university of groningen / campus fryslân

Vitens

Internship Report

Minor space

BSc Data Science & Society

Academic Year 2025-2026

Semester 1A & 1B, Year 3

Ditmer de Heer

S5547121

Methods to identify deviating flowsensors

Research on which data science techniques could be used to locate malfunctioning flowsensors in a water distribution network

Student
Ditmer de Heer
S5547121
d.c.de.heer@student.rug.nl

Internship organisation
Vitens
Oude Veerweg 1
8019 BE, Zwolle

Internal supervisor
Loes Bouman
l.bouman@rug.nl

External supervisor
Yvonne Hassink
yvonne.hassink@vitens.nl

Preface

During my time in the Data Science & Society programme I learned about many different data science techniques, when to apply them and how to interpret the results. While courses like 'Statistical and Machine Learning', 'Computer Vision I: AI for Images' and 'Introduction to Speech Technology' broadened the scope of machine learning possibilities, I missed a practical application, in a different form than the Field Project. Where the Field Project provided us with hands-on experience, it did not satisfy my wish to experience what it is like to be a data scientist within a larger whole. That wish came forth out of more societal orientated courses, in which the interaction between both technology and more specifically data science with society played the main role.

This wish led to my search for an internship during my Minor space, one that combined my technical skills and my social orientation. After contacting a wide arrange of companies which I thought would fit into this scope, I received an invitation from Vitens, a Dutch drinking water company, for a meeting to discuss the possibilities of an internship. During this conversation at the main Vitens office in Zwolle it became clear for me that there was a lot I could learn from the Vitens Data Science team. Likewise, Yvonne and Mattheüs – two data scientists who were present – also saw the benefits of having an intern who could provide a fresh look at the organisation and projects they were working on. Soon after, Vitens and I agreed upon doing an internship under the wing of the Data Science team.

At first the main subject of my internship was kept broad, since soon after starting the internship, I was given the freedom to choose a specific subject on my own. I could choose any project from the backlog of the team, as long as I thought I could handle it with my level of knowledge. In the end I choose neither of those projects, because why would I make it easy for myself? Instead, I heard about the problem of deviating sensors in the water distribution network, with the main issue being that it was unknown which sensors, and to what extent, deviated from the real values. It turned out to be a 'classic' "we do not have a ground truth" data science problem, one of which I had some ideas on about how to approach it. After discussing my ideas with Yvonne, my Vitens internship supervisor, we defined it into a task. The main goal of my internship at Vitens was to research different data science techniques that were previously used in scientific literature for similar problems, and find out if they could be used to tackle the problem of identifying deviating flowsensors without having a ground truth.

Table of contents

Introduction

My internship at Vitens started in September and ended in December, having lasted for a total of 3,5 months. Vitens is the largest drinking water company of the Netherlands, providing drinking water to the provinces of Utrecht, Gelderland, Overijssel, Flevoland and Friesland. For my internship I researched several methods for the identification of deviating flowsensors in this large network of which the quality must be maintained. This involved literature research to sensor failure and calibration, the creation of synthetic data, the development of several machine learning models and the search for relevant evaluation metrics for these models. The final product consists of an advice on how to further build upon the developed methods and what other possibilities could be explored in the future. It is supported by the results of different versions of machine learning models I build and available literature and research. This internship report will cover these 3 and a half months at Vitens, including some more information about the company, my project and my development during my internship.

Description of Internship Organisation

Internship Organisation

Out of the 10 water drinking companies in the Netherlands, Vitens covers the largest area of the country, providing clean drinking water to approximately 5,8 million clients, ranging from households to factories (*Vitens; Duurzaam drinkwaterbedrijf*, 2025). To do this, Vitens extracts groundwater and makes drinking water out of it through extensive processes that are finetuned for the unique compositions of water of every extraction location. Afterwards the drinking water has to be transported through an extensive network of pipes, that run from the production locations to every client (*Vitens; Organisatie*, 2025).

This complex continuous operation comes with many challenges: A growing population causing a higher demand, less precipitation during summer periods resulting into lower water levels, and a growing complexity in the network. These challenges ask for a wide range of knowledge within the company, ranging from mechanics that have the skills to install or repair crucial elements in the network and developers who do research and make changes to the network to mitigate those challenges.

During my internship I was part of the data science team, consisting of 4 data scientists. The team falls under the umbrella of the Business Development (BD) department of Vitens. The main roll of BD is to do research, explore new technologies and do experiments to find new innovative ways to help the company.

In my internship I worked hybrid, with on average about 2 days a week in the office and 3 from home. Every day we worked from home we had a daily start-up session in which we either discussed the most important things for that day or had some small talk to start off the day.

Internship Assignment

As mentioned before in this report, my main internship assignment consisted of researching new innovative ways to detect and locate deviating flowsensors in Vitens' network, officially they may deviate 0,5% at maximum. An employee from another department came to the BD team on one of my first days with this problem, explaining that there are currently no automatic techniques to identify those sensors. While there was no ground truth available to test such methods, I had some ideas on how to approach this problem.

My first step was to search for relevant literature in which similar problems were either solved or provided insight in how tackle such issues. Studies mostly used only the flow as a variable to determine if sensor deviated instead of using multiple variables such as pipe width, material or age. The literature research eventually resulted in two main methods that were promising to try, a Long Short-Term Memory (LSTM) Autoencoder model and an Extended Kalman Filter (EKF). Both models rely on recreating a time series containing faulty measurements without those faults, and then comparing this reconstruction with the originally measured time series. This comparison can tell something about how accurate the measured time series is, and therefor how well the flowsensor works. One precondition is that the models can correctly reconstruct said time series without introducing model error in the data themselves. To test this, I needed faulty time series data and a clean time series, the ground truth. The main problem, however, remained the lack of Vitens' data to test those methods in their effectiveness in reconstructing those time series. This is where synthetic data started playing a role.

After consulting the other data scientists on how to tackle this problem, the proposition was made to create my own synthetic data to evaluate the models in their

ability to identify deviating flowsensors. At first, I tried to create my own collection of time series using basic math functions, these however could by far not capture the complexity of real life waterflows in a Water Distribution Network (WDN). Instead, I learned about EPANET, an open-source software package to simulate WDNs and generate data using so called inp-files containing information on physical characteristics of the water network such as pipe length, material, location etc. (US EPA, 2014). In Python I used the WNTR library which uses this EPANET software to generate synthetic data based on Net3, an in literature often used network for research on WDNs (*Overview — WNTR Documentation*, n.d.).

Running the simulation based on Net3 was only one half of the needed synthetic data, since this only resulted into a clean time series, while for the experiment I needed both clean and 'noisy' data to represent deviating flowsensors. For this I developed two functions, one that can either add random or proportionate noise to the time series – depending on the mode you select – and one function that slowly moves the entire baseline, in case a sensor slowly deviates over time. The amount of noise and movement that is added to the time series can be adjusted when applying it to the clean time series.

While working on the synthetic data I was already building the first version of the LSTM Autoencoder model. This model consists of an encoder and decoder part. The encoder learns to extract the most useful representation of the data and compresses the amount of data before it is given to the next layer. The decoder learns how to unpack this compressed representation into a full time series, one which should look similar to the original input. The theory behind applying the LSTM Autoencoder to this problem is that the noise in the flowsensors is taken out during the compressing part of the model, because this noise is not relevant to the underlying pattern in the time series, and therefore does not return in the reconstruction. The type of noise does matter in the reconstruction, for instance random noise is almost taken out completely, while drifting is not (Chen et al., 2025; Shin et al., 2024).

After obtaining the synthetic data I could run a first test of this LSTM Autoencoder, which did not work very well at all since it did not capture the general pattern of the time series at all. To help it with training I searched for similar projects on GitHub, finding that many of them used a time window function to take small samples out of the time series and feed those into the model, which helps the model

with learning the underlying pattern. After recreating this function and its counterpart function that stitches all the samples back together, I trained the LSTM Autoencoder again, leading to better results because it could understand that there was a returning daily pattern visible. There was still one problem however, for time series with a lower range the model seemed to just provide one almost straight line based on the mean of the time series. To tackle this, I normalized the time windows before putting them into the model based on their mean and standard deviation, which is most of the time called Z-normalization. This not only resulted into better results for those time series that were reconstructed as a straight line, but also in better results for all the sensors. Pattern details such as smaller peaks at the top or bottom of the time series were more easily identified by the model.

The next step was to evaluate the effectiveness of the LSTM Autoencoder in reconstructing the flowsensors' time series without noise. I wanted to use metrics that could also be used for the Extended Kalman Filter I still had to build, so that I could compare the methods afterwards. For this I decided to use an improvement percentage, based on a comparison between the Mean Squared Error (MSE) of the clean data and noisy data, and the MSE of the clean data and the reconstruction of the model. This percentage shows how much closer or further away the values of the reconstruction are to the clean data than those of the noisy data. A positive percentage means that the reconstruction has more accurate values than the deviating time series, while a negative percentage means that the reconstruction is worse than the measured values.

For the LSTM Autoencoder, all the synthetic noisy sensor data with an overall error of 0,5% or higher were improved by the model, with improvements up to 94%. At the same time, reconstructions for sensors with an overall error lower than 0,5% were worse than the measured noisy data, having negative percentages that go down to -200%. This can be seen as making a copy of an original print, the copy will be worse than the original. A reconstruction of a close to perfect measured time series will be worse than the measured values. Nonetheless, it was still important to take this into account, for if the model would be applied to real data, it is valuable knowledge to know that the model does not always make the time series better.

After getting these results I tried different versions of the LSTM Autoencoder, creating versions with more layers and versions with more nodes per layer. These models however started overfitting (the model started following the noise, leading to

worse RMSE values), so it was decided that a LSTM Autoencoder with 2 layers in both the encoder and decoder part of the model with 128 and 64 nodes was the best fit.

Next in the process was the creation of the EKF model, which is not a machine learning model that first has to train and can then be applied directly, but relies on input along the way. In essence, it first learns how certain parameters should be set, and then along the way continues learning and adjusts the parameters accordingly. The model first needs a 'starting point' from which to plot a time series and a 'speed' which stands for the models' sensitivity to new input. The EKF tries to recreate a time series based on input from other sensors or the sensors' own measured time series. This input in combination with the set speed decides how steep and in which direction the line in the time series has to be drawn (Huang et al., 2024).

The results of this EKF model were worse than those of the LSTM Autoencoder. While it still improved most flowsensors' noisy time series, it did so in a worse way than the previous method did. Improvement percentages were noticeable lower, although the sensors with negative percentages were slightly better reconstructed compared to their LSTM Autoencoder counterparts. However, they were still all negative, meaning that the reconstruction was worse than the originally measure time series.

Before putting the EKF method aside entirely I tried to tweak the settings of the model to improve the outcome. I had already normalized the values during the first test, having learned from the first model, so there were two things I tried. The first of them was applying a grid search to look for the best starting position and speed of the model, this improved the results a little bit, but it was still worse compared to the LSTM Autoencoder. The second thing I tried was increasing and decreasing the speed setting of the EKF, meaning that I changed the models' sensitivity to input it received along the way. Both increasing and decreasing the speed did not help, it only led to overfitting and underfitting. After these results I decided with Yvonne to continue with the LSTM Autoencoder.

This was not the end point however, for this was only the first step of deciding if the method was good enough to continue. The more interesting part was seeing if, with only using the noisy time series and the reconstruction of the model, we could find out if and how much a flowsensor deviates. For this I calculated every value you could calculate without using the ground truth, which involved different versions of

the Root Mean Squared Error (RMSE) between the noisy and reconstructed time series and R^2. At first, I did not see a correlation between these values and the average deviation of the sensors, so I started combining different evaluation metrics. This led to the creation of the anomaly score, using the with range normalized RMSE value and R^2. The anomaly score only had a correlation of 0,38 with sensor deviation and was also not very easily explainable or made insightful because of its complexity.

Having hit a wall going into this direction I decided to train a decision tree on my evaluation metrics, with the goal of deciding whether a sensor deviated more than 0,5% or not. To my pleasant surprise it categorized the sensors from the synthetic data perfectly, based on the RMSE value of which I at first thought would be too simple to capture deviating flowsensors. When further exploring this path, I found out it had a correlation of +/- 0,97 with the average sensor deviation, meaning that it also correctly identifies how much a sensor deviates.

When testing this on the real data from flowsensors in Vitens' network this theory proved problematic. According to the results, some sensors would deviate more than 200% from their real values, something that when looking at other indicators can simply not be true. After looking further into the real data, I found out that bigger pipelines with sensors showcase different behaviour than smaller pipelines. The flow could change more rapidly between two neighbouring points in time in these bigger pipelines than in the smaller ones, changes so abrupt that the LSTM Autoencoder did not capture them in its reconstruction, leaving those sudden spikes out like they were noise. This resulted in higher RMSE values compared to the smaller pipelines, meaning that a comparison and decision on how much a flowsensor deviates is impossible to make with the current knowledge.

Further testing the LSTM Autoencoder by trying different compositions of layers and types of noise led to the conclusion that next to the problem of the different pipe sizes there is also the problem that when a sensor slowly starts to deviate over time the model does not recognize this. This was seen in the graphs, the LSTM Autoencoder followed the noisy drifting pattern perfectly, instead of the expected ground truth. After considering this it was concluded that further development was needed, either by categorizing the flowsensors from small pipelines to large pipelines and providing a different RMSE threshold per category or by trying another model to tackle the drift problem.

These approaches are outside the scope of my internship project, but I have looked into them. The first option of making categories and defining which RMSE value indicates a deviating flowsensor is time and resource consuming. This because you would have to find out at which size the pipeline's width starts effecting the RMSE value and then decide what RMSE value will become the threshold without knowing how many categories and thresholds you will end up with. The second option, the one where other methods are explored, might be more probable to explore. In one of my last weeks there was a talk about DiTEC, a system to simulate WDNs and predict pressure levels in such a network based on a given situation (Degeler et al., 2025). This could be an area to explore for Vitens after my internship, the research and development of DiTEC is still under development however, and it is uncertain when flow is added to the system.

Internship Results and Output

The final product of my internship is the documentation of my research on possible methods to identify deviating flowsensors. This documentation consists of both an extensive document containing information on all my steps and thought patterns and two jupyter notebooks containing the code to generate synthetic data and code to run the two methods I researched. Furthermore, it includes the synthetic data and its reconstructions and the relevant evaluation metrics.

Pictures of the code can be found in Appendix D, due to the sensitivity of the data and research only a few snippets were taken out to showcase an example of the code. In Appendix C are some graphs and images to showcase the results of the models and give an example based on the synthetic data.

In addition to the above-mentioned documentation, I also held two presentations during my internship to explain my research. The first presentation was about halfway through my research during a biweekly BD meeting. For this presentation it was important to not make it as technical as it is, since while the data science team is part of BD, there are also BD employees that do not have the technical knowledge to fully understand the workings of a LSTM Autoencoder. I decided to keep the explanation of the model relatively simple, explaining only what it does without going into the details of why it works. After this presentation suggestions

were done to look into a slower deviation over a longer period of time and that maybe this method could also be applied to different type of sensors. I took the first suggestion into account later during my internship by running this test. The second suggestion went outside the scope of my internship, but would certainly be interesting for Vitens to further investigate.

The second presentation was towards the end of my internship and was during a data science deep dive, a biweekly meeting in which a technical presentation was held among interested developers, network analysts and data scientists. In this presentation I could go more into detail and also discuss the EKF method, which was not tested yet during my first presentation. Since it was at the end, the main goal was to explain what I had done and ask if the others had ideas on how to tackle the issues I encountered. The conclusion of this presentation was that I had done interesting work, and that the promising aspects DiTEC – while still being under development – could prove interesting for this problem.

Evaluation

Learning Outcomes

Before the start of my internship, I had to define a set of learning outcomes which I would further develop myself in. Since the assignment was more precisely defined afterwards, the formulation of the final learning outcomes has changed a bit, but the content remained the same. Below is a list containing the learning outcomes, for the original learning outcomes I refer to Appendix A, the internship plan I had to define beforehand.

- Develop and apply evaluation methods to assess the accuracy of the models.
- Identify the consequences of deploying the models within the organisation, while mitigating the negative effects.
- Communicate and interpret the models to other stakeholders within the company.
- The application of legal frameworks (such as the GDPR) on real life scenarios when handling sensitive data.

- Work professionally within the company by understanding company policies and practices.
- Recognize business implications of deploying data science techniques and ensure those are aligned with the company's interests.

During my internship I learned a lot about evaluating machine learning methods. It at times was difficult to think of metrics I could use to compare the two models and also look for an indicator of sensor deviation. What I learned with the anomaly score I created was that sometimes when your thought patterns become so complex that even yourself start losing track you have to go back to the basics. It was then, after taking a step back that I saw the correlation between the RMSE and sensor deviation, something I had not seen earlier.

This assignment, because of its research-oriented approach, did not focus too much on the consequences of deploying potential finished models within Vitens. Even though, the original reason I wanted to do this is because I saw that potentially solving the issue of deviating sensors could contribute a lot to the company. The data of these sensors are used to take action, and by identifying faulty data streams you can make better decisions knowing this.

Communication, a professional work attitude and recognizing business implications of my research were one and the same for me during the internship. They were intertwined in the sense that I found it important to not only share my progress within the data science team, but also with the original stakeholder that came to us with this problem. I regularly planned a meeting with him and others from his department to share my results, making sure they knew what was going on. I did this because soon after the start of my internship I saw that some people within the company did not dare to trust certain models or their conclusions. I wanted to ensure that the people I worked with understood what was going on, so that from the start they knew why and how the model came to its conclusions.

The consequence of working mostly with synthetic data was that I did not have to work with any legal frameworks regarding sensitive data. It was not until a later stage of my internship that I started working with real flowsensor data. For this data I had to work in a secure online environment instead of in Visual Studio.

Next to my predefined learning goals I also learned a lot about hydrology and simulations of complex water networks. Since the domain of WDNs is quite specific I

did not know a lot about the workings of it, but during my internship I learned more and more about it, either through literature or through learning about projects others were working on. Everyone, from data scientists and data engineers to others within the BD team was willing to tell me about their projects and by doing so deepen my understanding and knowledge on WDNs.

I also grew personally during this internship. Most of the time when I meet a new group of people or situation, I always first try to fully understand the interactions and relations between every element before I find my own place and feel confident enough to actively share my vision. At the start I was a bit overwhelmed with the amount of completely new knowledge that came my way, unsure how to find my role within the team. What helped finding this place was the trust of the entire data science team and especially Yvonne who, during our weekly meetings, expressed that she found my knowledge and view valuable to the team. In the later weeks of my internship I noticed that I became more comfortable with sharing my thoughts and ideas, leading to more interesting conversations and bonding than before.

Contributions to the Company

By doing this internship I did not only learn myself but also shared my knowledge and view with the data science team and others I worked with. The results of my internship assignment show which methods may and which may not be interesting to explore in the future. Also, during the countless of meetings, from the daily stand-ups, BD meetings and weekly data meetups, I had the opportunity to both learn about and contribute to other ideas, projects and areas of interest. My contributions and interactions with others were positively received by the company, which expressed itself in an offer to continue working at Vitens during my bachelor's degree as a work student.

Value of the Programme

Throughout the courses of Data Science & Society I learned a lot of skills, that individually may not make sense together, but I experienced that the combination of social and technical courses made that I could more easily grasp the new knowledge about WDNs. My research skills mainly helped with my internship assignment, which started as literature research and developed itself into a technical application of

theories I read about. Especially the presentations, student led classes and group projects helped with communicating my findings with others, both in an approachable and technical way. This helped making connections with people and overall made the internship assignment better as a whole.

Future Development

These past few months at Vitens taught me a lot about how a data scientists operates within an organisation. I am aware that Vitens is of course only one organisation and that at other companies things may be organised or done differently, but nonetheless this experience has been very valuable for me. Since I was directly, from the first day, involved in BD meetings, data science deep dives and the weekly data science + engineering updates I immediately became part of the team and company. These meetings not only gave me hands-on experience with data science, but also with other fields.

For the future of my internship assignment for Vitens I have documented my research, code and shared my results with different stakeholders within the company. The problem of deviating flowsensors remains, but it might now be more clear which approaches might and might not work. In addition to this, I will continue working at the Vitens' data science team for the upcoming half year, perhaps not on the same topic as during my internship, but I am sure that this will provide me with even more experience and ideas for my future.

References

Chen, X., Fan, Y., Lin, X., Ding, Y., You, D., & Zhou, W. (2025). Adversarial Data Augmentation Enhanced LSTM Autoencoder for Anomaly Detection in Industrial Pipeline Networks. *Journal of Signal Processing Systems*, *97*(2), 197–207. https://doi.org/10.1007/s11265-025-01962-x

Degeler, V., Hadadian, M., Karabulut, E., Lazovik, A., van het Loo, H., Tello, A., & Truong, H. (2025). DiTEC: Digital Twin for Evolutionary Changes in Water Distribution Networks. In T. Margaria & B. Steffen (Eds.), *Leveraging*

Applications of Formal Methods, Verification and Validation. Application Areas (pp. 62–82). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-75390-9_5

Huang, Y., Thomas, M., Bartos, M., & Sela, L. (2024). Employing Extended Kalman Filter for Faulty Sensor Detection in Water Distribution Systems. *Engineering Proceedings*, *69*(1), 28. https://doi.org/10.3390/engproc2024069028

Overview—WNTR documentation. (n.d.). Retrieved December 10, 2025, from https://usepa.github.io/WNTR/overview.html

Shin, Y., Na, K. Y., Kim, S. E., Kyung, E. J., Choi, H. G., & Jeong, J. (2024). LSTM-Autoencoder Based Detection of Time-Series Noise Signals for Water Supply and Sewer Pipe Leakages. *Water*, *16*(18), 2631. https://doi.org/10.3390/w16182631

US EPA, O. (2014, June 24). *EPANET* [Data and Tools]. https://www.epa.gov/water-research/epanet

Vitens; Duurzaam drinkwaterbedrijf. (2025). https://www.vitens.nl/Over-Vitens/Elke-druppel-duurzaam/Rubriek-Duurzaam-drinkwaterbedrijf

Vitens; Organisatie. (2025). https://www.vitens.nl/Over-Vitens/Organisatie

Appendices

Appendix A; Internship approval form

university of groningen

Form for Approval of Internship

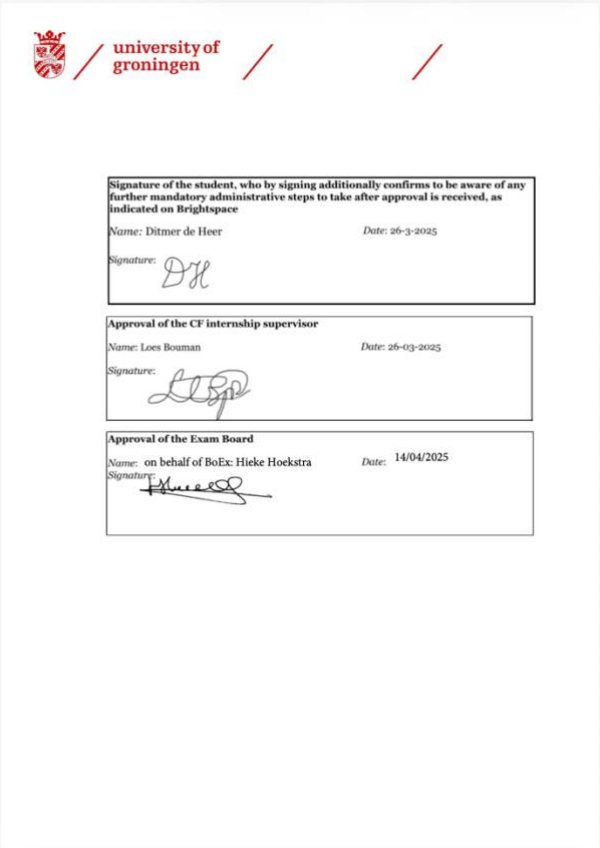
Student name	Ditmer de Heer
Student number	S5547121
Name of internship	Internship Data Science
Amount of ECTS	20 ECTS
CF supervisor	Loes Bouman
Internship organisation (and location)	Vitens; Oude Veerweg 1, 8019 BE Zwolle, Netherlands
Supervisor at internship company	Yvonne Hassink
Supervisor contact details	yvonne.hassink@vitens.nl

Justification

List the main topics of the internship.	For my internship at Vitens I will contribute to the following topics and tasks: • The evaluation and benchmarking of a Large Language Model (LLM) by finding ways to evaluate its performance. • Considering the possible ethical, legal and social effects of deploying a LLM within the organization. • Seeking for ways to explain or interpret the decisions and/or results of a LLM to other departments within the company to ensure it is used correctly. • Assisting in handling sensitive data while ensuring privacy and security are in line with regulations like the General Data Protection Regulation (GDPR).
Mention the learning outcomes that will be achieved after successful completion of the internship.	Based on the topics and tasks described above I plan on achieving the following learning outcomes: • Develop and apply evaluation methods to assess the accuracy of a LLM. • Identify and mitigate the ethical, legal and social consequences of deploying a LLM in an organisation. • Communicate and interpret LLM decisions for non-technical colleagues and departments. • The application of legal frameworks such as the GDPR on real life scenarios in the context of client data. • Work professionally within the company by understanding company policies and practices. • Recognize business implications of deploying data science techniques and ensure those are aligned with the company's interests.
Specify why the internship adds to your DSS programme.	The main reason for choosing to do an internship during my minor is to gain more practical experience with data science tools and tasks; and with working in a business environment which comes with its own challenges. Currently there are not any clear measurement methods available for LLMs and thus deploying them can cause potential risks. By working on a LLM and its evaluation metrics I combine both the technical - creating and adjusting it - and the societal - ethical considerations on when it is good enough to be used in practice - aspects. This can be especially challenging in a corporate setting where different interests from different departments or people may play a role. Furthermore, working with sensitive data in line with the GDPR also provides an opportunity to combine the Governance courses of the DSS program with the Data Science courses. So far in the programme we mainly focussed on those two separately, while in reality they are intertwined. During this internship I can get the chance to combine in practice and learn from experienced data scientists how they deal with this. To summarize, I think this internship allows me to obtain practical experience with the different topics I learned during the DSS courses, which would enrich and further develop the knowledge I gained from the DSS programme.

university of groningen

Specify the ECTS and workload by giving an estimated time schedule of the internship where you describe the frequency and planned period of meetings with your CF supervisor and your internship supervisor. (Remember: 1 ECTS equals 28 hours of workload.)	The workload of the internship will be 560 hours, equalling 20 ECTS. This is based on 40 hours a week, for 14 weeks. The internship will start at the end of august or beginning of september. The exact starting date will be decided later with the internship organisation. Meetings with the CF supervisor will be held if either the supervisor or I request one. To enable continuous communication, I will send a short overview of the state and tasks of the internship to my CF supervisor every 2 weeks to keep them informed. Halfway through the internship an interim evaluation will be organized where the CF supervisor and internship supervisor will meet to discuss my progress - preferably with me present as well. Meetings with the internship supervisor will take place on a regular basis since I will be working together with them.
Describe the method of assessment and the assessment criteria.	For the method of assessment and the assessment criteria I refer to the assessment forms appended to the 24-25 Internship Manual DSS.



university of groningen

Signature of the student, who by signing additionally confirms to be aware of any further mandatory administrative steps to take after approval is received, as indicated on Brightspace

Name: Ditmer de Heer *Date:* 26-3-2025

Signature:

Approval of the CF internship supervisor

Name: Loes Bouman *Date:* 26-03-2025

Signature:

Approval of the Exam Board

Name: on behalf of BoEx: Hieke Hoekstra *Date:* 14/04/2025

Signature:

Appendix B; Internship Logbook

Week	Hours	Tasks done/progress update
Week 36	40	- I was introduced to the company and co-workers. - Found a topic I am interested in and started working on it. - Learned quite a lot about how AI systems are integrated in a company in a safe way.
Week 37	32	- Started with the research for my topic. - Started making data for the project, the real data doesn't have a groundtruth, so I will have to make my own data. - Contacted people within the company that could help me.
Week 38	43	- Continued creating the synthetic data, with a potential new library I will explore next week. - Met other interns within the company with whom I visited a production location of the company.
Week 39	37	- Made synthetic data using the wntr python library which uses EPANET to stimulate water networks.

		- Started creating a LSTM autoencoder.
Week 40	40	- Finished making the first LSTM autoencoder model. - Started running and evaluating the model on synthetic data.
Week 41	38	- Updated the first LSTM model to improve it's performance. - Created new metrics that can be used in situations where the ground truth isn't known. - Started taking notes for the documentation for the data science team of Vitens.
Week 42	40	- Went through my notebooks again to take some errors out. - Further documented the first method now that it's test version is finished. - Started working on a presentation for the Business Development team on the first method.
Week 43	34	- Continued with the creation of the presentation based on feedback from the other data scientists. - Started working on the Extended Kalman Filter method.
Week 44	40	- Experimented with different preprocessing steps and values for the Extended Kalman Filter. - Started documenting the risks of the anomaly score, since it isn't perfect but still better than random guessing.
Week 45	34	- Researched the possibility of using the RMSE as an indicator. - Visited the lab in Leeuwarden to get more feeling with the different aspects of Vitens and suggested the usage of computer vision applications in counting the amount of bacteria. - Had a 'halfway through' meeting with Yvonne and Loes on my progress, both were very positive on my learning curve and performance during the internship.
Week 46	40	- Received a dataset of measurements of sensors in the network of Vitens. - Started a first run on the real data, which didn't turn out the way I had hoped.
Week 47	40	- After analyzing the results of the real data it was concluded that the metrics I used do provide a certain indication on trustworthiness, but in its current state doesn't perform well enough to use. - Listened to a symposium on AI in the drinking water sector. - Researched DiTEC, an AI system network for drinking water networks.
Week 48	38	- Ran some final tests on the LSTM Autoencoder to clarify details for in the report for Vitens. - Emailed the creators of DiTEC to plan an online meeting. - Had some talks on the year plan for the Vitens data science team.

Week 49	32	- Meeted with one of the researchers and creators of DiTEC to talk about my project and the possibilities of using DiTEC in it. - Defined follow-up steps for the project for after my internship. - Started combining all pieces of my code into two notebooks for documentation for Vitens.
Week 50	40	- Finished combining all of my code and asked another data scientist to check it. - Finished the documentation for Vitens. - Started writing my internship report for university.
Week 51	32	- Final week of my internship. - Had a christmas dinner with the team. - Finished with my internship report and handed over the remaining documentation to Vitens.

Appendix C; Relevant graphs and figures

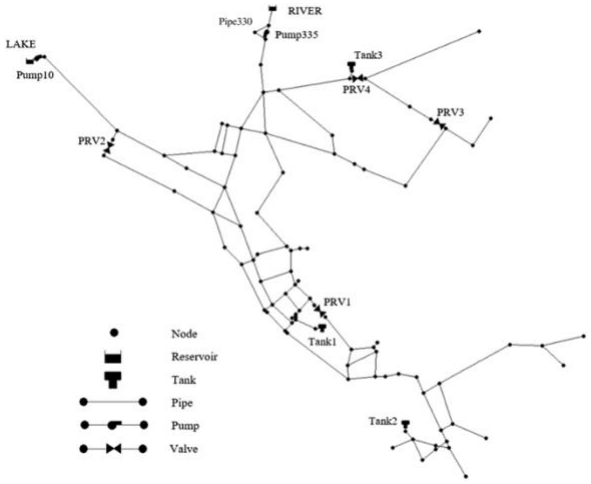


Figure 1, Net3 network structure

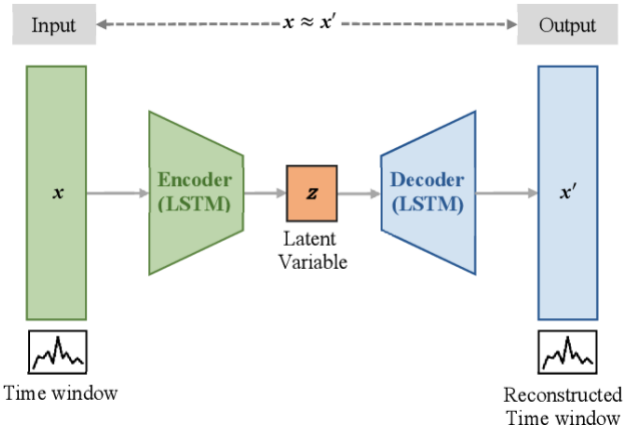


Figure 2, a schematic representation of a LSTM Autoencoder

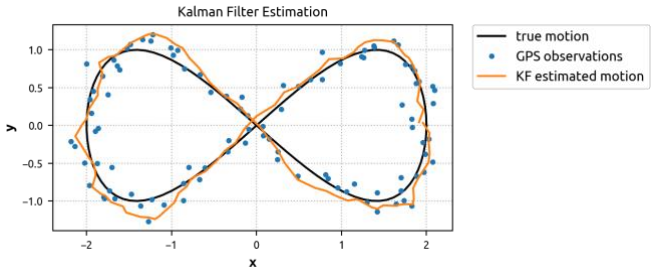


Figure 3, an example of how the Extended Kalman Filter is used in practice

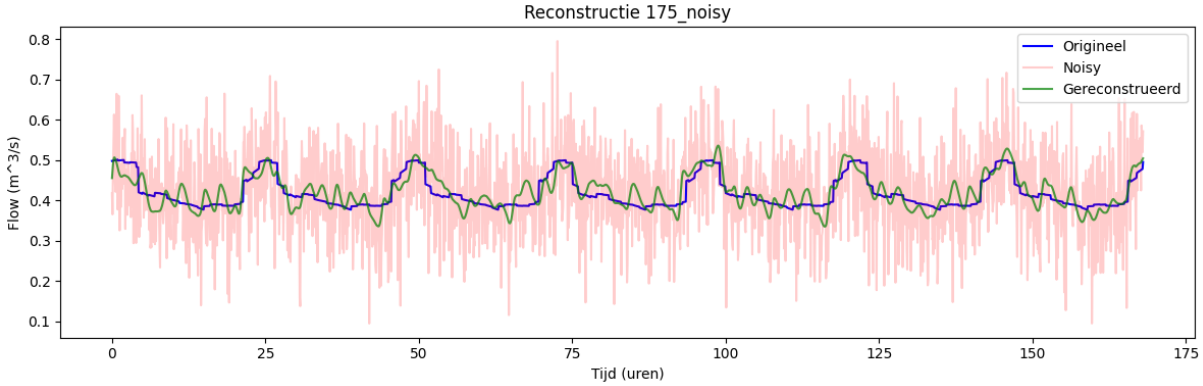


Figure 4, reconstruction of synthetic flowsensor 175 in Net3, made by the LSTM Autoencoder

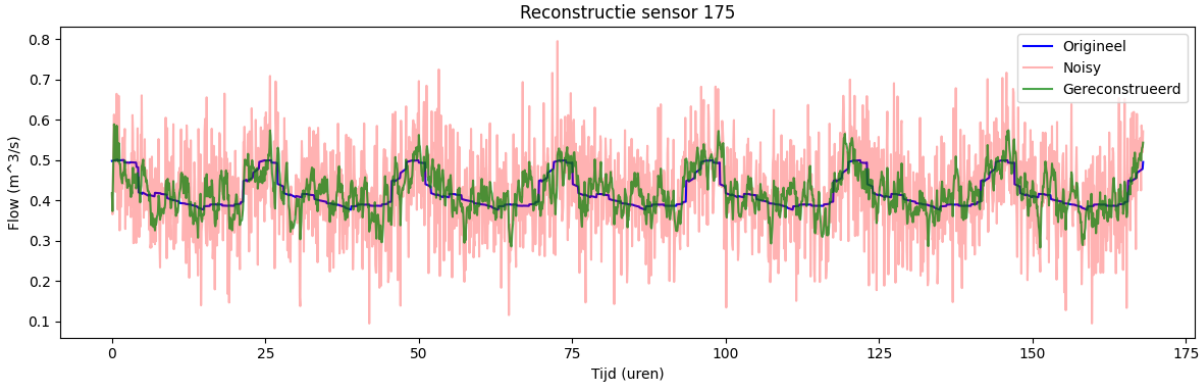


Figure 5, reconstruction of synthetic flowsensor 175 in Net3, made by the Extended Kalman Filter

Appendix D; Code fragments

```python
# Een functie die helpt door overlappende tijdframes te nemen waardoor de LSTM beter kan leren.
# De overlapping zorgt ervoor dat patronen beter herkend kan worden, wat het model helpt bij het voorspellen.
def tijdframes(data: np.ndarray, seq_len: int, stride: int = 1) -> np.ndarray:
    """
    data: (T, F), waarbij T het aantal tijdstappen is en F het aantal features (de hoeveelheid sensoren).
    seq_len: De lengte van elk tijdframe dat de functie moet maken.
    stride: Het aantal tijdstappen dat het verschuift voor het aanmaken van het volgende tijdframe.
    De functie geeft een array (N, seq_len, F) terug, waarbij N het aantal tijdgframes is.
    """

    # De tijd en features van de data wordt opgehaald om gebruikt te worden voor de tijdframes.
    T, F = data.shape
    N = 1 + (T - seq_len) // stride
    X = np.empty((N, seq_len, F), dtype=data.dtype)

    # Een loop die met behulp van de stride tijdframes aanmaakt.
    for i in range(N):
        start = i * stride
        X[i] = data[start:start+seq_len]
    return X
```

Figure 6, code of the timeframe function

```python
# Model architectuur, lagen en nodes kunnen aangepast worden afhankelijk van de complexiteit van de data, zolang de encoder en decoder
# maar symmetrisch zijn.
inp = Input(shape=(seq_len, features))

# Encoder
e = LSTM(128, activation="tanh", return_sequences=True)(inp)
e = Dropout(0.1)(e)
e = LSTM(64, activation="tanh", return_sequences=False)(e)

# Repeat vector
b = RepeatVector(seq_len)(e)

# Decoder
d = LSTM(64, activation="tanh", return_sequences=True)(b)
d = Dropout(0.1)(d)
d = LSTM(128, activation="tanh", return_sequences=True)(d)
out = TimeDistributed(Dense(features))(d)

# Het samenvoegen van het model
autoencoder = Model(inp, out)
autoencoder.compile(optimizer="adam", loss="mse")

# Callbacks om vroegtijdig stoppen en het leerproces aan te passen waar dat nodig is.
callbacks = [
    EarlyStopping(monitor="val_loss", patience=8, restore_best_weights=True),
    ReduceLROnPlateau(monitor="val_loss", factor=0.5, patience=4, verbose=1)
]

# Het trainen van het model, de batch size en epochs kunnen aangepast worden om het model meer of minder lang te laten trainen.
autoencoder.fit(
    X_train, X_train,
    validation_data=(X_val, X_val),
    epochs=20,
    batch_size=32,
    callbacks=callbacks,
    verbose=1
)
```

Figure 7, the LSTM Autoencoder architecture

```python
# Voor elke kolom een plot maken.
noisy_cols = [col for col in data_met_ruis.columns if "_noisy" in col and "_reconstructed" not in col]

for col in noisy_cols:

    # De schone en gereconstrueerde kolommen uit de data halen.
    clean_col = col.replace("_noisy", "")
    rec_col = f"{clean_col}_noisy_reconstructed"

    plt.figure(figsize=(12, 4))

    # Originele schone data.
    clean_col = col.replace("_noisy", "")

    # Plot van het origineel. (Het aantal secondes wordt door 3600 gedeeld om uren te krijgen voor betere leesbaarheid.)
    if clean_col in data.columns:
        plt.plot(data.index / 3600, data[clean_col], label="Origineel", color="blue")

    # Plot van de ruis.
    if col in data.columns:
        plt.plot(data.index / 3600, data[col], label="Noisy", color="red", alpha=0.2)

    # Plot van de gereconstrueerde data.
    if rec_col in data.columns:
        plt.plot(data.index / 3600, data[rec_col], label="Gereconstrueerd", color="green", alpha=0.7)

    plt.title(f"Reconstructie {col}")
    plt.xlabel("Tijd (uren)")
    plt.ylabel("Flow (m^3/s)")
    plt.legend()
    plt.tight_layout()
    plt.show()
```

Figure 8, code for the plots

```python
# Een functie die de hyperparameters van de EKF automatisch afstemt op de data.
def tune_ekf_hyperparams(z, dt=1.0):
    """
    z = meetwaardes (array)
    dt = tijdsinterval tussen metingen

    Vindt de optimale hyperparameters (q_pos, q_vel, r_meas) binnen een kleine tijdswindow van de data.
    Gebruikt hetzelfde window van de data om de RMSE te minimaliseren.
    """
    z = np.asarray(z, dtype=float)
    z = z[np.isfinite(z)]
    if len(z) < 20:
        return 1e-5, 1e-6, np.var(np.diff(z)) + 1e-6

    # Het plaatsen van een grid over mogelijke waardes voor de hyperparameters.
    q_pos_list = [1e-3, 1e-4, 1e-5, 1e-6]
    q_vel_list = [1e-4, 1e-5, 1e-6, 1e-7]
    r_meas_list = [np.var(np.diff(z)) * f for f in [0.1, 1, 10]]

    # De beste afstelling bijhouden.
    best_rmse = np.inf
    best_params = (1e-5, 1e-6, np.var(np.diff(z)) + 1e-6)

    # Grid search voor de beste hyperparameters op basis van de RMSE.
    for qp in q_pos_list:
        for qv in q_vel_list:
            for r in r_meas_list:
                x_est, _ = ekf_filter(z, dt=dt, q_pos=qp, q_vel=qv, r_meas=r)
                rmse = np.sqrt(mean_squared_error(z, x_est))
                if rmse < best_rmse:
                    best_rmse = rmse
                    best_params = (qp, qv, r)
    return best_params
```

Figure 9, Extended Kalman Filter grid search code

Appendix E; Extra's

Figure 10, a visit to KWR during one of my first weeks