. URL: `https://doi.org/10.5772/intechopen.80600`.

[12] Ioannis E Livieris et al. “Smoothing and stationarity enforcement framework for deep learning time-series forecasting.” In: *Neural Computing and Applications* (2021), pp. 1–15.

[13] Josef Perktold. *statsmodels*. Accessed: 2021-nov-17. Nov. 2021. URL: `https://www.statsmodels.org/stable/index.html`.

[14] Chunyong Yin et al. "Anomaly Detection Based on Convolutional Recurrent Autoencoder for IoT Time Series." In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* (2020), pp. 1–11.

[15] Chingis Oinar. *Autoencoders: Introduction and Practical Applications | by Chingis Oinar | Medium.* `https://chingisoinar.medium.com/autoencoders-introduction-and-practical-applications-3eb7b5c1c7fd`. (Accessed on 10/25/2021). Aug. 2021.

[16] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[17] Alaa Sagheer and Mostafa Kotb. "Unsupervised pre-training of a deep LSTM-based stacked autoencoder for multivariate time series forecasting problems." In: *Scientific reports* 9.1 (2019), pp. 1–16.

[18] Peder Ward. *Bruk av Anomaly detection i vanndistribusjonsnett*. Technical report - NTNU. Unpublished. 2021.

[19] Paul F Boulos et al. "Hydraulic Transient Guidelines for Protecting Water Distribution Systems." eng. In: *Journal - American Water Works Association* 97.5 (2005), pp. 111–124.

[20] Luke JURAN and Morgan C MACDONALD. "An assessment of boiling as a method of household water treatment in South India." eng. In: *Journal of water and health* 12.4 (2014), pp. 791–802.

[21] Marcelo Domingos et al. "A new automated solar disc for water disinfection by pasteurization." eng. In: *Photochemical & photobiological sciences* 18.4 (2019), pp. 95–911.

[22] Paul Bourke. *Auto-regression (AR).* `http://paulbourke.net/miscellaneous/ar/`. (Accessed on 11/18/2021). Nov. 1998.

[23] Adam Hayes. *Autoregressive Integrated Moving Average (ARIMA) Definition.* `https://www.investopedia.com/terms/a/autoregressive-integrated-moving-average-arima.asp`. (Accessed on 11/09/2021). 10.

[24] Jan Banas and Katarzyna Utnik-Banas. "Evaluating a seasonal autoregressive moving average model with an exogenous variable for short-term timber price forecasting." eng. In: *Forest policy and economics* 131 (2021), p. 102564.