University of South-Eastern Norway

www.usn.no

FMH606 Master's Thesis 2023
Industrial IT and Automation

Surveillance system for drainage pumps with the use of machine learning

University of South-Eastern Norway

Vebjørn Rimstad Wille

Faculty of Technology, Natural Sciences and Maritime Sciences
Campus Porsgrunn

University of South-Eastern Norway

www.usn.no

Course: FMH606 Master's Thesis 2023
Title: *Surveillance system for drainage pumps with the use of machine learning*
Pages: *102*
Keywords: *Flood, IOT, LSTM, Transformers*

Student: *Vebjørn Rimstad Wille*
Supervisor: *Håkon Viumdal*
External partner: *David Bergene Holm, Håvard Omholt*

Summary:
Bergene Holm AS, Avd. Kvelde is a planer factory, surrounded by vast fields that are susceptible to flooding. This thesis looks into the possibilities of building upon the *Tokyo Amesh* system. The objective is to investigate the feasibility of regulating a storm water facility, or predicting the inflow into the storm water facility with the use of machine learning. To achieve this, an IoT device was built to log and monitor the storm water facility. The collected data was combined with local weather station data, processed, and exposed to a machine learning algorithm for training to predict inflow into the sump tank. Both Long Short-Term Memory networks and Transformer encoder networks were used to train on this time series data to be able to learn multidimensional correlations between the local weather station data and the logged water level of the sump tank within the storm water facility. Both Long Short-term memory and Transformers showed promising results in the prediction, but fell short of being used as either a forecast system or a control system, due to the limited data of the level of the sump tank. The lack of a longer logging time severely impacted the performance of the machine learning networks, as they were not able to achieve acceptable performance. However, due to the promising results, improving this system in the future is highly possible.

Preface

This master thesis was written as a part of the Industry Master program, Industrial IT and Automation. Therefore, this thesis was written for both the company Bergene Holm AS, and USN. The motivation for this thesis was to look into the possibility of improving a storm water facility at Bergene Holm AS, Avd Kvelde, that received a lot of attention after a spring flood caused severe problems for the factory. The idea was to log the process data and see if it was possible to use machine learning to further enhance the performance of the facility. The changing behavior of weather and how it affects local environments has always fascinated me, making this master thesis captivating. It should be noted that the reader should have some basic understanding of deep neural networks, as this thesis does not delve into the basics of this topic. Due to confidentiality issues, no code from the IoT device is shown, as it contains passwords and other sensitive information. Throughout this thesis, 'Kvelde factory' will be shortened to 'Kvelde'. Lastly, I would like to thank my supervisor *Håkon Viumdal* for all the help and guidance he has provided. I would also like to thank my external partners from Bergene Holm AS, *David Bergene Holm*, and *Håvard Omholt*, both for the time they have dedicated from their otherwise engaged schedules, to help me with insightful comments in this thesis. I would also thank *Inge Gjerden* for his involvement in the design phase of building the HMI for the IoT system. Finally, I would especially like to thank Bergene Holm AS for the opportunity to work for them and write this master thesis, as it has been a good experience and a pleasure.

Porsgrunn, 14th May 2023

Vebjørn Rimstad Wille

Contents

List of Figures

List of Tables

Nomenclature

Symbol	Explanation
IoT	Internet of Things.
HMI	Human Machine Interface.
UPS	uninterruptible power supply
HTML	Hypertext Mark-up Language
CSS	Cascading Style Sheets
CSV	Comma-separated values.
RMSE	Root Mean Square Error.
MAE	Mean Absolute Error.
MSE	Mean square error.
LSTM	Long Shot-Term Memory.

1 Introduction

Increasing urbanization and climate changes increase the amount of storm water, enhancing the danger of flooding, as vast volumes of water can accumulate over a short amount of time and flow uncontrollably [1]. Such a situation can cause significant damage to society and infrastructure, making it essential to handle storm water properly. One of the best ways to manage storm water is with natural waterways or, in many cases, man-made artificial channels that lead the water away. This is often handled in the pre-building phase or during spatial planning, and is one of NVE's plans to handle storm water [1]. However, sometimes this is not possible due to geological difficulties, environmental changes, or prohibitive costs. In such cases, it becomes necessary to use mechanical means to remove storm water with the help of storm water pump facilities.

Storm water pump facilities consist of a sump tank, often made of concrete or plastic, to accumulate storm water. They also include adequate pumps that meet the necessary drainage rate to drain the sump tank to a desired level. There are mainly two types of pumps used: submerged pumps or water-less aerial pumps. Both have their advantages and disadvantages. The most significant advantage of the submerged pumps is that they are more resistant to corrosion as they are isolated underwater, and they have natural cooling by the flow of water surrounding them. If the inflow of water inside the water sump is too demanding for one pump, or to reduce wear on the pumps by spreading the work between them, more pumps are often used in one facility. Finally, a control system is needed to control the pumps and the desired water level in the sump tank. Such a control system could be a frequency converter with level sensors that activate the pumps based on pre-fixed heights in the sump tank, in the form of *LOW*, and *HIGH* trigger boundaries, or more advanced control systems like PLC's [2].

However, as technology advances, it might be possible to increase the effectiveness of storm water drainage facilities. A country that has made significant improvements to its storm water drainage pump systems is Japan. As Japan is susceptible to more violent typhoons and increasing urbanization in big cities like Tokyo, they are becoming more prone to storm water flooding. Therefore, since the 1950s, Tokyo has worked on technological improvements to its storm water pump system. These improvements not only contain better technologies to increase the effectiveness of the pumps themselves but also the use of weather data to understand the incoming weather. This system is called *Tokyo Amesh* [3].

Tokyo Amesh is a weather observation system that aims to observe the incoming weather and try to give an insight into the current weather situation. It accomplishes this by the use of a meteorological radar that sends radio waves that bounce off individual raindrops and can notify the density of the rainfall, and what direction it is heading. This information is combined with 84 local rain gauges to create a color-coded heat map that indicates rain intensity. This heat map is then updated in real time to constantly inform about the changing weather situation. The map is then used by the operators at the storm water pump facilities to initiate the pump system based on the incoming weather. This can then reduce the "flash" flood peaks, as the pumps have the possibility to already be running as the storm water enters the pump facility, thus reducing the risk of storm water flooding [3].

Precipitation models are often used to calculate the possibilities for flood by combining precipitation with snow melting and other weather parameters. However, these models are often affected by wide-ranging uncertainty [4]. Machine learning is becoming increasingly popular within the meteorology field, as there is a lot of freely available weather data to train and build prediction models. It is thus possible to build machine learning prediction models to forecast weather [5].

This thesis aims to explore the possibility of using local weather data, combined with process data from a storm water facility at the factory Bergene Holm AS, Avd. Kvelde. This data will be used to build a machine learning algorithm that could serve as an observer for the storm water drainage system at Kvelde. Taking inspiration from the *Tokyo Amesh* system, the machine learning algorithm is used to increase or decrease the activation level for the water level inside the sump tank. The idea is that the model would predict the volume of water that will flow into the system ahead of time, and based on that prediction, "start" the pumps earlier to drain away the inflow faster than it can accumulate.

2 Situation At Kvelde

Bergene Holm AS, Avd. Kvelde is a planer factory located in Kvelde village, which is a part of Larvik municipality. This factory lies close to the river Numedalslågen and is surrounded by vast fields.

These fields slightly concave inwards towards the factory building, creating a low point next to the road that runs alongside the factory. The height differences can be seen in Figure 2.1, taken from Kartverket [6]. By inspecting the graph, it is possible to see that points A, C, and E all slope inwards towards points D and B, where point B is the absolute lowest point.

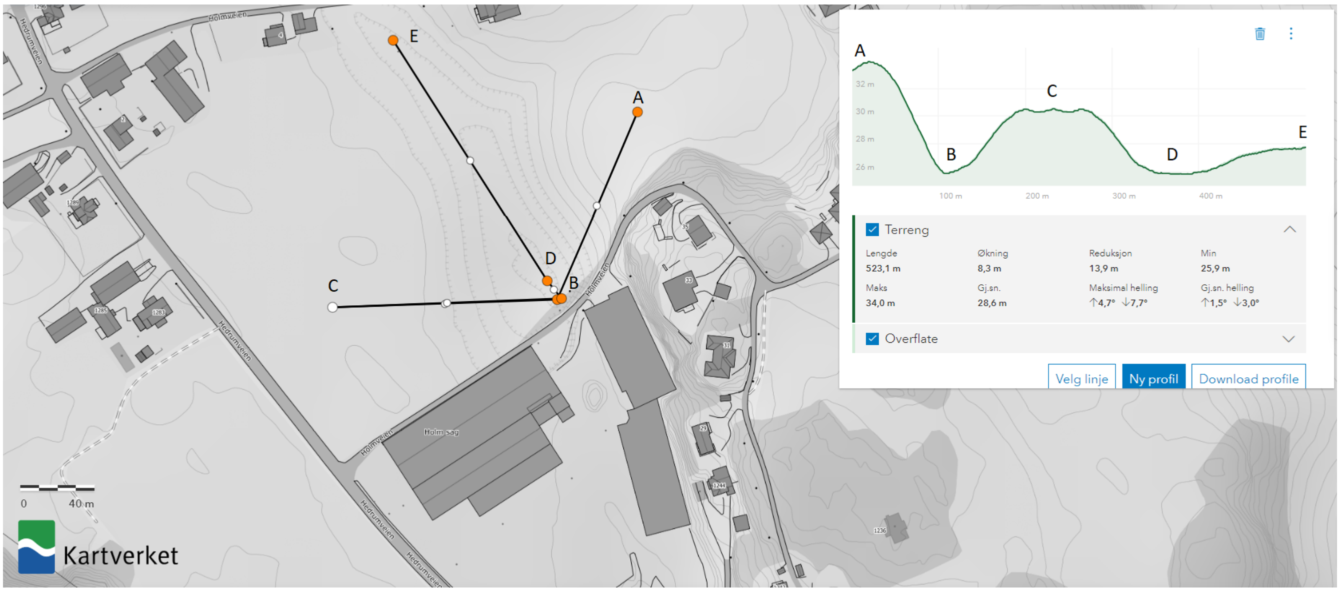


Figure 2.1: The topographic nature of the fields.

In wintertime, the cold seeps into the ground and freezes, creating ground frost on the fields. Then, rainfall, snowmelt, and other occurrences create a potential flood problem, as the storm water does not drain away into the ground, which is saturated with frost. This leads to accumulated water around the building at point B. If the water level gets too high, the production must halt, as the main power intake could be submerged, causing problems.

Today, this problem is fended off by a storm water facility installed at point $\boldsymbol{B}$. This storm water facility consists of two water pumps of the type *NP 3153 LT410*. More information about these pumps can be found in Appendix C.

These pumps are assembled at the bottom of the sump tank at point $\boldsymbol{B}$. The entire system is controlled by two *PS220* frequency converters, each controlling one of the pumps. The working principle of this system is that the *PS220* frequency converters monitor the level of the sump tank using a level sensor, and based on a fixed activation parameter, they start the pumps to drain the water down to a fixed deactivation level.

This system has a limited Human Machine Interface (HMI) on the unit itself for monitoring the situation and performance of the pumps. However, it lacks any logging capabilities to hold and display historical process data. The frequency converters do have internal registers that hold process parameters in real time, but do not store them. More on this will be discussed later

3 IoT System Design and Data Acquisition for Machine Learning

As mentioned in Chapter 2, the frequency converters did not provide any logging capabilities and had a limited HMI only. It was therefore decided that an Internet of Things (IoT) logging and HMI solution should be built to log the desired registers from the *PS220* converters, and to inspect the current operations of the pumps along with any alarms in the HMI. A system sketch of the IoT system can be seen in *Figure* 3.1.

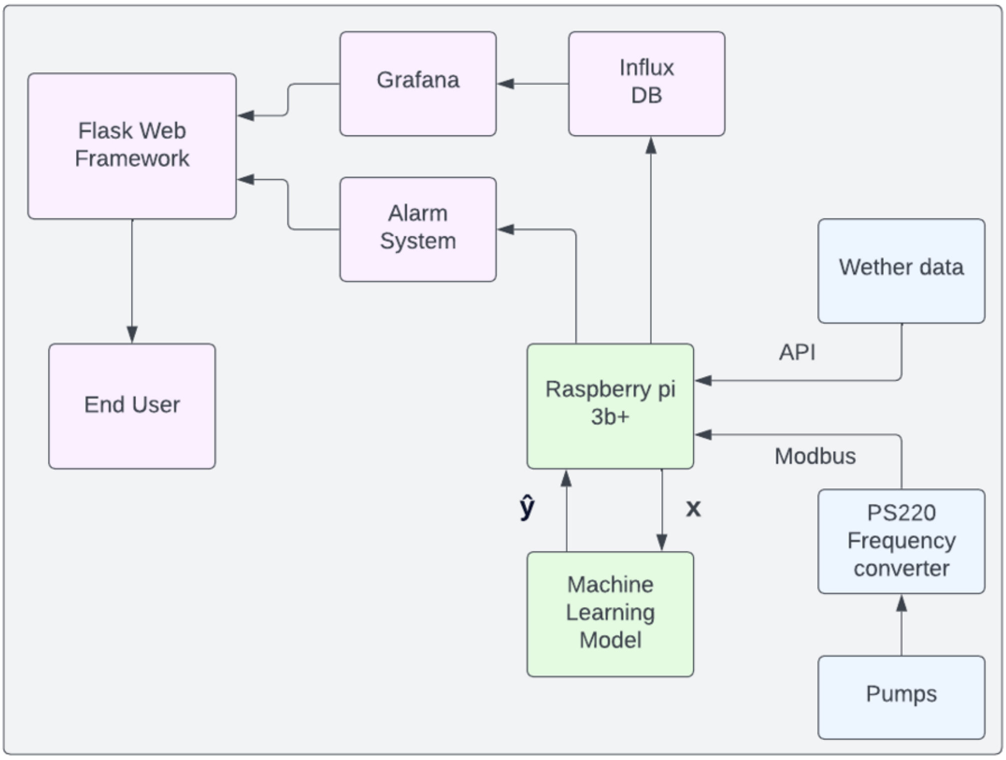


Figure 3.1: An overview of the system sketch.

The system sketch shows the main structure of the IoT system. Here, the pumps communicate with the *PS220* frequency converters and send that information via Modbus to the Raspberry Pi. The Raspberry Pi also retrieves weather data via an API communication to feed updated weather predictions into the database for use in a ”rolling” prediction of the inflow. This will be described in more detail later. The Raspberry Pi then sends all this data into the database, where it is stored and displayed to the end user via the Flask web framework.

The IoT system consists of mainly three parts: the Logger, the HMI, and the Machine Learning model. The logger's main objective is to retrieve the data from the desired registers from the *PS220* frequency converter. The data from these registers is then stored in a database to display on the HMI and for data analysis. The HMI is responsible for monitoring the current state of the pumps in real time. It retrieves the latest value from the register in real-time, to monitor the pumps, and historical data from the database to display long-term trends and performance. Lastly, the machine learning model will attempt to predict the inflow into the well, and then ”regulate” the level of the sump tank.

3.1 The Hardware

3.1.1 PS220 Frequency converter

The *PS220* Frequency converter that drives the pumps are, so-called *PumpSmart* frequency converters. These frequency converters are capable of creating intelligent pumping systems that have embedded algorithms designed for pump-specific problems. More information about them can be found in *Appendix* B.

These *PS220* Frequency converters have internal registers that contain information about the pump in real time, and some of these registers are interesting to log over time. A list of these registers can be seen in *Figure* 3.2.

Name	Register address	Description, group 52
Reference	40002	Reference or Setpoint used both for auto and speed override
Alarm word 1	40052	Data in 1 - Parameter 6.203
Alarm word 2	40053	Data in 2 - Parameter 6.204
Pump speed (rpm)	40054	Data in 3 - Parameter 1.208
Motor current	40055	Data in 4 - Parameter 1.07
Motor run time	40056	Data in 5 - Parameter 1.224
Start level (hysteresis)	40057	Data in 6 - Parameter 1.237
Stop level	40058	Data in 7 - Parameter 80.09
Energy counter, kWh	40059	Data in 8 - Parameter 1.20
Energy counter, MWh	40060	Data in 9 - Parameter 1.19
Digital inputs	40061	Data in 10 - Parameter 10.1
Smartflow	40062	Data in 11 - Parameter 1.217
Sump level	40063	Data in 12 - Parameter 1.205

Figure 3.2: A list of available registers from the PS200 frequency generator.

The specific registers to log are: Alarm word 1 and 2 to create an alarm list, Pump speed, Motor current, and Smartflow for pump run-time surveillance, Start level and Sump level for well integrity purposes, and finally Energy consumption and Motor run time for maintenance reasons.

3.1.2 Logger

To build such an IoT system, a Raspberry Pi 3B+ was used as the main computer. The reason for choosing a Raspberry Pi 3B+ was mainly that it was easily available. The main requirements for such a system are the capabilities to execute Python scripts, access to an RJ45 Ethernet port, and a relatively small size so it could fit inside an IT server cabinet, all of which the Raspberry Pi 3B+ has. The Raspberry Pi also has the advantage of having a good performance to cost ratio, as the Raspberry Pi 3B+ is quite affordable and still powerful enough to fulfill these types of tasks. More information about the specification can be found at RPI.com. An image of the specific Raspberry Pi board can be seen in *Figure* 3.3.

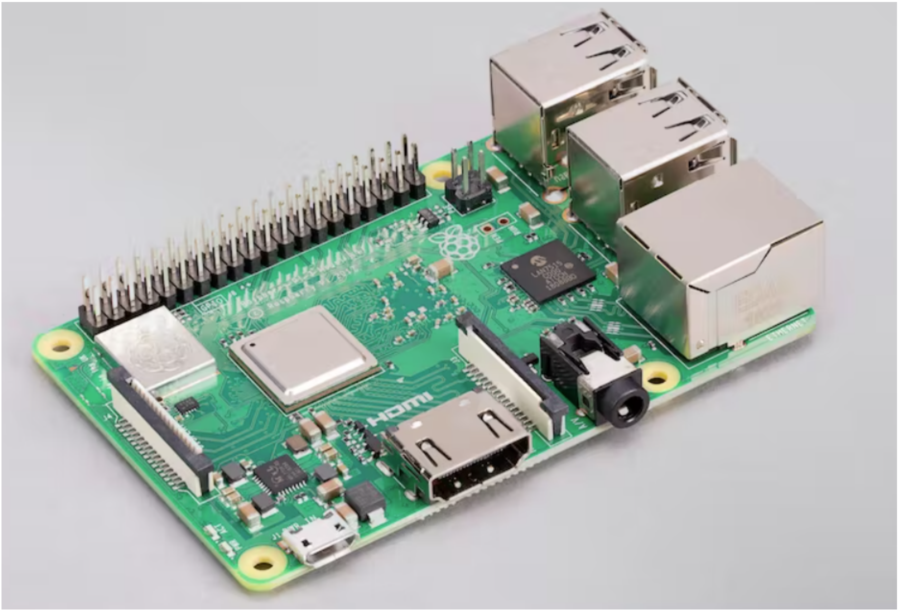


Figure 3.3: An image of the Raspberry Pi 3B+ taken from Raspberry Pi official website: RPI.com.

Since Kvelde Factory often experiences very short electrical blackouts, an Uninterruptible Power Supply (UPS) unit was installed as a Raspberry Pi HAT. This was done to circumvent the problem where the Raspberry Pi rebooted into a ”broken” state where the software did not redeploy correctly. An image of this UPS can be seen in *Figure* 3.4. More information about this UPS can be found at sixfab.com.

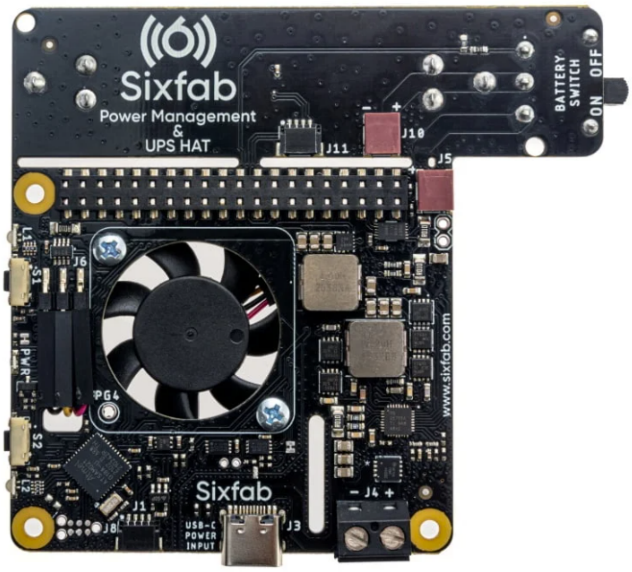


Figure 3.4: The UPS HAT connected to the Raspberry Pi unit.

3.2 The Software

The Python programming language was chosen for the task of building the IoT device. It was chosen as Python has a vast array of libraries, and it is well supported on the Raspberry Pi, making it the ideal language for such a task. To build the IoT device, a collection of applications were used together to make the system work. This is referred to as *the software stack*.

3.2.1 The Software Stack

The software stack shows the structure of the different applications running on the Raspberry Pi for the IoT system to work. An overview of the software stack of this particular system is shown in *Figure* 3.5.

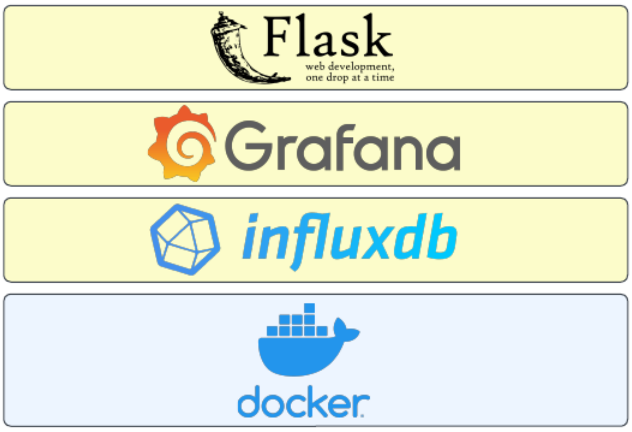


Figure 3.5: The Stack of the software architecture of the IoT system.

At the bottom of the stack lies Docker. Docker is a platform for building, shipping, and running applications in containers. Containers are a lightweight and portable way to package and isolate applications with their dependencies, so that they can run consistently across different environments. More about Docker can be found at Docker.com [7]. Docker was chosen to run the applications as it provided containerization, making the system flexible, and robust. Containerization also made it possible to go to Docker Hub, a platform for downloading pre-built containers, and download container images for both Grafana and InfluxDB. The advantage of using pre-built containers is that they are fully functional programs, and only need a few adjustments to run on this IoT system. Then by containerizing the Flask application, the IoT system would have three separate independent containers.

These containers could then be connected together and deployed at the same time on the Raspberry Pi with the help of Docker Compose. Docker Compose helps to configure and run all the containers based on a *YAML* requirement file [7], making it very easy to maintain and configure all the containers. Another benefit with Docker Compose is that if the Raspberry Pi were to reboot, it would handle the operations of automatically redeploying the containers, and "boot" up the programs again, and thus make the software stack more robust

Next in the stack is InfluxDB, a time-series database that allows for flexible and efficient storage of data without the need for predefined table structures or relationships. This makes it well-suited for time-series data, as it also provides tools to work with the real-time data inside the database. Tools such as the ability to down-sample data for long-term storage, continuous queries for real-time data analysis, and retention policies for managing data life-cycle. It also integrates with a variety of tools and services commonly used in the monitoring and analytic ecosystem, such as Grafana. More information about InfluxDB can be found at influxdata.com [8]. Thus making InfluxDB a good choice for the IoT system, as the type of data are all time-series. As it was also decided to use Grafana to visualize the database, it was a good fit.

Further up the stack is Grafana. Grafana is a platform for data visualization, monitoring, and analysis. It allows users to create, explore, and create dashboards based on data from various sources such as databases, cloud platforms, and monitoring systems. More about Grafana can be found at Grafana.com [9]. In the IoT system it retrieves the data from InfluxDB to be displayed in Grafana dashboards. This made it easy to display the data, as it had good integration with InfluxDB. The dashboards could then be used inside the HMI as HTML widgets, and thus eliminating the need for creating own graphs in the HMI which saved time. The Grafana graphs also had high functionality such as zooming and embedded time shifting included in these HTML widgets, making them ideal.

Lastly, on top of the stack lies the Flask application. Flask is an open-source web framework for building web applications in Python. It is a lightweight and flexible framework that provides a simple yet powerful set of tools and features for web development, and was therefore a good choice for this IoT system. More information about Flask can be found at flask.palletsprojects.com. As Flask is a web framework, it uses HTML and CSS to build and style web pages. This made it easy to build and design user specific HMI due to the flexibility of HTML and CSS, for then to be deployed it on the local network. The reason for choosing a HMI that was web-based, is that it did not require a specific operating system to access. As it can be accessed by all types of devices that support a web browser, therefore making it truly cross-platform. It also made the HMI accessible on multiple devices as long as they were connected to the same network as the Raspberry Pi, and therefore making every computer screen a HMI.

As Flask was the back-end of the HMI, it also handled the alarm system that was responsible for reading the alarm registers from the *PS220* frequency converters. It did this by multi-threading a Modbus communicator program to read the internal registers of the *PS220*. A list of these alarm words can be seen in the Quick guide in *Appendix* D.

3.3 The Human Machine Interface (HMI)

The HMI was built with user specifications in mind, the user wanted a ”clean” and easy-to-understand User Interface (UI). It was therefore decided to keep the information minimalist. This meant no use of any graphical animation in the design itself, and to hide unwanted information deeper in the HMI. In *Figure* 3.6 is the *home page* or the first page that meets the user. The main page holds two widgets, containing information about both *PS220* frequency converters. To the left of the *home page* is the navigation bar for traversal within the HMI.

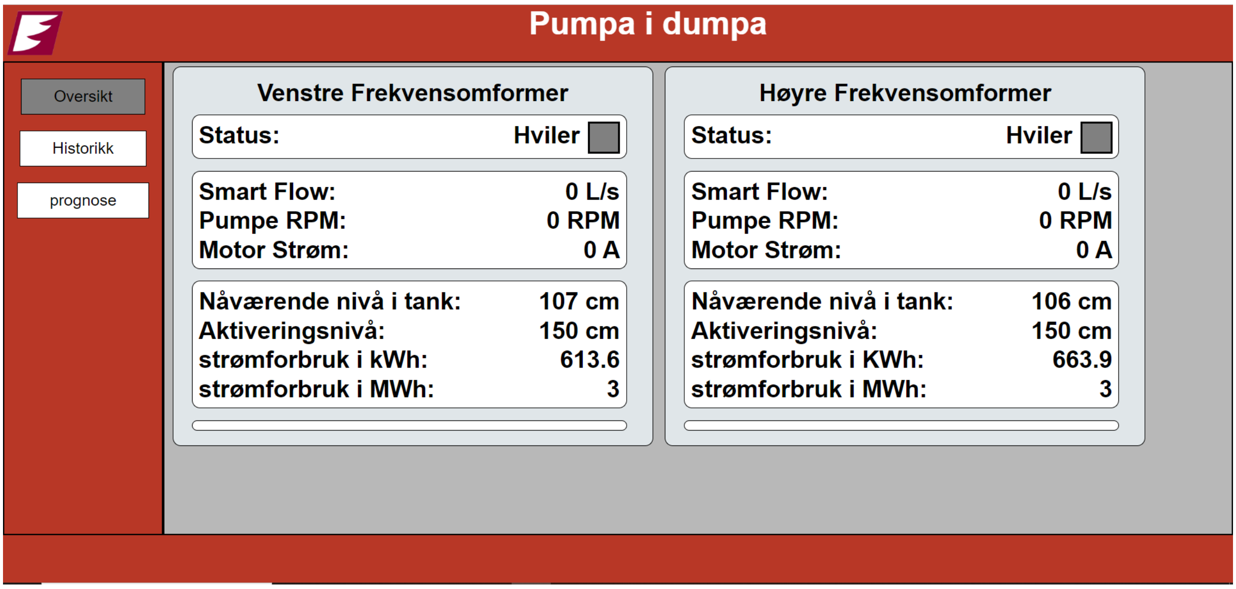


Figure 3.6: The front page of the HMI.

It is in these two widgets where the high-level overview of the pumps is displayed. Here is the status of the pump which shows what type of mode the pump is currently in. The pump can have 4 states that are; *resting*, *running*, *warning*, and *error* mode. These modes give the user an indication about the state of the pumps. An image of the pump running is shown in *Figure* 3.7

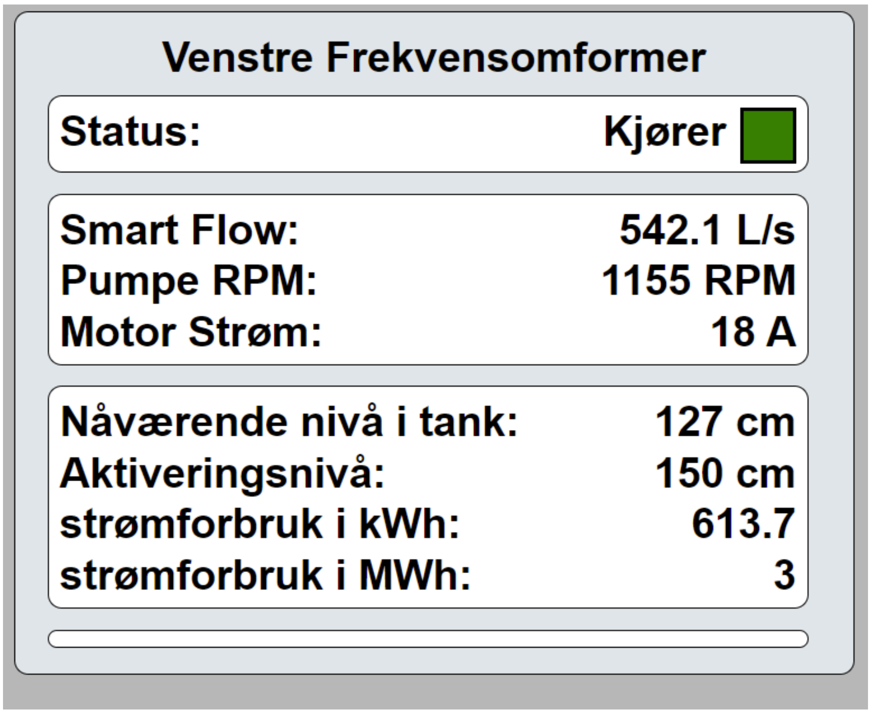


Figure 3.7: The HMI showing the Left Pumo running.

Onward are the run parameters such as *Pumpe RPM*, *Motor strøm*, and *Smart Flow*. These parameters show the user information about the performance of the pumps when it is running. Where *Smart Flow* shows an estimate of how much water the pump drains away in liters per second, *Pumpe RPM* shows the Revolution of the pump motor per minute, and lastly, the *Motor Strøm* shows the motor current usage. Combined, these parameters can give the user some information about the overall pump performance when running.

Further down is the information about the current level in the tank, visible by the parameter *Nåværende nivå i tanken*. Followed by the activation level, given by *Aktiveringsnivå*, together these show an expected range of the level within the tank for the user. For example, if the current level is above 150 cm, there is something wrong with the control system. Lastly is the total power consumption of the pumps in both kilowatt per hour (kWh) and in Megawatt per hour (MWh) that the pumps have used over time, displayed by the parameters *Strømforbruk i kWh*, and *Strømforbruk i MWh* respectively. This gives the user an overall information about run time and power consumption.

At the bottom of the sections is where the alarm list is contained. The HMI is designed in such a way that the alarm list is collapsed if there are no alarms to display. This was done to reduce distraction for the operator, as there was no point in showing an empty alarm list. All of the text elements of the parameters are clickable and will take the user to the *inspect* section of the HMI showing historical data about that specific element as shown in *Figure* 3.9.

If the user, however, wanted to see an overview of all the historical data of the HMI, they could navigate to the *Historikk* tab in the navigation bar, this leads to the Historic page. Here all the historical data from the parameters are shown in 6-hour resolution in the form of Grafana widgets. Here it is possible for the user to see comparing trends across the data elements, this might give an insight into the long-term performance of the pumps. *Figure* 3.8 shows the *Historikk* page open, and displays side by side information between the left and right *PS220* frequency converters.

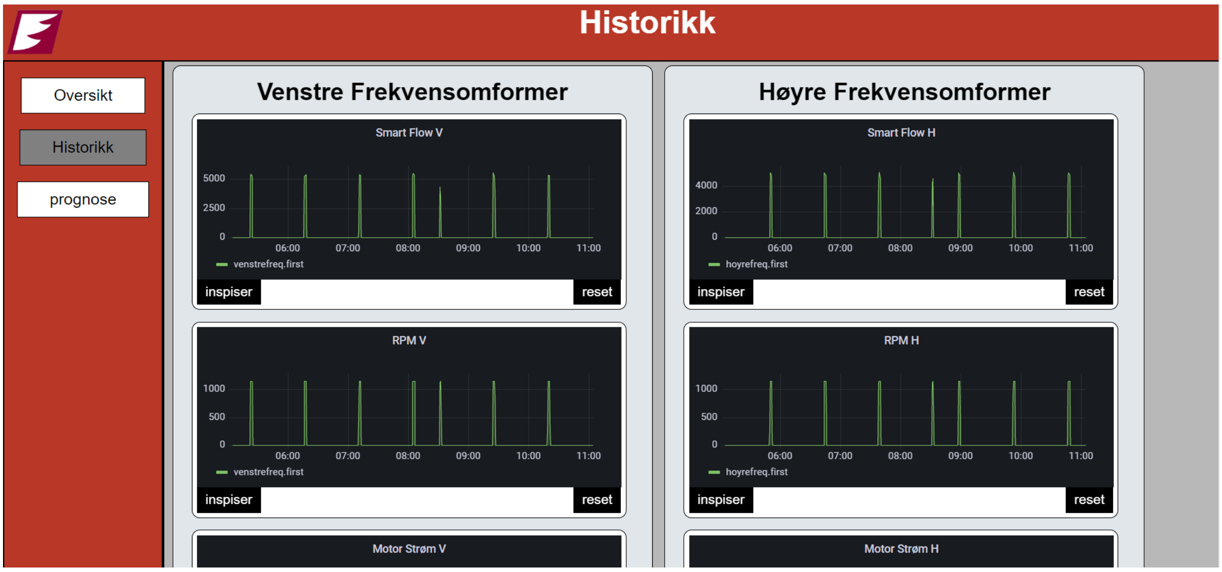


Figure 3.8: The history page of the HMI showing all the data elements.

If the user were to use the built-in zoom function from Grafana in these widgets, the *reset* button could be pressed to undo the zoom and reset the widget to a 6-hour resolution. If the user, however, wanted to see more than a 6-hour resolution, the *inspiser* button redirects the user to the inspect page of that particular parameter, as seen in *Figure* 3.9.

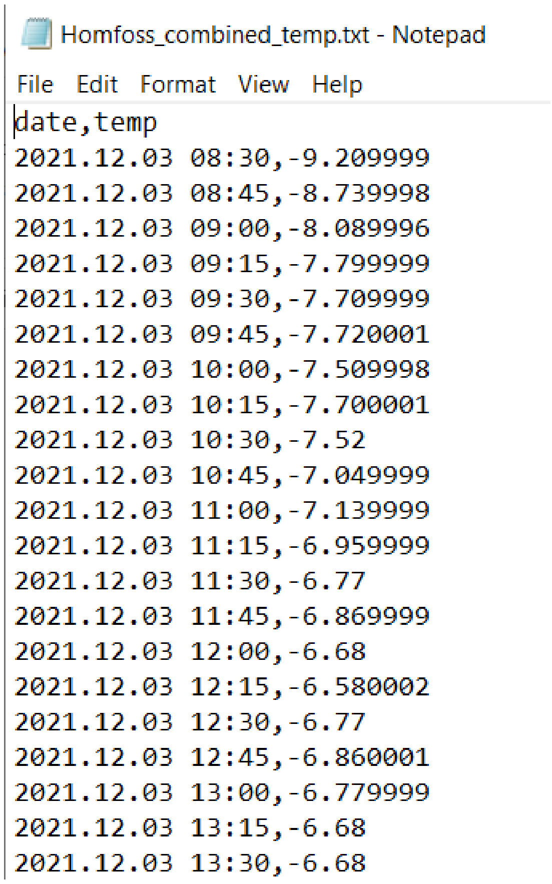
The inspect page of the HMI offers the possibilities to scale the time window of the data stored in Influxdb and displayed by Grafana. When on the inspect page for a specific parameter, the user can use the time bar to scale the data to 1 day, 1 week, 1 month, or 1 year. This gives the user the ability to see how the data stored has changed over a longer period of time. There is, however, some caution needed when using the 1-year time scale of the Grafana widget. Since querying one year of data is extremely computer demanding, it might crash the Influxdb instance if there are too many data points to retrieve. *Figure* 3.9 shows the inspect page of the logged parameter *Nåværnede nivå i tank*.

Figure 3.9: The inspect page of the HMI showing the historical data in detail.

Lastly on the navigation bar is the *prognose* page which is the page where all the weather prediction should be. Here the user would have the opportunity to view the prediction that the machine learning algorithm would produce. This, however, was cut due to time constraints.

3.4 Data gathering

To be able to perform machine learning, relevant data is required for the algorithm to learn. It was therefore decided to collect data from local weather stations near Kvelde factory, combined with the historical data of the level of the sump tank collected from the IoT system. These local weather stations were Hedrum and Holmfoss weather stations, both stations had available data for download as comma-separated values (CSV) format via SeNorge.no [10]. Hedrum weather station provided historical data of precipitation and snowmelt each day whilst Holmfoss provided historical data of temperature and the height of Numedalslågen. An example showing the temperature CSV file can be seen in *Figure* 3.10.



Homfoss_combined_temp.txt - Notepad

File Edit Format View Help

```
date,temp
2021.12.03 08:30,-9.209999
2021.12.03 08:45,-8.739998
2021.12.03 09:00,-8.089996
2021.12.03 09:15,-7.799999
2021.12.03 09:30,-7.709999
2021.12.03 09:45,-7.720001
2021.12.03 10:00,-7.509998
2021.12.03 10:15,-7.700001
2021.12.03 10:30,-7.52
2021.12.03 10:45,-7.049999
2021.12.03 11:00,-7.139999
2021.12.03 11:15,-6.959999
2021.12.03 11:30,-6.77
2021.12.03 11:45,-6.869999
2021.12.03 12:00,-6.68
2021.12.03 12:15,-6.580002
2021.12.03 12:30,-6.77
2021.12.03 12:45,-6.860001
2021.12.03 13:00,-6.779999
2021.12.03 13:15,-6.68
2021.12.03 13:30,-6.68
```

Figure 3.10: The CSV file containing temperature data from SeNorge.no.

These data were chosen based on the idea that they might have the most impact on the water level in the sump tank. Temperature affects the ground, as mentioned, with ground frost and will have an effect on how much water that goes into the sump tank, rather than in the ground. The amount of rain that has fallen will have an effect on how much storm water is accumulated on the ground, and therefore affect the level in the sump tank, the same goes for snow melting as melting snow accumulates water. Lastly, the height of Numedalslågen might have some correlation with the level of the sump tank as they are both susceptible to the same natural impacts.

3.4.1 Data pre-processing

The CSV files were loaded into Python via the Pandas library. As Pandas is an open-source data analysis and manipulation library for Python. It provides powerful tools and data structures for working with time-series data. Such tools including data cleaning, data scaling, transformation, and visualization to name a few.

Pandas revolves around two main data structures: Series and DataFrame. A Series is a one-dimensional labeled array that can hold any data type, while a DataFrame is a two-dimensional labeled data structure with columns of potentially different data types. DataFrames are similar to tables in a relational database or an Excel spreadsheet, and provide a powerful way to manipulate and analyze data. More about Pandas can be found at pandas.pydata.org [11].

To use the data to train a machine learning algorithm, it was important that all the data had the same time resolution. A problem, however, occurred when inspecting the different data as they indeed had different time resolution. The level of the sump tank was logged every other second, whilst the temperature and the level of Numedalslågen were logged every 15 minutes, and lastly both snow melt and precipitation only had one reading each 24 hours.

This problem was solved by choosing a common time resolution for all the data sets, and then "scale" the data accordingly, it was therefore decided to select 1 hour as a common range. The level of the sump tank, temperature, and the level of Numedalslågen were all averaged out with the intervals of 1 hour, whilst both snow and precipitation were linearly interpolated with a 1 hour interval. Thus creating a common time range of 1 hour for all the data-sets.

For a flood to occur, the inflow would need to be larger than the outflow of the system. Therefore, *currentLvl* was inadequate to use for prediction as it does not say much about inflow by itself. An image of the level of the sump tank can be seen in *Figure* 3.11. It was therefore decided to look at the gradient of *currentLvl* as this would represent the inflow per hour into the sump tank. The transformation was done by finding the slope of the *currentLvl*, and removing the large spikes generated by the pump drainage. It was important to distinguish between natural drainage into the ground, and large drainage spikes from the pumps. This was important since the model would need to learn the natural drainage into the ground. Then the gradient of *currentLvl* was multiplied by the volume of the sump tank to get the inflow in Liter Per Second (L/h), this new dataset was named $inflow$. The transformation from *currentLvl* to $inflow$ can be seen in *Figure* 3.12.

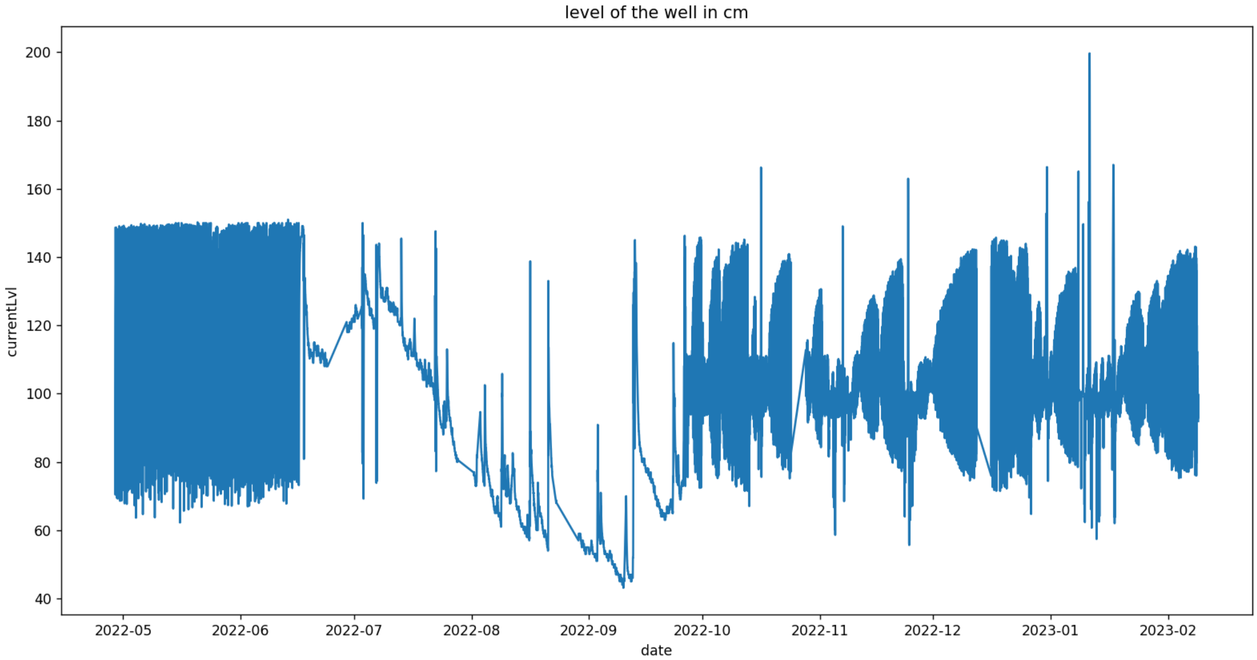


Figure 3.11: The level of the sump tank.

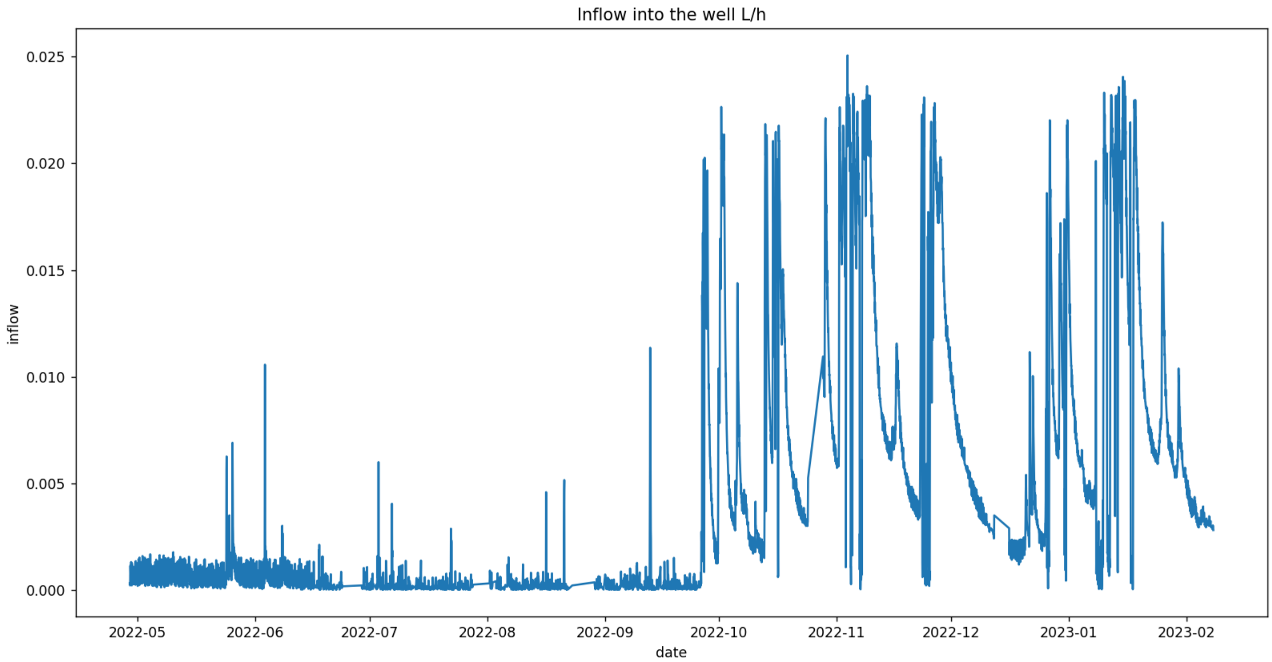


Figure 3.12: The transformation from level to inflow into the sump tank each hour.

After all the data were properly constructed and finalized for machine learning training, the data were split into test and training sets. However, unlike some machine learning methods, this splitting was not randomized, meaning that the ”cut” for the test and training set was a fixed date that separated the test from the training set, *Figure* 3.13 shows the ”cut” between train and test set. It was important that the test set consisted of newer samples than the training set, as that would represent the future prediction of a rolling prediction system.

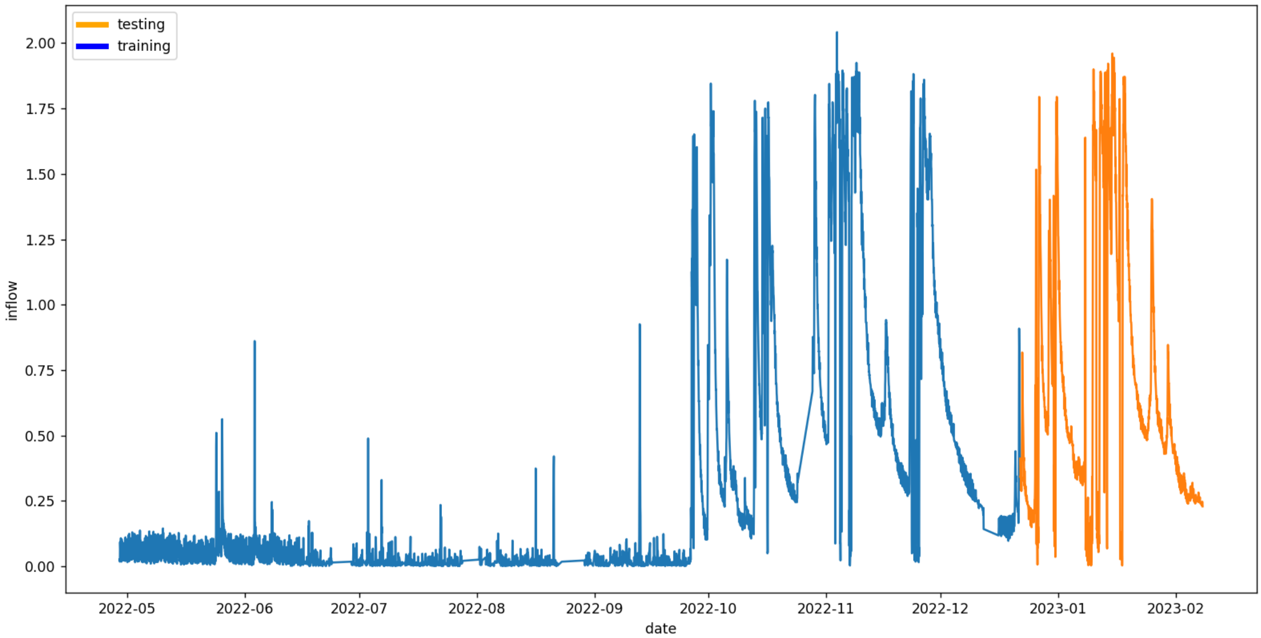


Figure 3.13: The split between training data and test data.

Then to frame the data into a supervised learning set, the sliding window method was used. The sliding window method is a technique to restructure the time-series data into a supervised learning set for the machine learning algorithm [12]. This was done by segmenting both the training and test data into two sets, the X and Y datasets. Where X holds the *n_past* values, or the number of samples to use to predict the next Y value, including the previous Y value, while Y holds all the values to predict. This gave the X dataset the shape of (a, b, c), and Y the shape $(a, 1)$. Where a is the number of matrices for X and the number of Y values in the Y dataset, and they need to be the same length as each matrix is used to predict the next Y value. b is the number of past samples to use, and c is the number of variables to use to predict the next Y value. An example of a slice from the sliding window method can be seen in *Table* 3.1. This slice gives an insight into how the method works, as it needs, in this case, seven data points from all six datasets to predict one value of the next $inflow$.

temp	rain	currentLvl	snow_melt	river_level	inflow (x)	
14.5775	2.6786	77.3166	0.0	1.7085	0.006357	
14.6500	2.8328	77.2814	0.0	1.7120	0.006368	
14.715	2.9869	77.2462	0.0	1.7122	0.006380	inflow (y)
14.7600	3.1411	77.2111	0.0	1.7127	0.006391	0.006438
14.8325	3.2953	77.1759	0.0	1.7097	0.006403	
14.9575	3.4494	77.1407	0.0	1.7125	0.006415	
14.9775	3.6036	77.1055	0.0	1.7090	0.006426	

Table 3.1: An example of one slice of the sliding window.

Then these values were scaled. This was done to ensure that all the features were on a comparable scale. If the features are on different scales, some features may have a larger impact on the model than others, even if they are less important. This can lead to inaccurate or biased predictions. By scaling the data, it is possible to ensure that all features contribute equally to the model and that the model is not influenced by the arbitrary scaling of the data [13].

3.5 Time-series predicting using RNN

Recurrent Neural Networks are a category of Neural Networks that are known for their ability to have ”memory”, meaning that they use prior information when evaluating the next output [14]. This makes them quite useful in sequential time-series prediction as all data points in a sequential dataset are dependent on the prior data in the series [15].

RNNs, however, have two major flaws when using back-propagation through time: they are vanishing and exploding gradients. These are phenomena that occur when RNNs deal with long-range dependencies. The gradient either becomes too small, thus updating the weight of the network until the weights are so small that the network does not learn anything, or, in the case of exploding gradients, the weights get so large that they represent NaNs [14]. To combat this problem, Long Short-Term Memory (LSTM) was proposed. LSTM was designed to combat the problems back-propagation through time caused by implementing a gradient-based learning algorithm [16].

3.5.1 Long Short-Term Memory (LSTM)

The LSTM network consists of LSTM cells, each containing three main components: the input gate, the forget gate, and the output gate. The input gate ”protects” the content inside the memory cell by determining whether the memory cell should be updated with the current data. The output gate ”protects” other LSTM cells from being affected by the output of the last LSTM cell by deciding if the current output should be visible to the next cell or not [16].

The forget gate is a key component of the LSTM architecture, providing the ability to reset the memory cell. Interestingly, the forget gate was not explicitly mentioned in the original paper "Long Short-Term Memory" [16]. Instead, the paper describes a mechanism called "constant error carousel" that serves a similar purpose to the forget gate. The forget gate was introduced in the paper "Learning to Forget: Continual Prediction with LSTM" [17]. An image of the LSTM cell, taken from thorirmar.com [18], can be seen in *Figure* 3.14.

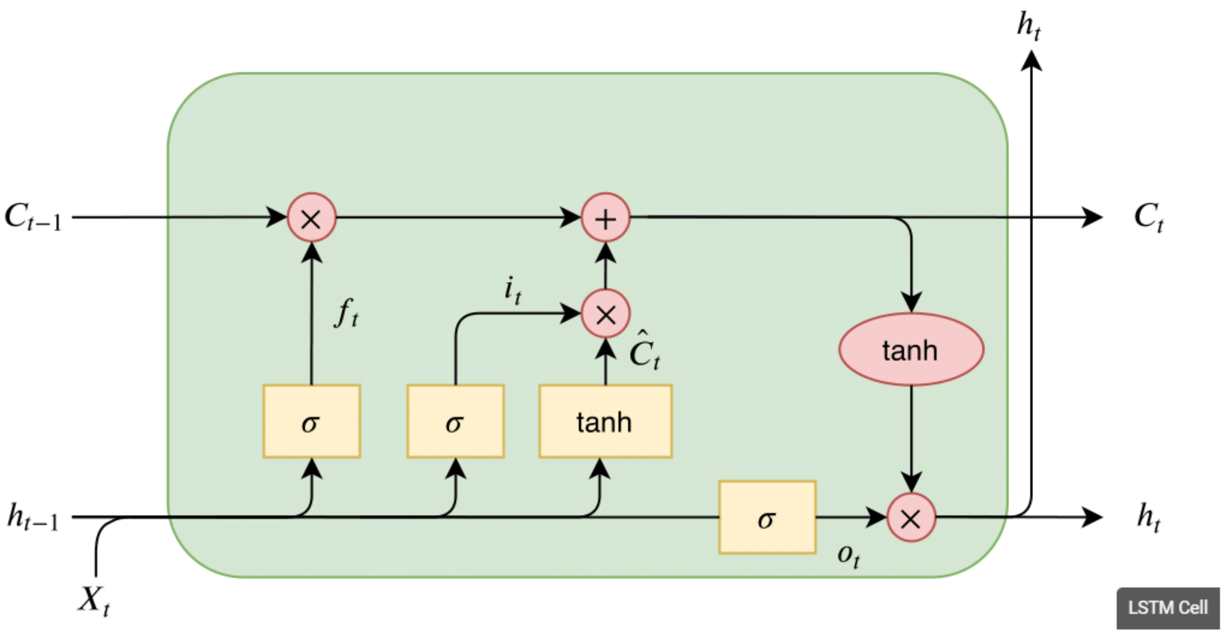


Figure 3.14: An example of the LSTM cell taken from thorirmar.com [18].

The LSTM network in this project consisted of LSTM layers with N number of cells in each layer, with the Rectified Linear Unit (ReLU) activation function followed by dropout layers. The dropout layers help prevent overfitting the model by randomly dropping neurons throughout the layer [19]. The ReLU function was chosen as the activation function as it has some properties that work quite well for LSTM networks. ReLU is computationally cheaper, making training the model faster, and most importantly, due to the activation function's sparsity-inducing properties, there are no gradient vanishing effects [20].

Finally, the last layer was a Dense layer with 1 neuron to match the shape of the *train_y* dataset, ensuring that the output of the model would be of the shape $(a, 1)$. Then the LSTM network was compiled using the "adam" optimizer with the Mean Square Error

(MSE) as the loss function, before being fitted with the training set and tested with the test set.

3.6 Time-series predicting using Transformers

Transformers is a machine learning model that was originally built to translate languages. This model, unlike LSTM, does not use recurrence in its architecture and relies solely on the attention mechanism [21]. The attention mechanism allows the algorithm to focus on specific parts of the data [22]. An image taken from the paper ”Attention Is All You Need” [21] can be seen in *Figure* 3.15, showcasing the structure of the Transformers model architecture.

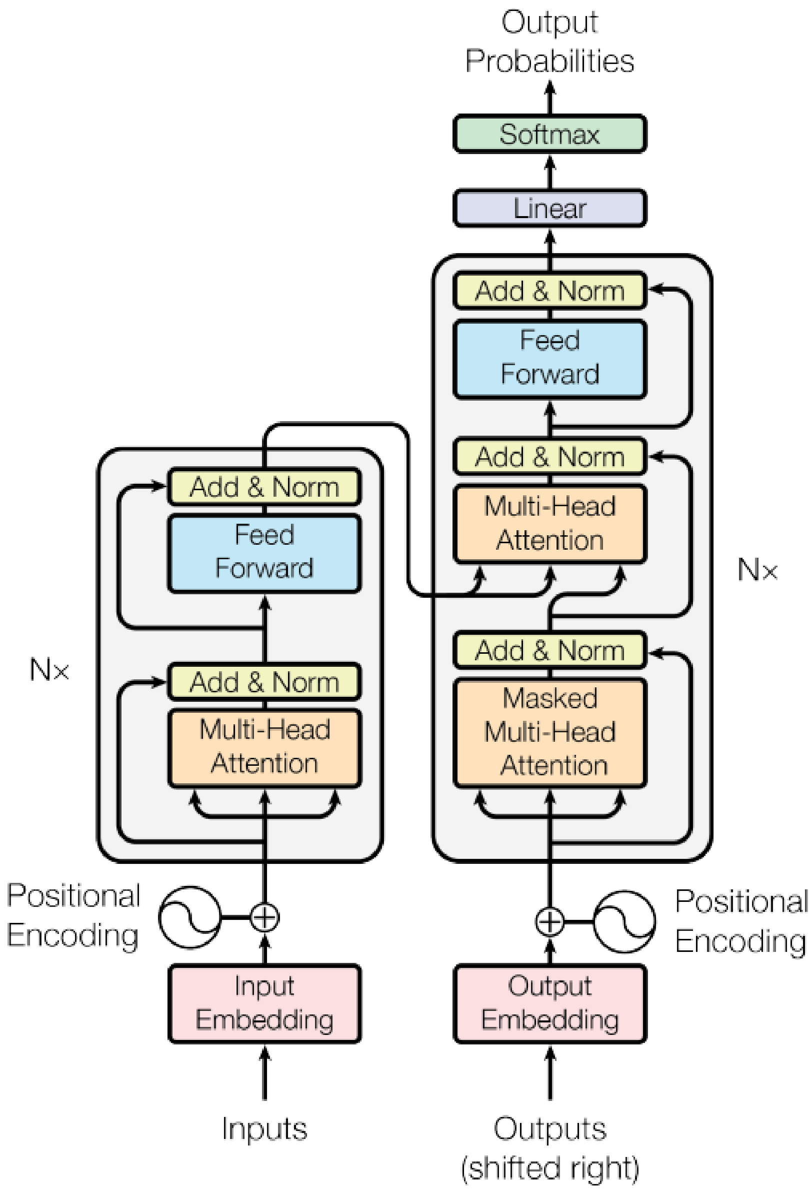


Figure 3.15: The model architecture of the transformer model.

Figure 3.15 shows the Transformers model structure, which consists of two parts: the encoder and the decoder. The encoder is visible on the left and the decoder on the right. The decoder of the Transformers is usually used for sequence-to-sequence tasks such as machine translation or text summarizing [21].

However, when it comes to time-series prediction, the goal is to predict the next value in the sequence based on a fixed number of previous values. In this case, the output sequence has a fixed length of one, and there is no need for a decoder to generate the output sequence. Instead, the encoder is used to extract relevant features from the input sequence, which can then be used to make the prediction. Then the output of the encoder is passed through one or more fully connected layers to generate the final prediction. The Encoder itself consists of the Multi-Head attention layer followed by the Feed Forward layer. More information about Transformers can be found in the paper ”Attention Is All You Need” [21].

The structure of the encoder in this project started with normalization and attention layers. This layer consisted of a normalization layer to normalize the input data, followed by a Multi-Head attention layer with n numbers of attention heads, and an m size for the query and keys for the attention heads. Then a dropout layer was introduced for the same reasons as in the LSTM network. The next part of the encoder was the feed-forward layer, which also consists of a normalization layer, followed by a 1-dimensional convolution layer. The use of a 1-dimensional convolution layer in the feed-forward network provides parameter efficiency and computational speed, as well as better generalization performance compared to fully-connected layers [21].

3.7 Tuning and validating the machine learning

Tuning a Machine Learning model is no trivial task as there are a lot of parameters that directly affect the performance of the model, therefore a systematic approach was used to train the hyperparameters of the model.

The epoch parameter of a Machine Learning model tells the model how many iterations through the datasets it takes in one learning phase [23]. The task of tuning the epoch for a Machine Learning model can be both tedious and challenging, therefore an early stopping mechanism was introduced in the training phase for the models. This was done since, if the model is under-trained it has a tendency to underfit, while the opposite is true for an over-trained model [24]. The early stopping mechanism would therefore stop the training at a certain point when the performance of the validation starts to decline [24]. This is a good compromise to solve the challenging problem of how long a model should train.

The rest of the model parameters were tuned systematically by tuning one of the parameters at a time, and inspecting the Mean square error (MSE) for both the loss and the validation loss functions for each training epoch, to then change parameters accordingly. Where the goal is to get as low MSE score as possible. For example, increasing dropout layers or increase complexity by adding layers if the MSE of the loss function is too high and see if it reduces. Thus indicating that the model shows tendencies to underfit.

Then the Root Mean Square Error (RMSE), and the Mean Absolute Error (MAE) were inspected to give an indication of how well the prediction of the test set with the tuned model performed. The difference between RMSE and MAE is how they penalize error. RMSE penalizes large errors, and is therefore useful when it is important to emphasize that large errors are unwanted [25]. MAE, on the other hand, looks at the magnitude of the error, and does not consider the direction of the error [25].

Lastly, due to the stochastic nature of machine learning models [26], both the LSTM and transformer models were tested 30 times, and the results were plotted in box plots. This was done to show the uncertainty or spread of the performance for both models.

4 Data Exploration

A concern surrounding the data analysis was that the data would be quite poor. As the IoT Logger had only logged data from the level of the sump tank in the time range from 28.04.2022 to 08.02.2023, making the time range short. The problem with this is that weather-based data is heavily affected by seasonality, since weather changes based on the season. Due to the short logging time, it meant that the data would not represent a full year, and would therefore not contain these trends that seasons create. This will be a problem for the machine learning algorithm, as it won't be able to learn these trends.

4.1 Results of the Data Analysis

Figure 4.1 shows the scatter matrix between all the datasets. Here it is possible to view the trends and correlations between two datasets. The diagonal line shows the histogram for each dataset, and by looking at the different histograms, it can reveal something about the spread of the data. The histograms for *rain*, *snow_melt*, and $inflow$ are all quite tail-heavy, while the others have a more bell-shaped histogram. The tail-heavy histograms might pose a problem for the machine learning algorithm, as it might be harder for it to find the underlying patterns.

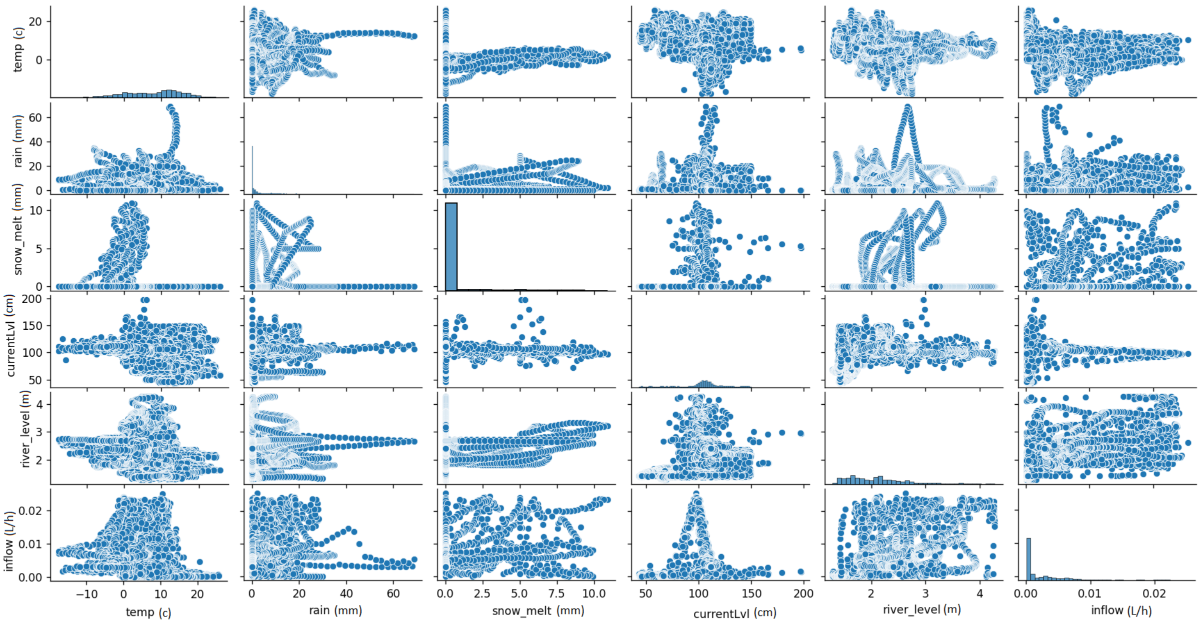


Figure 4.1: The Scatter matrix of the data showcasing the interaction between datasets.

By looking at the scatter matrix, it was not easy to discern any clear linear relationships, nor any non-linear relationships. This, however, was expected, as the scatter matrix only shows relationships between two datasets. The inflow into the sump tank is likely a combination of non-linear relationships between multiple datasets. For example, the inflow would be affected by both temperature and rainfall at the same time. This property would not be visible in such a scatter matrix.

As it was not easy to identify any linear correlation from the scatter matrix, a heat map was generated. The heat map in *Figure* 4.2, shows the linear correlation between the different datasets. On the left side is a heat rod with a scale that indicates how strongly the data is positively or negatively correlated. The reason for the diagonal being all 1 is because there will always be a perfect positive linear correlation between the data and itself. By looking at the column at the far left, it is possible to see all the linear correlations between other datasets and the ***inflow***. Here it is clear to see that ***currentLvl*** has vastly less correlation with the other datasets compared to the ***inflow*** dataset, thus indicating that it was necessary and advantageous to do the transformation mentioned in chapter 3.4.1

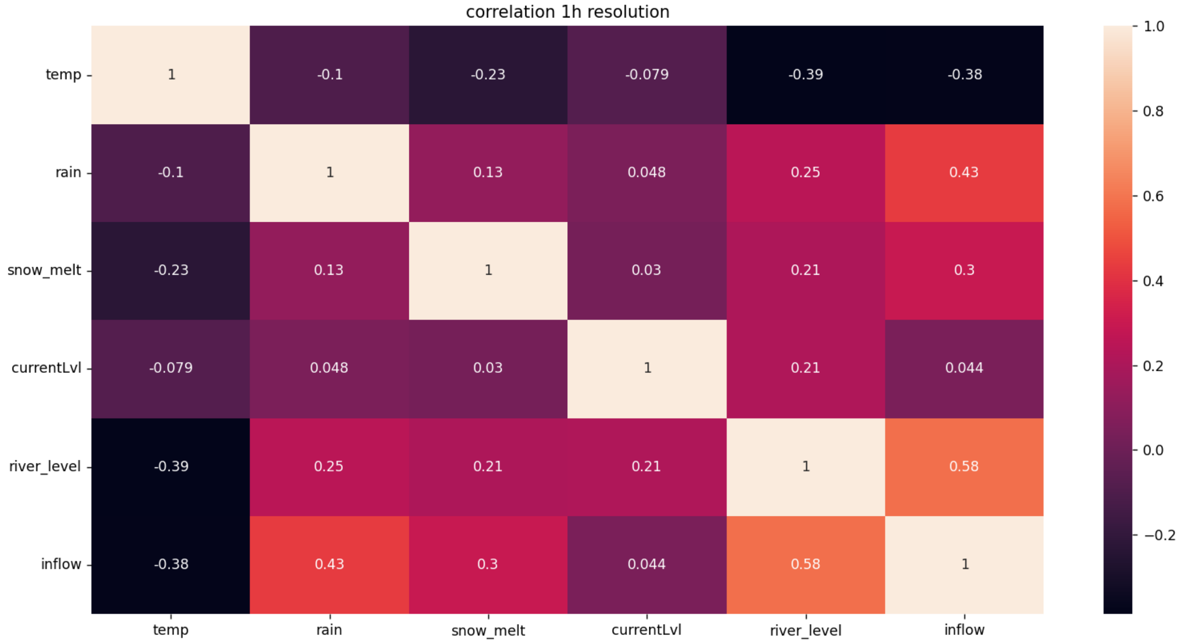


Figure 4.2: The correlation heat-map.

The correlation between *temp* and *inflow* is weakly negatively correlated with a value of -0.38. This correlation can be explained by the nature of ground frost that was discussed earlier. The lower the temperature, the more likely there is ground frost. That will impede the flow of water from draining into the ground, and thus more water will flow into the sump tank, increasing the L/h. However, it might also be due to a lack of representative data, as discussed, leading to an overrepresentation of negative temperature throughout the dataset. With more representative data, there might be less overall correlation.

Both the *snow_melt* and *rain* correlations with *inflow* are weakly positively correlated with values of 0.3 and 0.43, respectively. This might be due to the fact that in the summertime, the fields drain the water into the ground. The fields would therefore work as vast water magazines that prevent water from flowing into the sump tank. In the winter, the rain is "stored" as snow on top of the fields, and therefore there is less "rain" that flows into the well. The *snow_melt* would only affect the *inflow* in the latter part of winter and early spring, when snow turns into storm water. Since the data were retrieved at the end of April, there is a vast underrepresentation of *snow_melt* in the dataset since the captured data "misses" the snow melting season at Kvelde.

When it comes to the correlation between *river_level* and *inflow*, it has a moderate positive correlation with a value of 0.58. This makes intuitive sense, as the water level of Numedalslågen is also susceptible to all the same weather impacts as the storm water facility at Kvelde. However, Numedalslågen is a power regulated river, and is therefore susceptible to stagnation or increased flow rate based on human interaction. This might lead to changes in the water level of the river, regardless of snow or rainfall. This is likely the reason for the rather poor correlation between both ***rain*** and ***snow_melt*** with ***river_level***.

Overall, there are only small to moderate linear relations between ***inflow*** and the other datasets, and there aren't any clear non-linear relationships seen in the Scatter matrix in *Figure* 4.1. The Scatter matrix only looks at the relationship between two datasets, and therefore it is harder to see the underlying non-linear relationships that appear between combinations of multiple datasets. Since the inflow into the sump tank should be predicted based on all the datasets at each time step, these underlying non-linear relationships are the ones that affect the prediction. It is therefore advantageous to use deep learning neural networks, as they are capable of learning the intricate patterns and capturing these underlying non-linear relationships. As neural networks have multiple hidden layers, and take data points from each of the datasets as the input layer, they have the opportunity to "look" at the whole dataset, and capture the non-linear patterns that are not so apparent at first glance.

5 Predicting using Machine Learning

As mentioned in chapter 4, the lack of seasonality will affect the performance of the machine learning algorithm. When inspecting *Figure* 3.12, in *Chapter* 4, there is a major increase in inflow that happens around the month of September. As the seasons change towards Autumn and Winter, there is a change in weather that further increases the inflow.

This, however, is bad news for the machine learning algorithm, as there will be an overrepresentation in the data where ”almost” nothing happens to the inflow into the sump tank. Due to the fields absorbing the storm water, less water drains into the sump tank between the start of May and the end of September.

If the machine learning model were to train on this data, it could lead to a biased model that is trained on too much data that has less influence on the inflow, and therefore create a ”lazy” model. An example of the LSTM model being exposed to this biased data can be seen in *Figure* 5.2. Here the prediction is ”cutting” off all the reduction in inflow and has a stable low point that is unwanted. This stable low point is likely due to the overrepresented data that fluctuates around 0 from May to September. It was therefore decided to cut the dataset at 28.09.2022 and the new dataset interval can be seen in *Figure* 5.1.

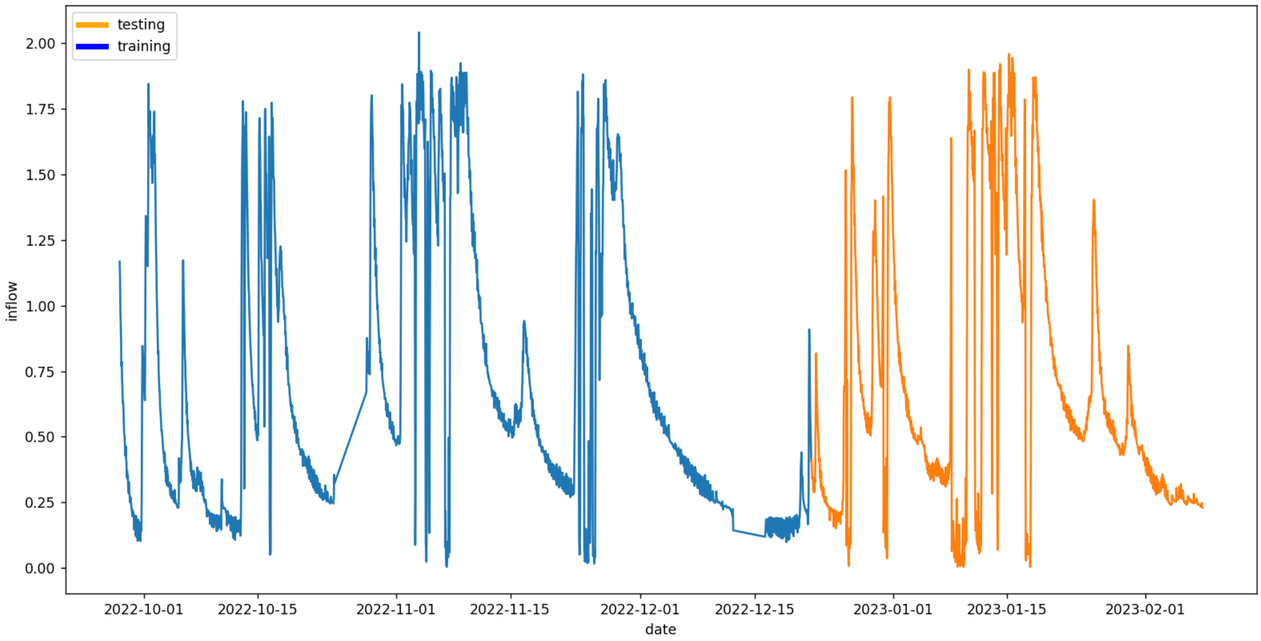


Figure 5.1: The new shortened training and test set.

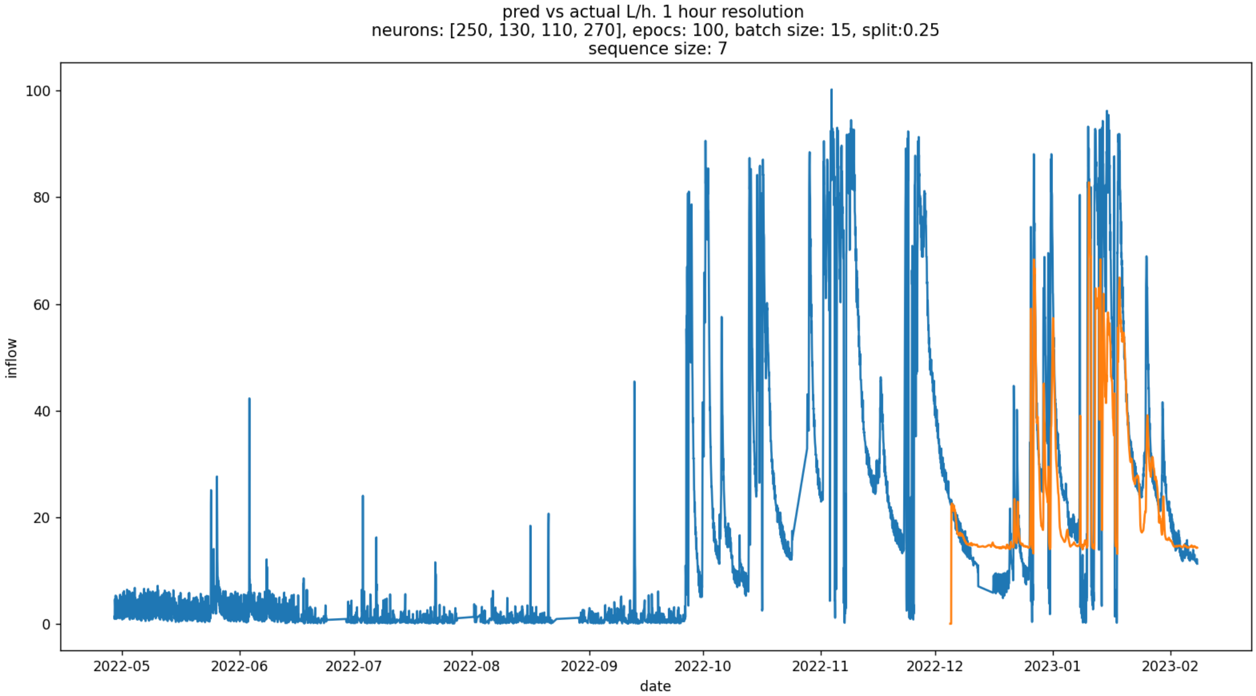


Figure 5.2: LSTM model trained on all the data-sets and under-preforming.

The new dataset would therefore represent the Autumn and Winter season, where the inflow into the sump tank would be most pressing. Thus, it would be more representative of the problem of trying to predict floods. It is, however, important to note that this machine learning algorithm will not be able to predict anything beyond the seasons that are captured by the dataset. Since no year has equal weather behavior, this model will most likely perform poorly on next year's data of the same season, as the algorithm won't be able to capture the seasonality difference that occurs between different years of the same season, due to the lack of data.

Note that at the start of the prediction results plot, represented by the yellow line, there will always be a sharp spike up from zero before the prediction starts, visible in *Figure* 5.2. This is an artifact from the use of the sliding window method.

The artifact occurs as the sliding window offsets the dates, at the first prediction, by the number of ***sequences*** used in predicting the next Y value. To prevent this offset, the array holding the predicted values was shifted $n - \boldsymbol{sequences}$ forward in the array and the beginning was filled with zeroes. This will only affect the plotting of the graph for visual representation only, and not the RMSE nor the MAE of the models.

5.1 LSTM tuning and results

5.1.1 Tuning the LSTM model

After tuning the hyperparameters of the LSTM model with the parameters shown in *Table* 5.1, the following result shown in *Figure* 5.3, were produced. Here the LSTM network model had 3 hidden layers with 50, 70, and 10 neurons in each layer respectively, with dropout layers of 0.2 between each of these layers, seen by the ***dropout*** parameter. The model also used 7 past samples in the sliding window method to predict each inflow value, seen by the ***n_past*** hyper-parameter. The model had a maximum of 100 epochs to train, meaning if the early dropout function did not kick in, it could only train a total of 100 epochs. Lastly, the split ratio between the test and validation set from the training set was split with a 25% ratio. The *Figure* 5.3 shows the model's training performance, by plotting the Mean Square Error (MSE) for the validation set and test set over the training epoch. The y-axis represents the MSE and the x-axis is the amount of epochs. By inspecting this plot, it is possible to see that the early stopping mechanism stopped the training around 30 Epochs. The model is also underfitting since both the MSE value for both validation loss and the loss function "flattens" out around 0.3 and 0.2 respectively.

Number of neurons	n_future	n_past	epochs	n_batchh	split_ratio	droput
50, 70, 10	1	7	100	100	0.25	0.2, 0.2, 0.2

Table 5.1: The hyperparameters from tunig phase.

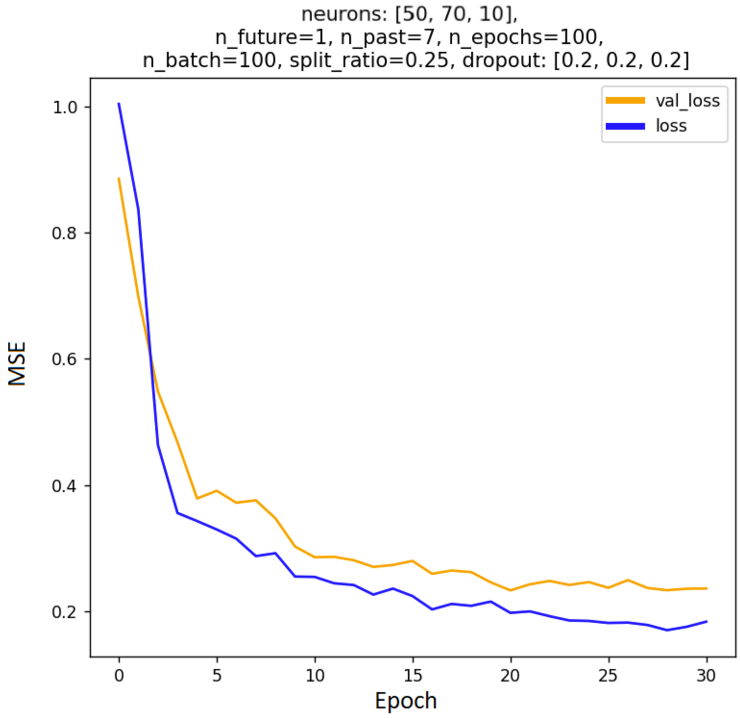


Figure 5.3: The LSTM model underfitting.

To mitigate the model from underfitting, the complexity of the model was increased to attempt to make the model capture the underlying patterns. This was done by increasing the number of hidden layers from 3 to 4, increasing the number of neurons in the hidden layers, and reducing the number of batches that go through the model before updating the weights. The updated parameters can be seen in *Table* 5.2, and the new result of this tuning can be seen in *Figure* 5.4.

Number of neurons	n_future	n_past	epochs	n_batchh	split_ratio	droput
150, 170, 110, 130	1	7	100	50	0.25	0,.2 0.2, 0.2 ,0.2

Table 5.2: The updated parameters for the LSTM model.

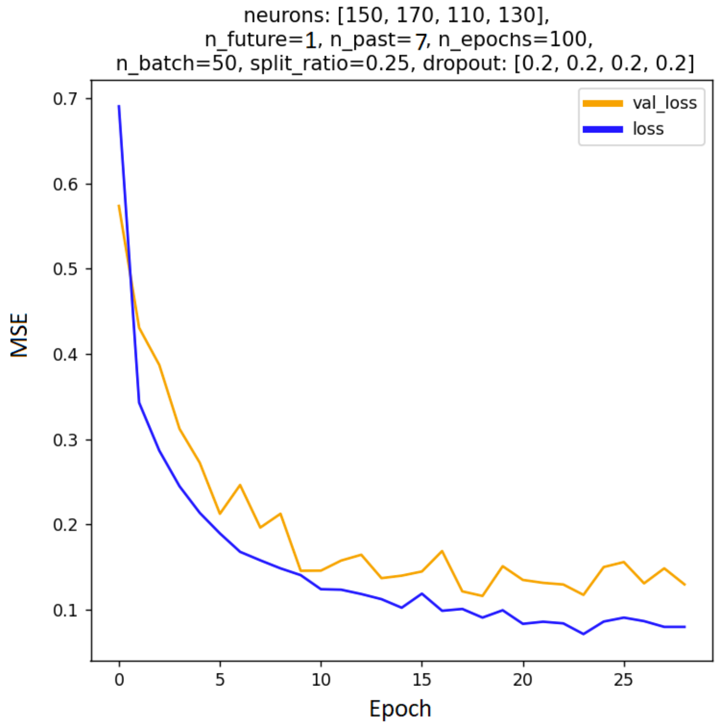


Figure 5.4: The LSTM model underfitting less.

Here, it is possible to see that increasing the complexity helped with the underfitting, as the MSE of the loss function flattens out around 0.1. However, it introduced some noise in the validation loss function. This is likely due to the dropout layers introducing noise as they drop random neurons. The dropout layer was removed to attempt to reduce its impact on the validation loss. This indeed led to a less noisy result as seen in *Figure* 5.5. It was therefore decided to move forward with this model, as further testing resulted in worse results.

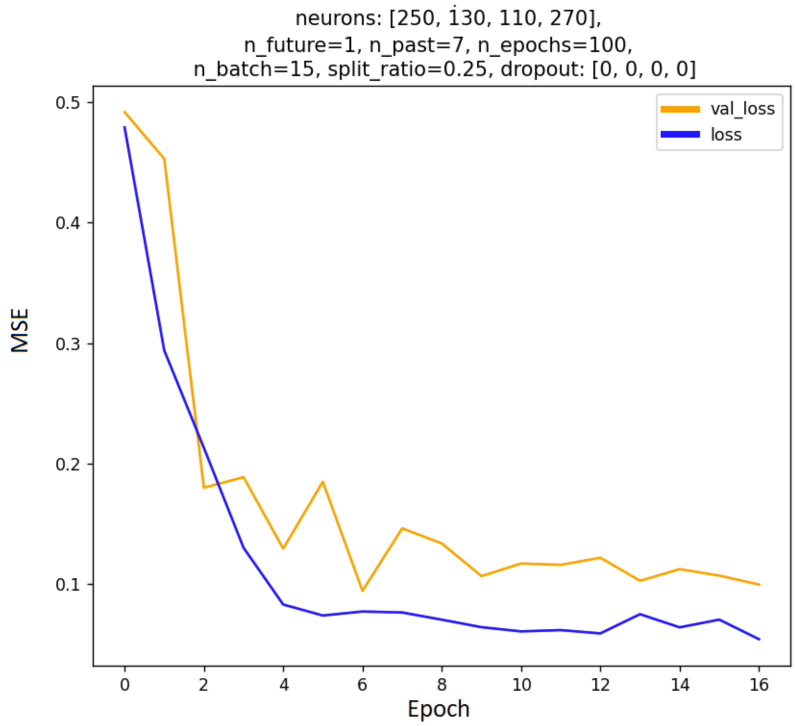


Figure 5.5: The LSTM model with ok fit.

5.1.2 LSTM model performance

Figure 5.6 shows the result of both the training and predicting performance of the tuned LSTM network. The left plot is the same training result as seen in *Figure* 5.5, shown above, and to the right is the result of the prediction of this particular model.

Besides the prediction result suffering from some ”bleeding artifacts”, seen by the yellow prediction line having large spikes in the negative direction at some predictions, the overall performance of the LSTM looks okay, as the prediction line does follow the actual inflow value, represented by the blue line.

It is, however, important to note that the predictions never reached the top of the peaks, and by zooming in on the end of the prediction, shown in *Figure* 5.7, there is an offset. This offset likely comes from the model using the last inflow input as the best current prediction, and therefore all the predictions are shifted to the right creating the offset. There is also a lot of noise from the model trying to fit the rapid changes that the inflow data generated.

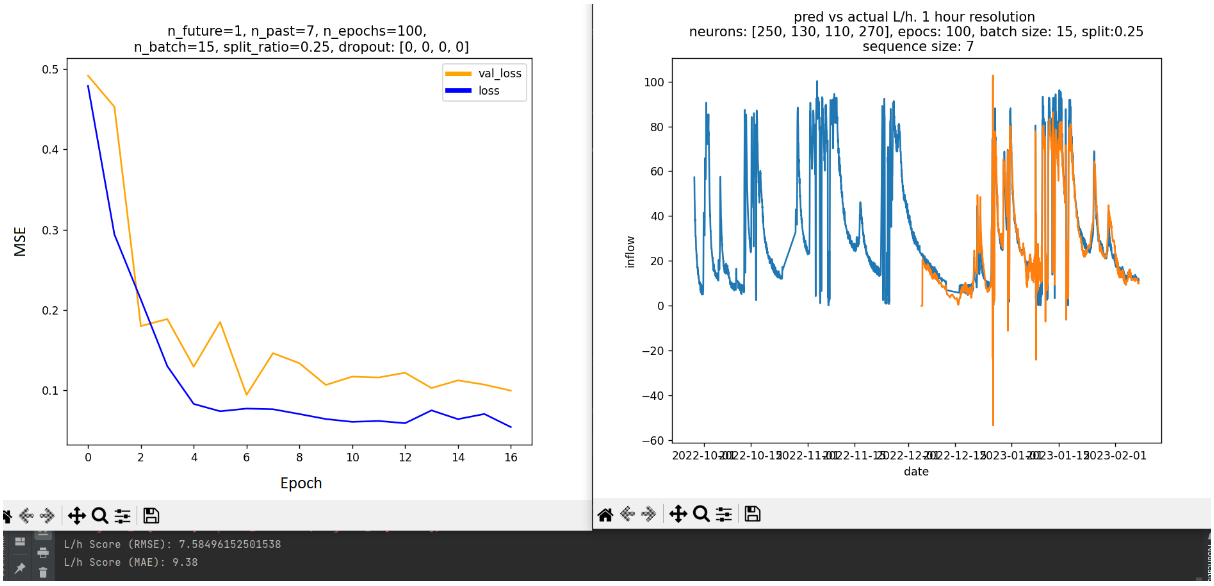


Figure 5.6: The full overview of training and testing the LSTM model.

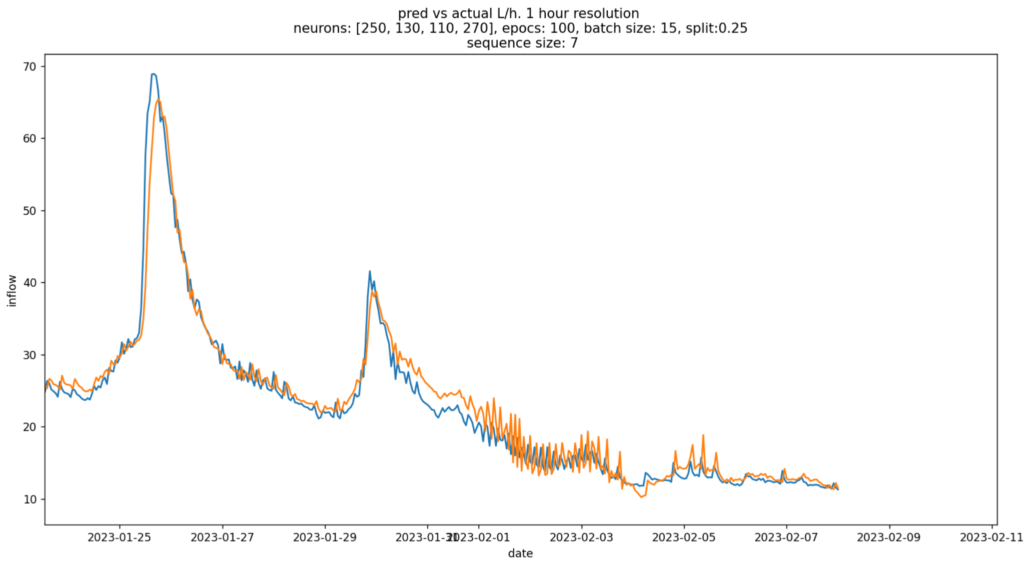


Figure 5.7: The prediction vs actual curve zoomed in at the end.

5.2 Transformers tuning and result

5.2.1 Tuning

The tuning of the transformer was performed in much the same way as the LSTM, the only difference was that the Transformer network had different hyperparameters to tune. The hyperparameters that were tuned for the first transformer can be seen in *Table* 5.3.

num_blocks	num_head	head_size	batch_size	epochs
4	4	156	100	200
n_future	n_past	split_ratio	dropout	mlp_dropout
1	7	0.2	0.1	0.2

Table 5.3: The hyper-parameter list for the first Transformer encoder performance.

Here the ***num_blocks*** represents the number of encoder blocks the transformer network consisted of, in this case 4 identical encoder blocks stacked on top of each other. Each encoder block consisted of 4 parallel attention mechanism heads in the multi-head attention layer, represented by the ***num_head***. Each of the attention mechanisms had a dimension of 156, given by the ***head_size*** parameter. The rest of the parameters were the same as for the LSTM, with some other values, except the ***mlp_dropout*** which was the dropout layer for the Multi-Layer Perception.

These parameters led to the first result seen in *Figure* 5.8. By looking at the MSE of the loss vs validation loss functions, this model is also underfitting like the LSTM model. However, to improve the model's performance, more data were given in the training phase. As mentioned at the start of *Chapter* 5, giving the LSTM model more training data hurts the performance, this however, was not the case for the transformers. This is likely due to the Transformers Attention mechanism being able to preform relatively well, regardless of being exposed to bias data.

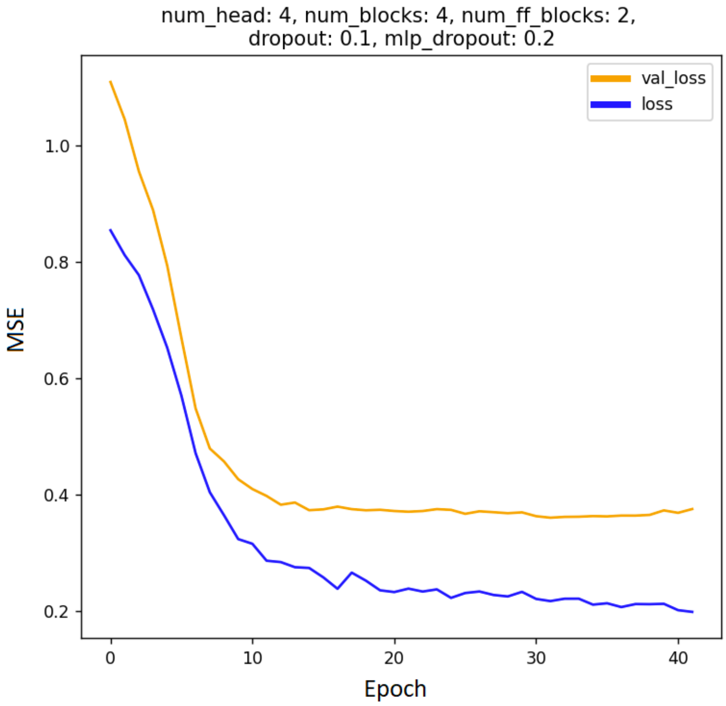


Figure 5.8: The transformer underfitting.

By increasing the time window from 28.09.2022 to 01.08.2022 and retraining the Transformers, the underfitting decreased. Then, by tuning the dropout layers along with the number of heads, the following parameters were obtained, shown in *Table* 5.4. The result of these tuning steps further increased the performance, as seen in *Figure* 5.9. Here, both the validation loss and the loss functions flatten out at acceptable levels. The loss function has reduced significantly compared to the underfitting seen in *Figure* 5.8 above. For more results of the tuning phase, see *Appendix* E.

num_blocks	num_head	head_size	batch_size	epochs
4	6	156	100	200
n_future	n_past	split_ratio	dropout	mlp_dropout
1	7	0.21	0.2	0.1

Table 5.4: The hyper-parameter list for the second Transformer encoder performance.

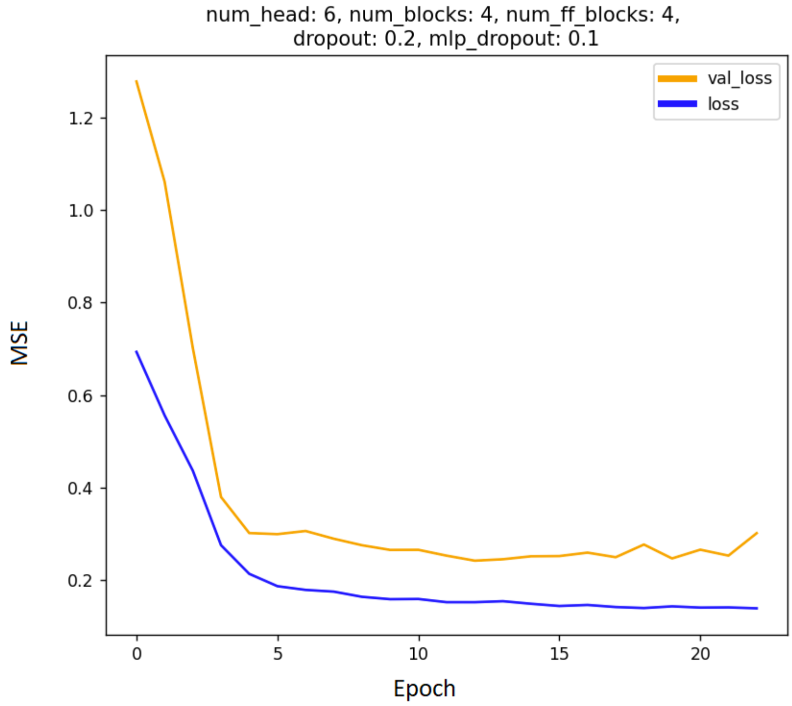


Figure 5.9: The Transformer preforming well.

5.2.2 Transformers performance result

Figure 5.10 shows both the training and prediction performance from the last trained model, same as for the LSTM. By looking at the prediction plot, the model does a good job of following the blue inflow curves but has some overshooting on the highest peaks. The overshoot can be seen by the yellow prediction line having a higher level around the largest spikes. By zooming in on the latter section of the prediction result, as seen in *Figure* 5.11, the prediction curve has the same type of offset as in the LSTM, but with a smoother transition curve that does not attempt as hard to fit the rapid changes from the inflow data.

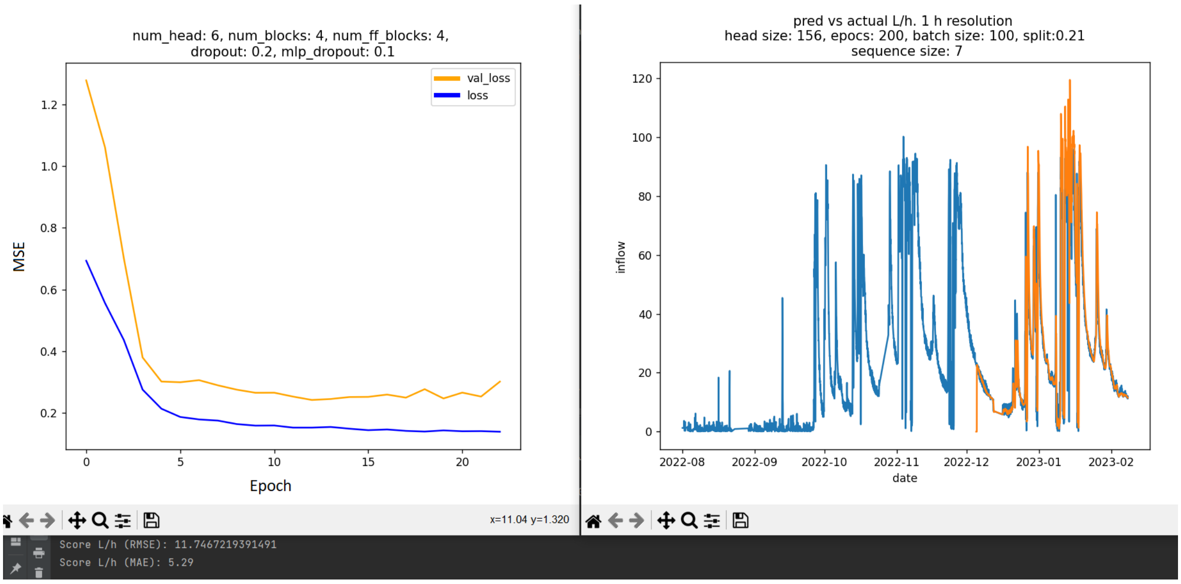


Figure 5.10: The overview performance of the training and testing of the Transformer.

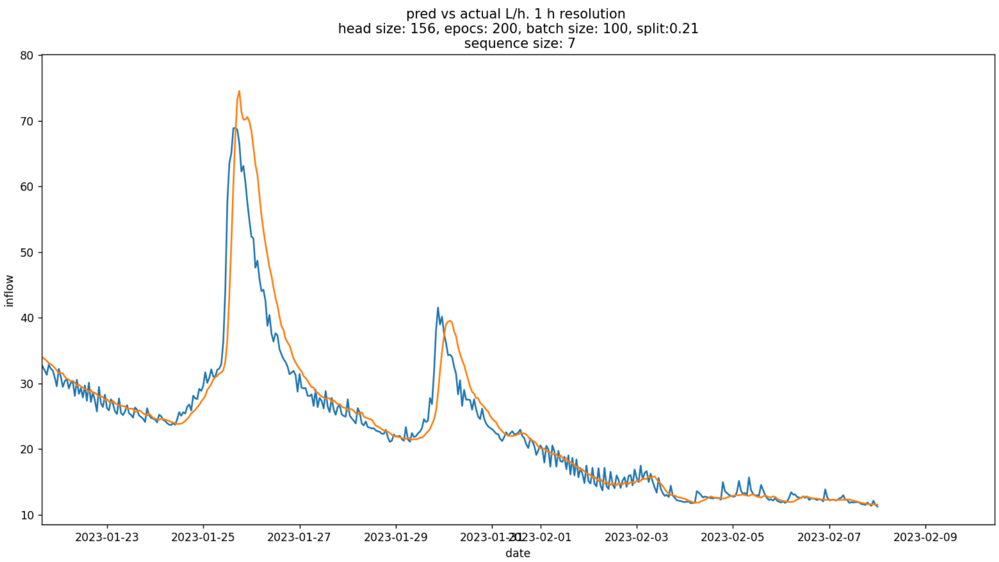


Figure 5.11: The zoomed end of the prediction.

5.3 Comparison between LSTM and Transformers

The LSTM and Transformer behaved differently on the prediction, where the LSTM had problems reaching the tops of the peaks and therefore underestimated higher inflow peaks. The Transformers did not attempt, as hard as the LSTM, to follow the rapid changes from the inflow, and therefore underestimated smaller inflow changes. This is especially present in the difference between RMSE and MAE of the two models, as seen in the box plot in *Figure* 5.12. The box plot shows the uncertainty in the 30 iterations for both the LSTM and the Transformers. The boxes represent the spread of the data with the black line in the middle showing the median, and the whiskers the maximum and minimum value. Lastly, the diamonds represent "outliers", performance of the models that are too poor, and therefore categorized as outliers.

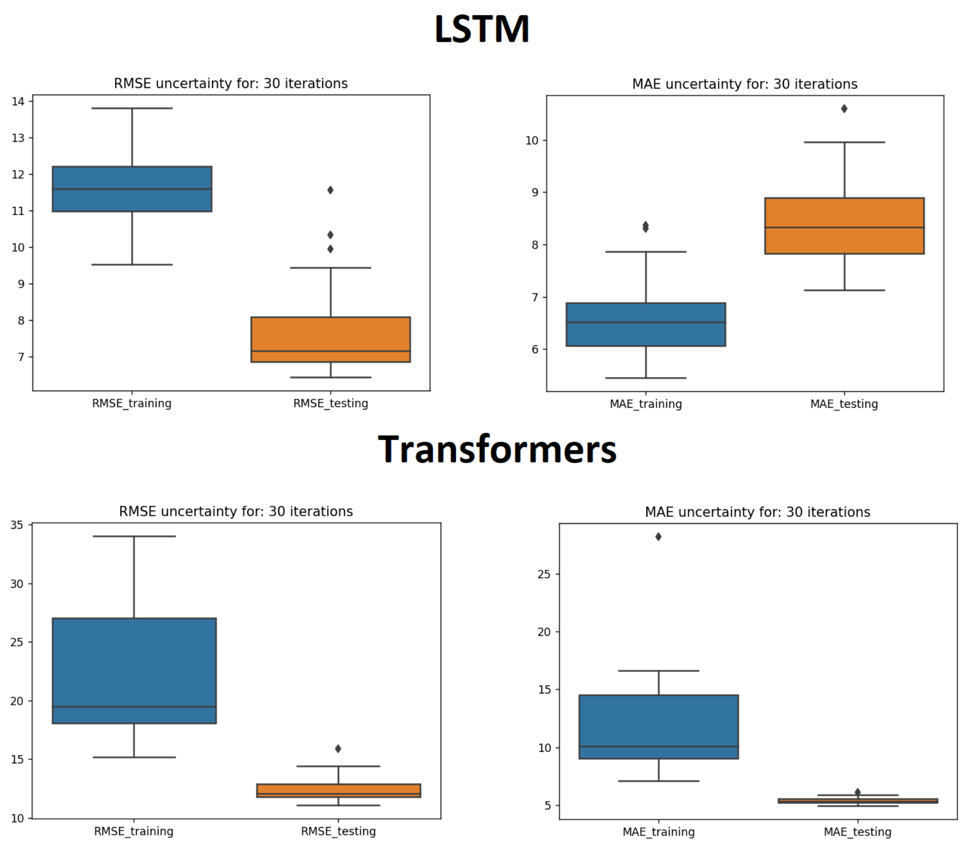


Figure 5.12: The uncertainty of 30 LSTM and Transformers iterations.

Here, the LSTM has an overall lower RMSE with 11.5 L/h median for training and 7.1 L/h median for testing. Where the the Transformer had RMSE value of 19.9 L/h, and around 13 L/h respectively, but the Transformer has a much better MAE in the test prediction performance at around 5 L/h median. This is likely due to the fact that the LSTM tries harder to follow the rapid changes than the Transformer. Therefore, the LSTM is penalized less since the error between the actual value and predicted value is on average smaller, thus receiving a smaller RMSE.

On the other hand, the Transformer has smoother transition lines and therefore gets a higher RMSE. The smooth transition, however, gives better performance on the MAE, indicating that the model is more tolerant of errors but is closer to the actual value. This is quite visible when inspecting the tail of both *Figure* 5.7 and *Figure* 5.11.

This means that the LSTM model, in its current state, is less likely to underestimate the inflow overall, due to the lower RMSE. But if used as a forecast system or controller, it could generate more noisy signals. This could lead to a bouncy forecast or more stress on the control system with a lot of unnecessary adjustments. While the transformer, in its current state, would be more likely to underestimate the inflow, and therefore might generate a less accurate forecast, but create a smoother signal. A smoother signal means a more stable prediction system or a less rapidly changing control system. Both of these qualities are important in such a forecast or control system, since both underestimating inflow and stress on the system are unwanted. The performance of the models, however, could likely be improved further with more tuning. However, more data would probably make a bigger impact on the overall performance.

5.4 Black box control system

The chosen machine learning model could then be loaded onto the Raspberry Pi, and with some implementation, work as either a forecast system or a control system. The forecast system would work by predicting the inflow, comparing it with the outflow, and displaying the risk of a flood forecast based on this to the end user. The control system would work by controlling the activation level of the sump tank, and therefore dictate how often the pumps would start, based on the inflow, and thus improving on the concept of the *Amesh* Tokyo system.

However, this was not done as none of the models would be robust enough to work either as a forecast system or a control system. More data and more tuning would be required for it to work within acceptable requirements. Since these models were only tuned on limited dates of one year, they would likely perform worse, when out of the specific seasons. It was also therefore decided that any API communication with weather data was unnecessary, as this API communication would be used as a rolling prediction system for the deployed machine learning system on the Raspberry Pi.

6 Conclusion

The system as a whole would not be able to work today; this is mainly due to the lack of performance of the machine learning models. Because of the lack of seasonality in the datasets, and the extremely small time window of training data, the machine learning models would perform poorly as either a forecast system or a control system. That being said, the models indeed show potential and might improve significantly in the future when more data is available and after further hyper-parameter tuning.

The IoT device, however, worked as expected and served its purpose as an HMI for the pumps, and as a logger. The IoT device managed to be a stable logger and gave the operator sufficient information about the state of the pumps. The HMI, however, was missing some features that should be addressed in future work.

Choosing the best machine learning model was not possible in the current state, as in an inflow prediction system both RMSE and MAE are important. Since a small RMSE indicates that the model is less likely to underestimate the inflow, it might create more noisy output signals. A smaller MAE, on the other hand, gives a less noisy output but is more likely to underestimate the inflow into the sump tank. Both of these problems are undesirable as underestimating inflow renders the system useless, and a bouncy prediction system is too unpredictable.

In conclusion, for the system to work as a prediction or control system, more time has to pass for the logger to collect more data for training. The data gathered in the short time the logger was operational was not sufficient to provide any acceptable results for the machine learning models. However, the results from the machine learning performance were promising, and if given more data with seasonality, and more hyper-parameter tuning, the models might be good enough to be used. In the future, they might be good enough to work as either a flood prediction system or a control system for the activation level of the sump tank.

7 future work

An important area for future exploration would be to revisit the system in a year or two, and conduct a fresh round of data analysis and training. This would allow us to see if the models would be able to predict a whole year, and learn patterns that lie in the nature of the seasonality in the data. Then, implementation of an API communication with the local weather station to create a rolling prediction system should be attempted. Lastly, the new machine learning model should be installed on the Raspberry Pi.

It would also be a good idea to restructure the IoT system, such that the logger and the HMI are moved into a cloud solution. The Raspberry Pi should only retrieve and send data from the frequency converters to the cloud. This would greatly increase the robustness and the security of the system. Since the logger is implemented on the Raspberry Pi, it is susceptible to SD-card corruption, leading to the loss of all the stored data. The Raspberry Pi is also susceptible to physical ”hacking”, although this is unlikely. Lastly, updating the HMI with the implemented machine learning model prediction should also be prioritized, to give the operator insight into what the model is predicting.

Bibliography

[1] *Lere om overvann*, nor. [Online]. Available: https://www.nve.no/naturfare/laer-om-naturfare/laer-om-overvann/, accessed: 02.05.2023.

[2] NVE, *Modul f2.304: Pumpeanlegg – prosjektering*, nor. [Online]. Available: https://www.nve.no/moduler/modul-f2-304-pumpeanlegg-prosjektering/, accessed: 02.02.2023.

[3] H. Mitamura and M. Fujie, 'Evolutionary transition of stormwater pump system in tokyo,' eng, *Journal of disaster research*, vol. 16, no. 3, pp. 421–428, 2021, ISSN: 1881-2473.

[4] *Lærebok Drenering og håndtering av overvann*, nor. Statens vegvesen, 2018.

[5] E. B. Abrahamsen, O. M. Brastein and B. Lie, 'Machine learning in python for weather forecast based on freely available weather data,' eng, 2018, ISSN: 1650-3686. [Online]. Available: http://hdl.handle.net/11250/2581934.

[6] Kartverket, *Hoydedata*. [Online]. Available: https://hoydedata.no/LaserInnsyn2/, accessed: 17.12.2022.

[7] Docker, *Develop faster. run anywhere.* [Online]. Available: https://www.docker.com/, accessed:01.04.2023.

[8] Influxdata, *It's about time. build on influxdb.* [Online]. Available: https://www.influxdata.com/, accessed:01.04.2023.

[9] Grafana, *Grafanalabs*. [Online]. Available: https://grafana.com/, accessed:01.04.2023.

[10] SeNorge, *Senorge.no.* [Online]. Available: https://www.senorge.no/, accessed: 10.01.2023.

[11] Pandas, *Pandas*. [Online]. Available: https://pandas.pydata.org/, accessed:17.04.2023.

[12] J. Brownlee, *Time series forecasting as supervised learning.* [Online]. Available: https://machinelearningmastery.com/time-series-forecasting-supervised-learning/, accessed:27.04.2023.

[13] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition*, eng. O'Reilly Media, Inc, 2022, ISBN: 9781098125967.

[14] IBM, *What are recurrent neural networks?* [Online]. Available: https://www.ibm.com/topics/recurrent-neural-networks, accessed:26.04.2023.

[15] V. Lendave, *A tutorial on sequential machine learning*. [Online]. Available: https://analyticsindiamag.com/a-tutorial-on-sequential-machine-learning/#:~:text=employing%20sequence%20modelling.-,What%20is%20Sequential%20Data%3F,stock%20price%20or%20sensor%20data., accessed:26.04.2023.

[16] J. S. Sepp Hochreiter, 'Long short-term memory,' eng, 1997.

[17] F. A. Gers, J. Schmidhuber and F. Cummins, 'Learning to forget: Continual prediction with lstm,' eng, *Neural computation*, vol. 12, no. 10, 2000, ISSN: 0899-7667.

[18] T. M. Ingolfsson, *Insights into lstm architecture*. [Online]. Available: https://thorirmar.com/post/insight_into_lstm/, accessed:30.04.2023.

[19] J. Brownlee, *A gentle introduction to dropout for regularizing deep neural networks*. [Online]. Available: https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/, accessed:30.04.2023.

[20] Y. B. Xavier Glorot Antoine Bordes, 'Deep sparse rectifier neural networks,' eng,

[21] A. Vaswani, N. Shazeer, N. Parmar *et al.*, 'Attention is all you need,' eng, 2017.

[22] Y. B. Dzmitry Bahdanau KyungHyun Cho, 'Neural machine translation by jointly learning to align and translate,' eng, 2015.

[23] Simplilearn, *What is epoch in machine learning?* [Online]. Available: https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-epoch-in-machine-learning#:~:text=An%20epoch%20is%20when%20all,dataset%20takes%20around%20an%20algorithm., accessed:28.04.2023.

[24] J. Brownlee, *A gentle introduction to early stopping to avoid overtraining neural networks*. [Online]. Available: https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/, accessed:28.04.2023.

[25] J. Wesner, *Mae and rmse — which metric is better?* [Online]. Available: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d, accessed:07.05.2023.

[26] J. Brownlee, *What does stochastic mean in machine learning?* [Online]. Available: https://machinelearningmastery.com/stochastic-in-machine-learning/, accessed:28.04.2023.

[27] Flask, *Flask*. [Online]. Available: https://flask.palletsprojects.com/en/2.3.x/, accessed:11.05.2023.

[28] T. Ntakouris, *Timeseries classification with a transformer model*. [Online]. Available: https://keras.io/examples/timeseries/timeseries_transformer_classification/, accessed:30.04.2023.

Appendix A

Task Description

University of
South-Eastern Norway

Faculty of Technology, Natural Sciences and Maritime Sciences, Campus Porsgrunn

FMH606 Master's Thesis

Title: Storm water flood prediction using machine learning

USN supervisor: Håkon Viumdal

External partner: Bergene Holm (David Bergene Holm and Håvard Omholt)

Task background:
Bergene Holm is one of the largest producers of construction lumber in Norway. That includes seven factories located around the southeast coast of Norway. The reason for the task I have chosen is to show Bergene Holm the possibilities of the data that they have accumulated over years. As well as learn more about the opportunity for historical data usage. Since there are a lot of Companies, including Bergene Holm, that only collects data but not uses it, this is a great opportunity to both explore how to create systems that utilize historical data and to use machine learning to try making mathematical models based on this historical data.
At the production cite in Kvelde, the factory is surrounded by vast fields, slightly descending in the direction of the factory. This causes a flood problem in the spring seasons when ice, snow and rain accumulate on the fields. This may lead to production shutdown due to electrical hazards, if the main power supply is submerged in water. Therefor a system for monitoring the pumps condition and effectiveness is required alongside with weather data.

To summarize, the main objective of this task is to create a flood prediction forecast system based on weather data and process data from two frequency generators for two pumps at the Kvelde planer factory.

Task description:
To achieve the main objective, the following subtasks are to be accomplished:
- Describe the prevailing situation with the potential flood problem, including the operation of the pumps.
- Make a literature study on local storm water flood prediction and potential outlooks
- Extend the pump system with an IOT device that communicates with the Frequency generators and store the data in a database (in this case a Raspberry pi 4).
- Design a web interface for HMI, with the access to the data in the database.
- Define an API communication to retrieve local weather in the area.
- Develop a model of the water dynamics on the field.
- Develop a machine learning algorithm that can predict the chance of flood, based on the weather forecast and the field model.
- Optional: Develop an alarm system to communicate with operators.
- Optional: Implement the developed alarm system

Student category:
Reserved for Industry master student in Industrial IT and Automation, Vebjørn Rimstad Wille

Is the task suitable for online students (not present at the campus)?
The student is employed at Bergene Holm, and will mainly work in their office/home

Practical arrangements:
The student will need to organize access to relevant dataset as a part of his Master thesis, with support from Bergene Holm, if needed. Equipment needed to run the project will be covered by Bergene Holm.

Supervision:
As a general rule, the student is entitled to 15-20 hours of supervision. This includes necessary time for the supervisor to prepare for supervision meetings (reading material to be discussed, etc).

Signatures:

Supervisor (date and signature): 01.02.2023

Student (write clearly in all capitalized letters): VEBJØRN RIMSTAD WILLE

Student (date and signature): 01.02.2023

Appendix B

PS220 PumpSmart

PumpSmart® PS220

Smart Control and Protection

PumpSmart PS220

PumpSmart®

The industry award-winning and patented pump control logic delivers real-time control and protection of your pumps while also providing valuable process insight. By protecting against pump failure due to process upsets, PumpSmart keeps your operation running longer and reduces unplanned repair activities and expense. By right-sizing your pumps to your system, we can reduce not only your energy consumption, but also wear & tear on your process systems.

ITT was an early adopter of variable speed pumping technology due to the inherent improvements that could be found in both efficiency and reliability of the pumping system. Leveraging electronic variable speed controllers allowed us to measure key parameters about the electric motor's performance and apply our unique expertise to create the PumpSmart Drive System. Functions unique to PumpSmart include calculating the flow and head of the pump without sensors, sensorless pump protection from process upset conditions, intelligent sleep and balancing load between multiple pumps.

Process Control

- Sensorless Flow Control
- Multipump Control
- Smart Control
- Intelligent Sleep
- Cavitation Control

Pump Diagnostics

- Sensorless Pump Protection
- Smart Flow
- % BEP Operation
- Smart Total Dynamic Head
- Flow Economy

PumpSmart® PS220

PumpSmart PS220 provides the next level in intelligent pumping by using a standard variable frequency drive and directly embedding pump specific algorithms onto the drive. The PS220 drive is a microprocessor based Direct Torque Controlled (DTC) adjustable speed AC drive. Pump specific algorithms combined with the advantage of sophisticated high-performance control of AC motors make it the ultimate variable speed drive solution for any pumping application.

3-Year Reliability Assurance Program

Protect your Goulds pump with a PumpSmart product and we'll GUARANTEE it against failure from pump process upsets for up to 3 Years[1]...or the parts are on us.

Goulds Pumps and PumpSmart Control Solutions are proud to offer this innovative new program that protects against pump failure that commonly results from inadvertent dry-running or operation against a closed discharge valve. If your Goulds pump fails while a PumpSmart Control product is on the job, we will provide the pump and seal repair parts free of charge.

See your Goulds Pumps representative to get started today.

[1] Program starts 3 years from pump shipment.

-PumpSmart is used across a wide-range of industries-

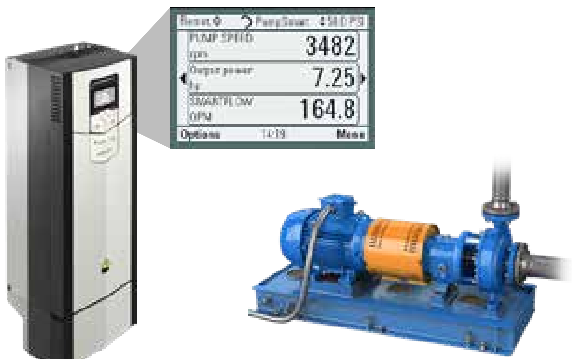
Sensorless Technology

Smart Flow & Smart TDH

Using speed and torque data from the motor and modelling the pump performance curve, PumpSmart is able to calculate the flow and the total dynamic head generated by the pump without instruments.

Determining the flow of a centrifugal pump can be a challenging exercise without a flow meter. PumpSmart is able to capture real-time data such as speed, torque and power and use this information to calculate the flow of the pump.

Smart Flow requires only four pieces of standard price book performance curve data. A self-calibration function takes into account changes in mechanical losses and volumetric efficiency, and separates the true hydraulic load to calculate the actual pump flow.



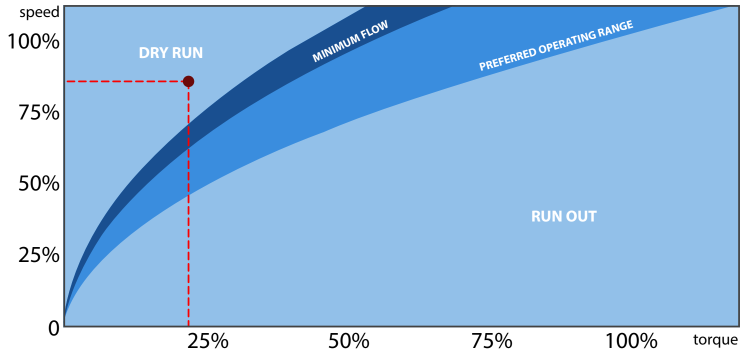
SmartFlow PID Control - PumpSmart allows controlling system flow in single pump or multi-pump systems without the need of an external flowmeter.

SmartFlow Flow Totalizer - Dial in a flow rate and let the PS220 do the rest. Run your batch operations to pump fixed volumes without using an external flowmeter.

Flow Economy - Flow Economy is a simple metric that defines how much fluid is moved per unit of energy. Similar to fuel economy of your car, Flow Economy defines how much flow (gpm or m³/h) can be moved with 1 kilowatt (kW) of power. Combined with Smart Flow, PumpSmart is able to calculate the Flow Economy of your pump allowing you to know what the true pump system efficiency is.

Sensorless Pump Protection

With patented sensorless pump protection algorithms, the PS220 determines the operating point of the pump at any speed and provides critical diagnostic information such as operation in relation to best efficiency point and protection against upset conditions such as dry-run, dead-head, shut-off, minimum flow and run-out.



Intelligent Sleep - A standard variable speed drive enables pump sleep mode based on the combination of system demand and a single preset minimum speed. Dynamic system conditions where static heads change make this method ineffective. The PS220's intelligent sleep mode function provides true protection against no demand conditions regardless of the user defined minimum speed.

Minimum Flow Bypass Control - By leveraging SmartFlow, the PS220 can trigger a relay output to energize a valve which will open and close a bypass line. A minimum flow setpoint triggers the bypass valve to open and when the pump reaches a user defined safe flow output which is corrected for speed by the PS220, the relay output triggers the bypass valve to close.

Intelligent Single & Multi-pump Control

Smart Torque Control

When changing the speed of a pump with a relatively flat head-capacity curve, a small speed change can result in a large swing in flow. This type of system can result in unstable flow, making control very difficult. (Fig.1)

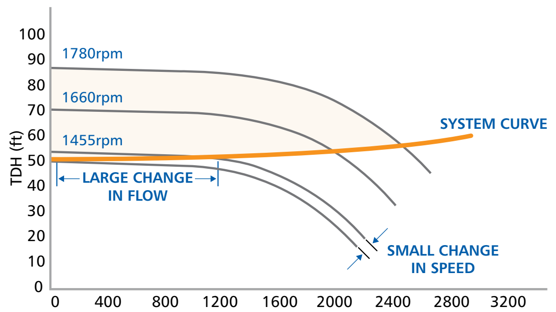


Fig.1

Smart Control is able to increase and decrease pump flow by changing the pump torque rather than the pump speed. Controlling to pump torque can change a relatively flat pump performance curve into a steep, easy-to-control pump performance curve. (Fig.2)

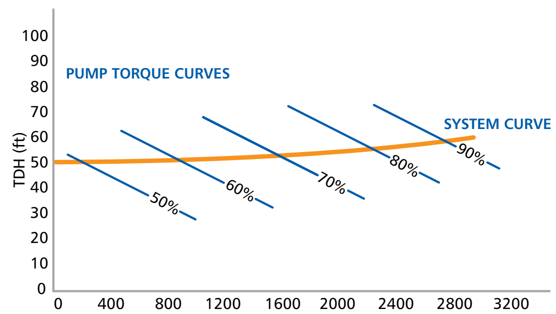


Fig.2

Multi-Pump Control

All too often, multi-pump systems end up running with all the pumps on, all the time. This situation leads to high vibrations, pressure buildup and excess energy consumption...to name a few. The PS220 runs only the pumps necessary to meet the system demand.

- Control up to a six-pump multi-pump system.
- Roaming master functionality allows for uninterrupted operation should any pump or drive become unavailable within the system.
- Balanced flow output between the operating pumps using Smart Torque Control functionality.
- Industry's first variable speed Sensorless Multi-pump Flow control solution.
- Selectable functions to limit the minimum or maximum allowable pumps to operate in a process.
- Switch leag lag status of pumps to maintain even wear between them based on runtime hours or number of Starts.
- Adjustable individual proof timers for staging and destaging pumps to reduce process fluctuations while bringing pumps online or taking them offline.

In summary, energy consumption is greatly reduced, and mean time between failure of the pumps and the surrounding system is vastly improved.

Process Control & Protection

As standard PumpSmart systems come equipped with advanced process control features that help optimize your pumping system for maximum uptime, reliability and energy savings.

PumpSmart is pump-specific and was developed to protect the pump and optimize pump control. PumpSmart can be applied to any manufacturer's centrifugal or positive displacement pump.

Horizontal Centrifugal Pump

Between Bearing Centrifugal Pump

Submersible Pump

Vertical Centrifugal Pump

Twin Screw/Gear Pump

Progressive Cavity Pump

Waste Water Functions

Pump Clean - The PS220 automatically detects and removes clogging substances from the pump impeller by monitoring the pump motor torque preventing damage from pump lockup.

Pipe Clean - This function allows flushing of the pipe system that helps with reducing sedimentation in the pipes resulting in lower wear on piping.

Pipe Fill - The PS220's pipe fill function allows gradual filling of a pipeline before normal process control operation.

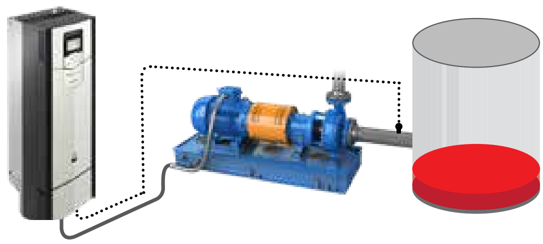
Snore - The PS220 snore function overrides the stop level to empty a tank for the purpose of removing oil & grease and other floating debris from the water surface. This results in a cleaner sump with eliminating the need to pump down and clean the sump manually.

Cavitation Control & Protection

Low suction pressure can lead to the onset of cavitation, resulting in reduced flow and lower pump efficiencies. Prolonged exposure can even result in eventual pump failure.

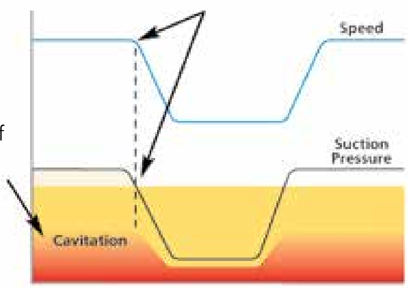
PumpSmart can monitor the suction conditions of your pump to protect against cavitation. Cavitation Control improves overall pump reliability in low Net Positive Suction Head (NPSH) services that routinely cause pump failure.

Typical Services:
- Evaporator
- Condensate
- Batch Transfer
- Unloading



As suction pressure drops to a critical level PumpSmart reacts by slowing down the pump.

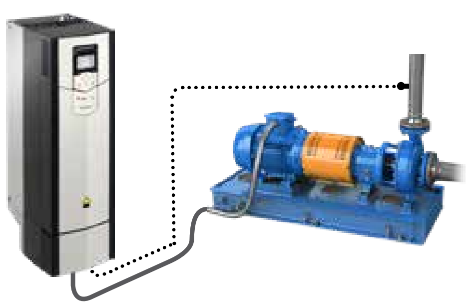
Operating a pump with low suction pressure can result in the formation of cavitation. Reducing the pump speed can reduce the NPSH requirements of the pump which can help suppress the onset of cavitation.



Integrated Process Control

The PS220 offers automatic pump control by integrating the pump controller in the drive. No external controller is required, making PumpSmart a simple and cost-effective solution for your pumping needs.

Process Control Features
- Single Pump
- Multipump
- Cavitation Control
- PID Smart Flow



Communications & Other Features

Ease of Configuration

The PS220 Wizards guide you step by step to take a PS220 from factory defaults to any application specific setup, making it one of the industry's simplest variable speed drives to commission and configure.

Bluetooth Connectivity

Using Bluetooth, the PS220 offers easy access to drive parameters and control using a bluetooth enabled smartphone or tablet.

Drive PC Tools

Drive composer provides a built-in drive control panel allowing users to start, stop, and set the direction, speed, and torque reference values of the connected drive.

- View and set drive parameters
- Custom workspace
- Custom windows
- Save and download parameters
- Control the drive using the built-in control panel
- Connects via USB through ACP-AP panel network

Flexible Connectivity to Plant Automation Systems

Fieldbus adapter modules enable communication and software. The PS220 is compatible with a wide range of fieldbus protocols. The plug-in fieldbus adapter module can easily be mounted inside the drive. Other benefits include reduced wiring costs when compared with traditional input/output connections. Fieldbus systems are also less complex than conventional systems, resulting in less overall maintenance. Adapters can be added to the drive at anytime.

Safe torque off as standard

Safe torque off (STO) is used to prevent unexpected startup and in stopping-related functions, enabling safe machine maintenance and operation. With safe torque off activated, the drive will not provide a rotational field. This prevents the motor from generating torque on the shaft. This function corresponds to an uncontrolled stop in accordance with stop category 0 of EN 60204-1.

The easy to connect and configure safety functions module (FSO-12 and -21) offers a wide range of safety functions and a self diagnostic function.

Removable memory unit

Stores all the software and parameter configurations in an easily replaceable and simple-to-install module. Situated on the control unit, the memory unit can easily be removed for maintenance, update or replacement purposes. This common type of memory unit is used throughout the PS220 series.

Hardware Options

Hardware Configurations

- Wall Mount Drives
- Ultra Low Harmonic Drives
- Flange Mount Options
- Cabinet Drives
- Regenerative Drives
- Drive Modules

Main Features on All Drives

- Enclosure classes IP20, IP21, IP55
- Safe torque off (STO) as standard
- Coated boards as standard
- Controllable cooling fan
- EMC filter option
- du/dt filter option
- Built-in choke

<u>Wall Mount Units</u>

Frames R1-R0
0.75Kw - 250Kw
(1HP - 350HP)

Frame Size	Height IP21 in/mm	Depth in/mm	Width in/mm	Weight lb/kg
R1	16 / 405	8.9 / 226	6.1 / 155	13.2 / 6
R2	16 / 405	9.8 / 249	6.1 / 155	17.6 / 8
R3	18.5 / 471	10.3 / 261	6.7 / 172	22 / 10
R4	22.6 / 573	10.8 / 274	8 / 203	40.8 / 18.5
R5	28.7 / 730	10.8 / 274	8 / 203	50.7 / 23
R6	28.6 / 726	14.1 / 357	9.8 / 251	99.2 / 45
R7	34.6 / 880	14.4 / 365	11.2 / 284	121.3 / 55
R8	37.9 / 963	15.2 / 386	11.8 / 300	154.3 / 70
R9	37.9 / 963	16.3 / 413	15 / 380	216 / 98

<u>Cabinet Drives</u>

Frames R6-R8
55Kw - 200Kw
(75HP - 200HP)

Frames R9-R11
200Kw - 500Kw
(250HP - 700HP)

Frames n x R8i
500Kw - 2800Kw
(500HP - 3000HP)

Frame Size	Height IP22/42 in/mm	Height IP54 in/mm	Depth in/mm	Width in/mm	Weight lb/kg
R6	84.4 / 2145	91.2 / 2315	16.9 / 430	26.5 / 673	528 / 240
R7	84.4 / 2145	91.2 / 2315	16.9 / 430	26.5 / 673	550 /250
R8	84.4 / 2145	91.2 / 2315	16.9 / 430	26.5 / 673	583 / 265
R9	84.4 / 2145	91.2 / 2315	32.7 / 830	27.5 / 698	825 /375
R10	84.4 / 2145	91.2 / 2315	32.7 / 830	27.5 / 698	1169 / 530
R11	84.4 / 2145	91.2 / 2315	32.7 / 830	27.5 / 698	1279 / 580

Additional Features for Cabinet Drives

- Cabinet light and heater option
- Marine construction option

Main Connection	
Voltage Range	3-phase, U_{N2} = 208 to 240 V
	3-phase, U_{N3} = 380 to 415 V
	3-phase, U_{N5} = 380 to 500 V
	3-phase, U_{N7} = 525 to 690 V
Frequency	50/60 Hz ± 5%
Motor Connection	
Voltage	0 to U_{N2} U_{N3} U_{N5} U_{N7}
Frequency	0 to +500 Hz
Motor Control	Direct torque control

Product Compliance
CE, UL cUL 508A or cUL 508C CSA C22.2 NO. 14-10, C-Tick, RoHS ATEX-certified Safe Disconnection Function Low Voltage Directive 2006/95/EC Machinery Directive 2006/42/EC EMC Directive 2004/108/EC Quality assurance system ISO 9001 & Environmental system ISO 14001
EMC according to EN 61800-3:2004 + A1:2012
Categories C3 and C2 with internal option

Engineered Options

Pre-engineered control panels for indoor, outdoor and water tight installations rated for NEMA 1 (IP21), NEMA 12 (IP54), NEMA 3R and NEMA 4/4X (IP66) environments.

- Simplex
- Duplex
- Triplex
- Quad

Packaged systems which include pumps, controls and piping providing a complete skid mounted solution.

Bucket Mounted options for easy installations into a motor control center.

Electrical Houses with HVAC, lighting and complete “turn-key” integration including customer specific options such as redundant PLC’s, transformers & soft starts.

240 Fall Street
Seneca Falls, NY 13148
Phone: 315.568.2811
Fax: 315.568.2418
www.ittproservices.com

B.PumpSmartPS220.en-US.2020-12

Appendix C

Flygt N3153 Pumps

The Flygt N-technology pump series for water and wastewater

HIGH EFFICIENCY WITH CLOG-FREE PERFORMANCE

FLYGT
a xylem brand

No clogging. No wasted energy. Just trouble-free pumping

Our Flygt N-pumps (1.3 kW – 680 kW) are designed to handle the world's toughest water and wastewater applications. And now, with our Adaptive N™ technology in all smaller pumps, you get a superior way to avoid clogging, reduce unplanned maintenance and cut your energy bills. That adds up to total peace of mind – and big savings over the long term.

Our vast fluid handling knowledge and dedication to research and development leads to technological advances and continuous improvement. That's why Flygt N-pumps are currently at work in millions of installations worldwide. Quite simply, they have proven to be the best and most reliable choice for both dry and submersible installations.

Sustained high efficiency saves money

When solid objects, such as stringy fibers and modern waste, enter the inlet of a conventional pump, they tend to get caught on the leading edges of the impeller vanes. This build-up reduces the impeller's efficiency, resulting in increased power consumption (**Fig. A**).

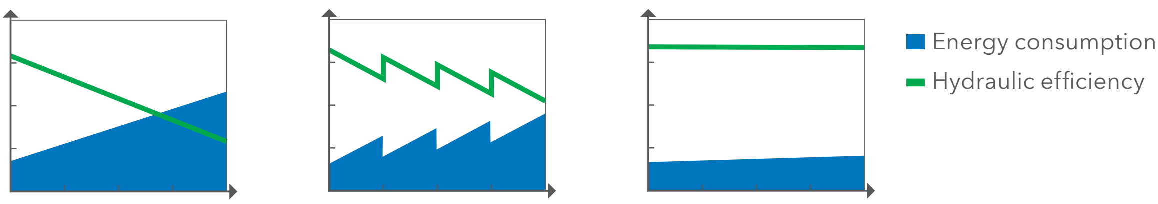
Avoiding unplanned service calls

With conventional wastewater pumps, a continued build-up of solids inside the impeller can trip the panel overload or motor protection function, causing clogging and leading to costly unplanned service calls (**Fig. A**). Even if the pump is running intermittently, hydraulic efficiency is reduced since the solids build-up needs to be removed by backflushing when the pump is shut off at the end of the operating cycle (**Fig. B**). Not until the next cycle begins is efficiency restored to its initial value when the impeller is free from solid objects. The Flygt N-technology has a mechanically self-cleaning design that handles the toughest modern wastewater challenges. With sustained high efficiency it minimizes running hours and energy cost over time (**Fig. C**).

Experience the power of N

Whether you are working with wastewater, stormwater or another application, you will find a broad range of N-pumps designed to take on the toughest challenges and get the job done. Robust, reliable and self-cleaning, they cut your energy bills and virtually eliminate unplanned maintenance.

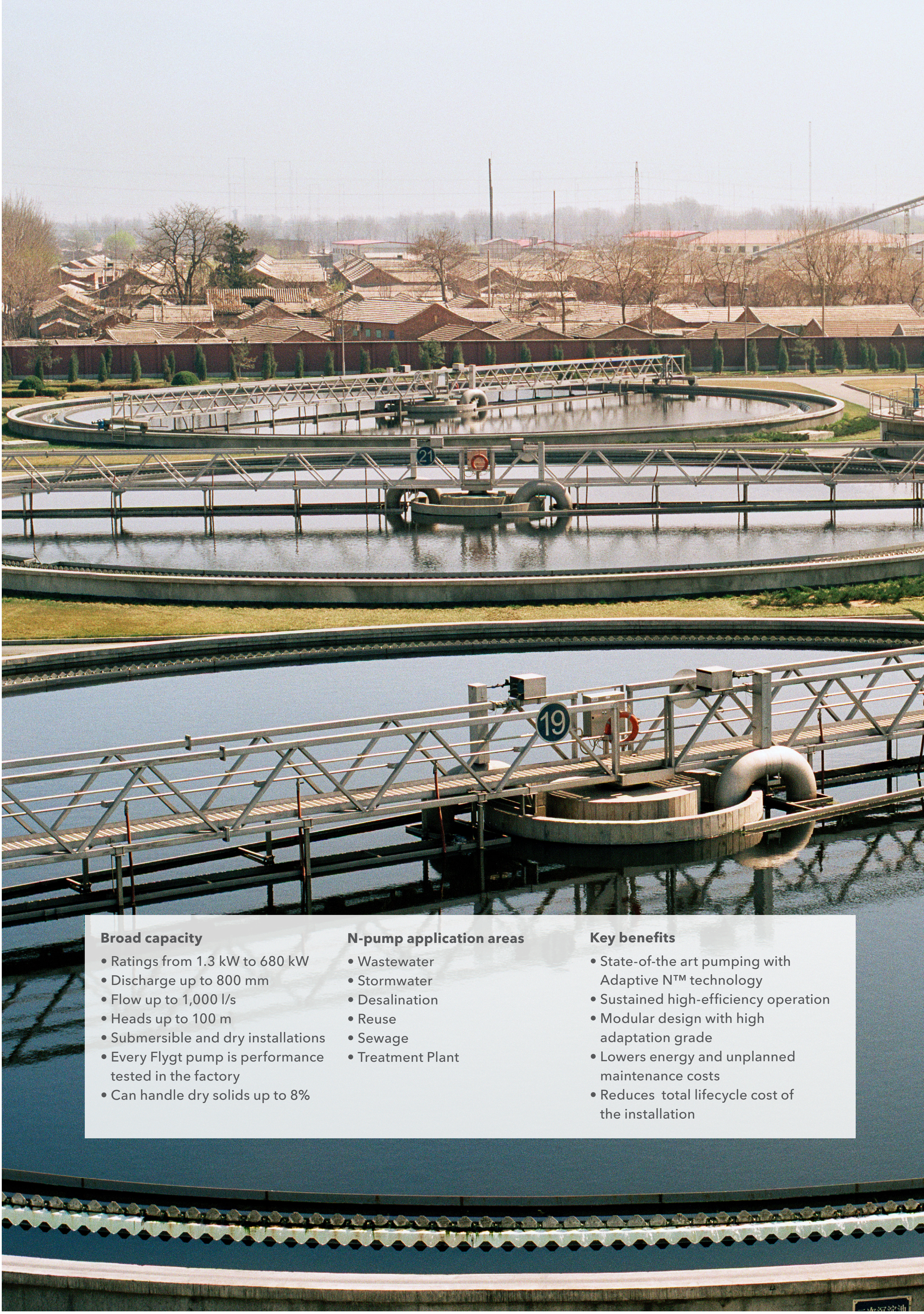
Sustained high efficiency with Flygt N-pumps



A. Conventional wastewater pump

B. Conventional pump running intermittently

C. Flygt N-pump



Broad capacity

- Ratings from 1.3 kW to 680 kW
- Discharge up to 800 mm
- Flow up to 1,000 l/s
- Heads up to 100 m
- Submersible and dry installations
- Every Flygt pump is performance tested in the factory
- Can handle dry solids up to 8%

N-pump application areas

- Wastewater
- Stormwater
- Desalination
- Reuse
- Sewage
- Treatment Plant

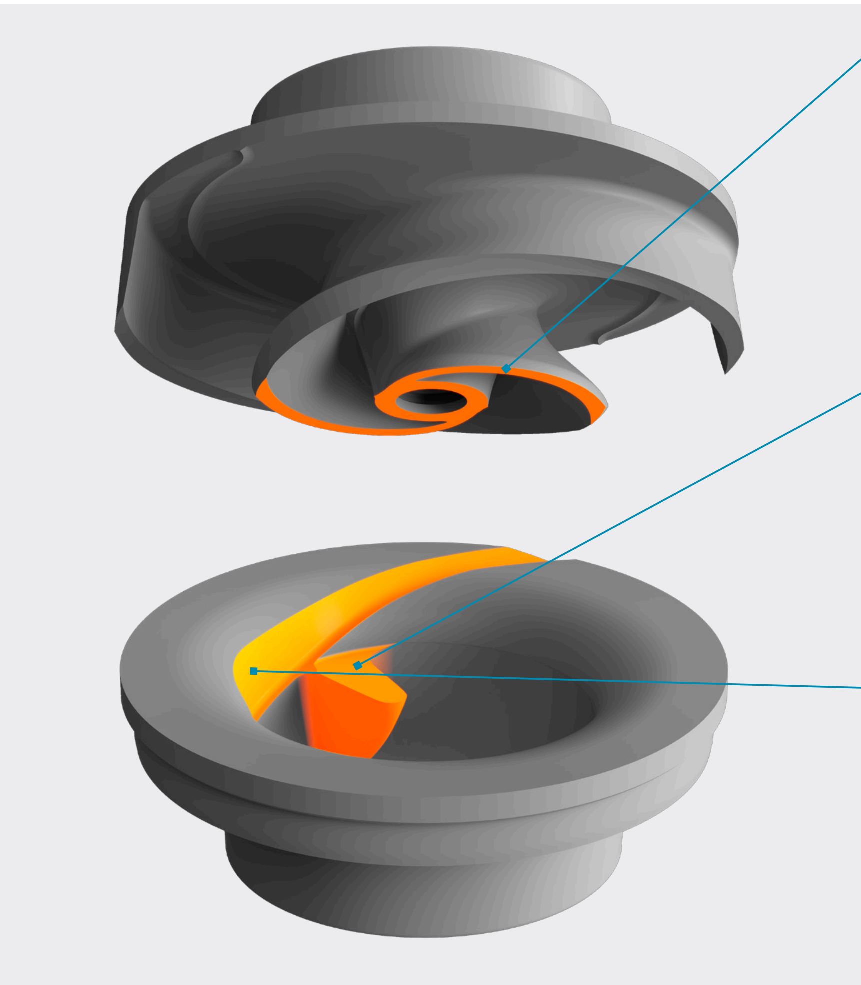
Key benefits

- State-of-the art pumping with Adaptive N™ technology
- Sustained high-efficiency operation
- Modular design with high adaptation grade
- Lowers energy and unplanned maintenance costs
- Reduces total lifecycle cost of the installation

Advanced technology guides the design of every component

From the motor and seals to the shaft and impellers, every component in a Flygt N-pump is designed, engineered and manufactured to optimize operation and prolong service life. Advanced technology guides the design of all aspects of the pump. One example is the Adaptive N hydraulic system, which is available only with lower-capacity pumps.

The fundamental N-technology, which was pioneered by Flygt, has been incorporated into our pumps for years. A more recent innovation is our Adaptive N impeller and Adaptive N hydraulic technologies (see below) which combine a unique geometry, dual-blade impeller and other patented features to give you sustained high efficiency and smooth operations. When larger objects enter the pump, the impeller lifts up due to the forces from these solid obects passing through. This self-cleaning design results in up to 25% lower energy consumption, regardless of impeller speed or duty point. It also minimizes vibrations, resulting in a longer life span for the mechanical components.



1. Backswept leading edges – ensures no sticking

When solids enter the pump, they are met by the N impeller. The optimized blade geometry, with its backswept leading edges, ensures that no material sticks to the impeller.

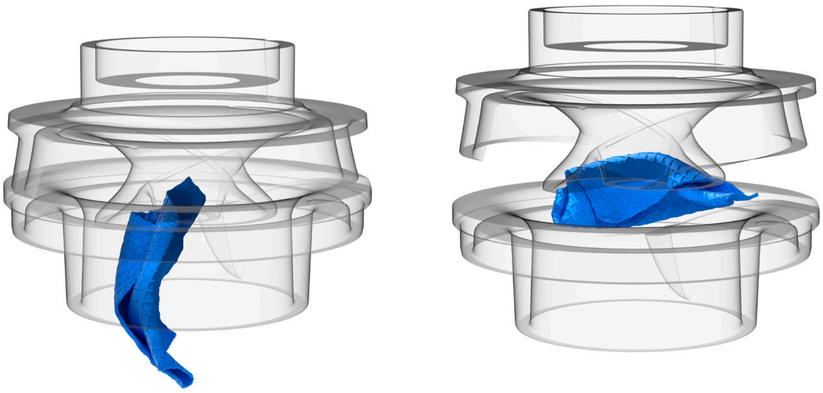
2. Integrated guide pin – clears the center

A guide pin inserted into the insert ring clears the center of the impeller by pushing solids along the leading edges towards the periphery of the impeller for removal.

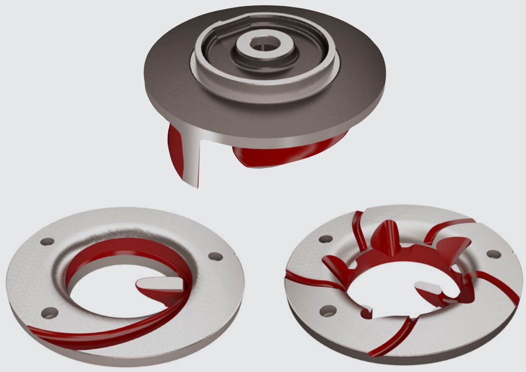
3. Relief groove – facilitates transport

When solids reach the perimeter of the inlet, they are transported inside the relief groove, guided along the edge of the impeller vane, through the volute and out of the pump.

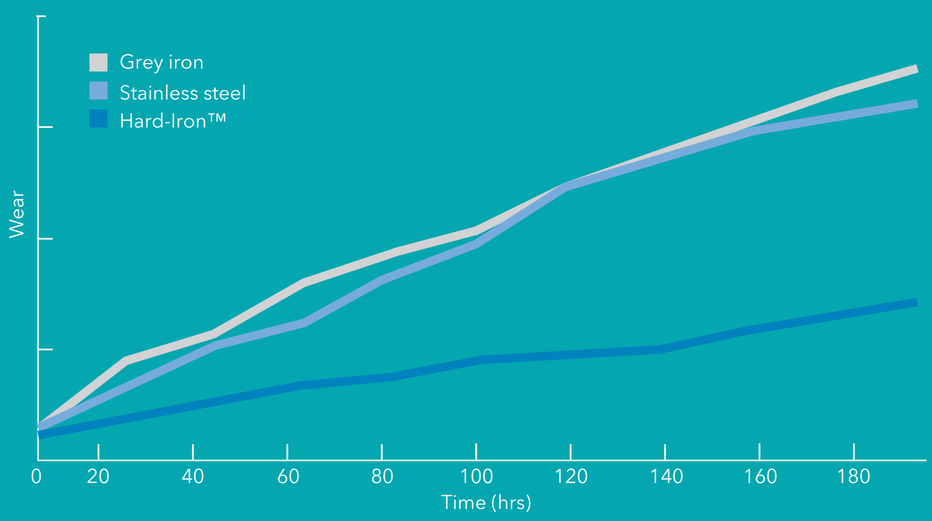
Adaptive N – lifts up for large objects
When larger objects enter the pump, the impeller lifts up due to the forces from these solid objects passing through. This avoids clogging and assures continuous, energy-efficient pumping.



Choice of impeller materials
With our Adaptive N impeller, you can also choose the optimal material type for your needs: Hard-Iron™, grey iron or stainless steel. Flygt's patented Hard-Iron alloy is developed specifically for tough wastewater applications. Accelerated wear tests prove that Hard-Iron (60 HRC) hydraulic components prolong the lifetime by a factor of five, compared to standard grey iron material.



Adaptive N™ hydraulic materials - accelerated wear test



After 200 hours, the Hard-Iron impeller proved to be five times more wear resistant than the grey iron version. The stainless steel impeller showed wear comparable to the standard grey iron material.

Low-capacity pumps

This series of Flygt N-pumps includes models capable of handling capacities up to 100 l/s. Like all Flygt N-pumps, they help reduce the total life-cycle costs of your installation.

1. Better heat transfer
Our specially designed and manufactured motor provides enhanced cooling because heat losses are concentrated around the stator. Trickle impregnated (not applicable for 3069) in resin (Class H insulation), the stator windings are rated at 180°C (355°F) and enable up to 30 starts per hour.

2. Cable entry
Water-resistant cable entry provides both sealing and strain relief functions to ensure a safe installation.

3. Sensors
Thermal sensors embedded in the stator windings prevent overheating. Optional leakage sensors in the stator and oil housings are also available.

4. Long-life bearings
Durable bearings provide a minimum service life of 50,000 hours.

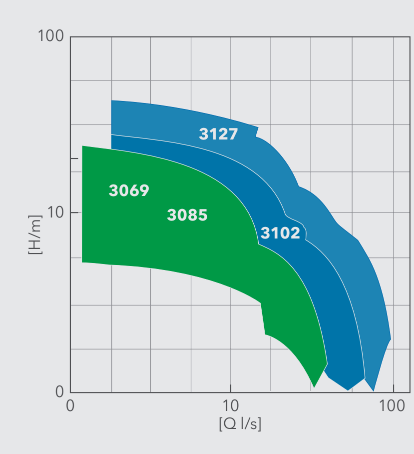
5. Enduring seals
The Griploc™ system consists of two sets of mechanical shaft seals that operate independently to provide double security against leakage.

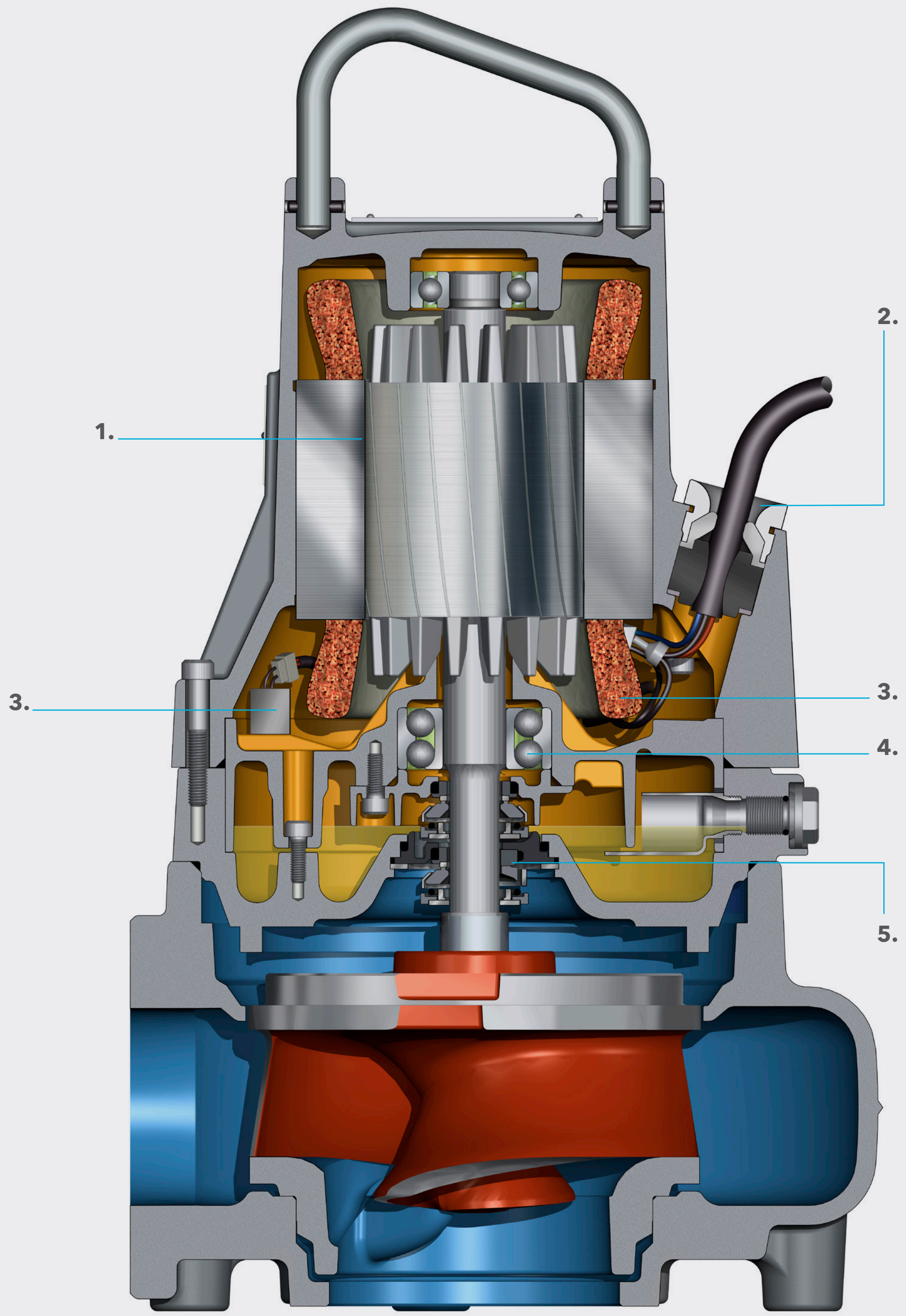
Compliance
Each pump is tested and approved in accordance with national and international standards, including 60034-1 and CSA. Pumps are available in explosion-proof versions for use in hazardous environments, and are approved by the Factory Mutual, European Standard and IEC.

POWER RATINGS AND SIZE

Model	**3069**	**3085**	**3102**	**3127**
Power rating - kW	1.5–2.4	1.3–2.4	3.1–4.5	4.7–8.5
Discharge size - mm	50	80	80	80
	65		100	100
	80		150	150

Performance, 50 Hz





1.
2.
3.
3.
4.
5.

Medium-capacity pumps

For demanding pumping duties, medium-capacity models handle fluid transport for capacities up to 500 l/s. Robust and highly efficient, they provide clog-free performance to achieve the lowest overall life-cycle costs.

1. Better heat transfer
Our specially designed and manufactured motor provides enhanced cooling because heat losses are concentrated around the stator. Trickle impregnated in resin (Class H insulation), the stator windings are rated at 180°C (355°F) and enable up to 30 starts per hour.

2. Efficient cooling
These pumps are cooled either by the surrounding liquid or liquid/air, in more demanding applications, with an internal closed-loop cooling system.

3. Inspection chamber
To increase operational reliability, an inspection chamber between the seal unit and the bearings enables rapid spot checks and maintenance. In the case of a seal failure, a built-in sensor provides an early warning of any fluid build-up, thus reducing the risk of expensive repair work.

4. Cable entry
Water-resistant cable entry provides both sealing and strain relief functions to ensure a safe installation.

5. Sensors
Thermal sensors embedded in the stator windings prevent overheating, and a leakage sensor in the inspection chamber minimizes the risk for bearing and stator failure.

6. Long-life bearings
Durable bearings provide a minimum service life of 50,000 hours.

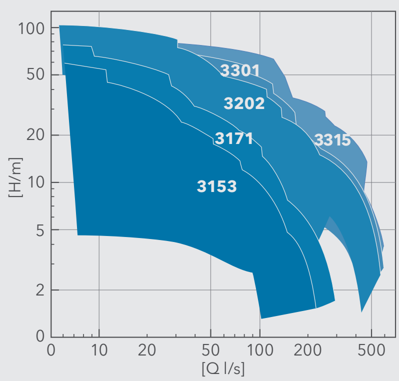
7. Enduring seals
The Flygt Plug-in™ seal with the Active Seal™ system offers increased sealing reliability and zero leakage into the motor, thereby reducing the risk of bearing and stator failure.

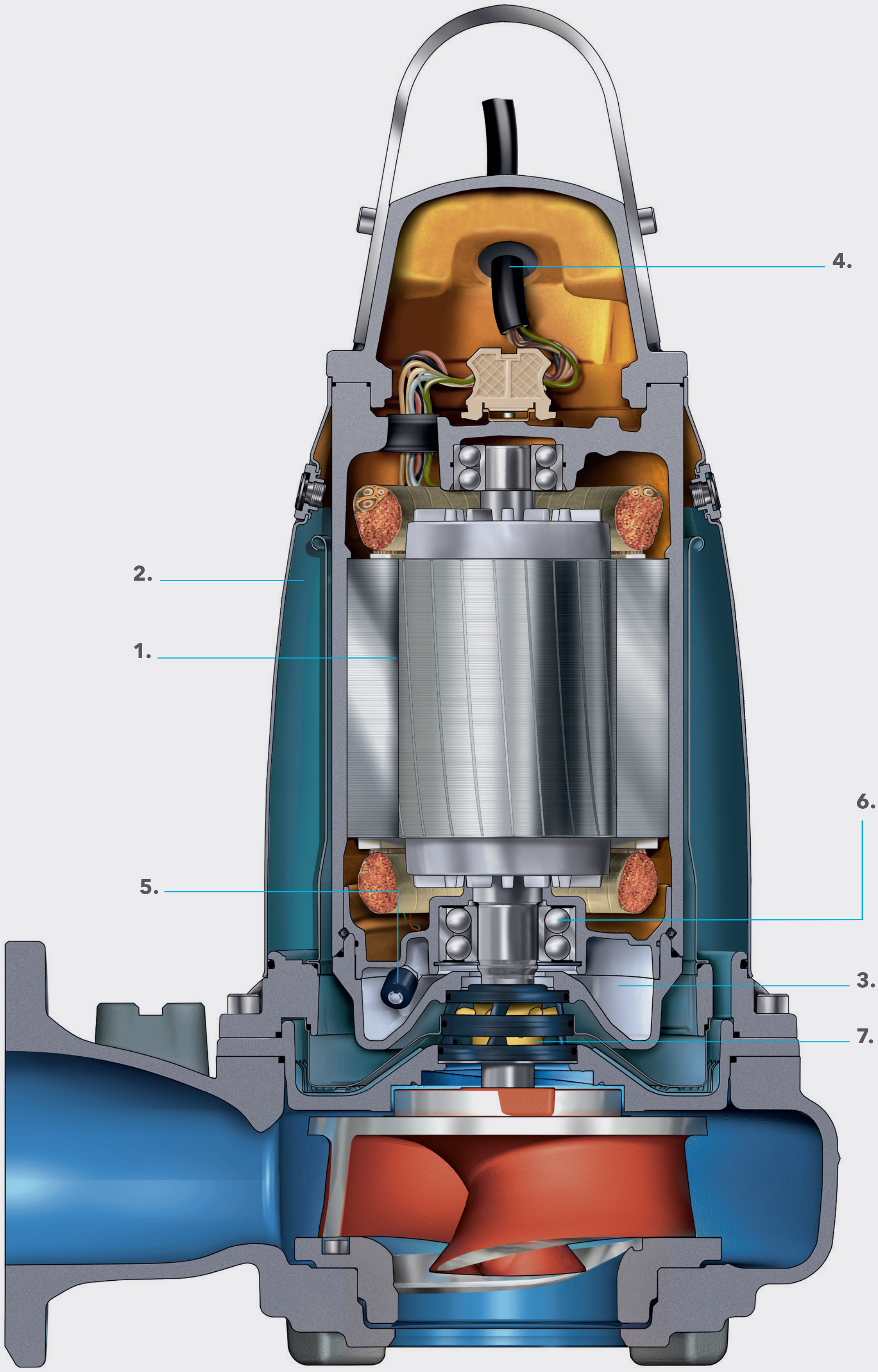
Compliance
Each pump is tested and approved in accordance with national and international standards, including 60034-1 and CSA. Pumps are available in explosion-proof versions for use in hazardous environments, and are approved by the Factory Mutual, European Standard and IEC.

POWER RATINGS AND SIZE

Model	**3153**	**3171**	**3202**	**3301**	**3315**
Power rating - kW	7.5–15	15–22	22–47	37–70	48–105
Discharge size - mm	80	100	100	150	150
	100	150	150	250	250
	150	250	200	300	300
	200		300	350	350
	250				

Performance, 50 Hz




4.
2.
1.
6.
5.
3.
7.

High-capacity pumps

When higher capacity is required, the Flygt N-pump series has pumps to do the job. These models deliver unprecedented pumping power – reliably and efficiently.

1. Better heat transfer
Our specially designed and manufactured motor provides enhanced cooling because heat losses are concentrated around the stator. Trickle impregnated in resin (Class H insulation), the stator windings are rated at 180°C (355°F) and enable up to 15 starts per hour.

2. Efficient cooling
These pumps are cooled either by the pumped liquid or liquid/air with an internal closed-loop cooling system.

3. Cable entry
Water-resistant cable entry provides both sealing and strain relief functions for a safe installation.

4. Sensors
Thermal sensors in the stator windings prevent overheating, and an analog temperature sensor monitors the lower bearing. The stator housing/ leakage chamber and the junction box are equipped with leakage sensors. The sensors decrease the risk of bearing and stator failure.

5. Long-life bearings
Durable bearings provide a minimum service life of 100,000 hours.

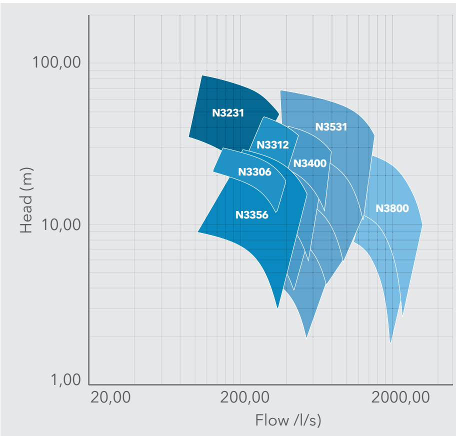
6. Enduring seals
Two sets of mechanical shaft seals work independently for double security. The Active Seal™ system offers increased sealing reliability and zero leakage into the motor, thereby reducing the risk of bearing or stator failure.

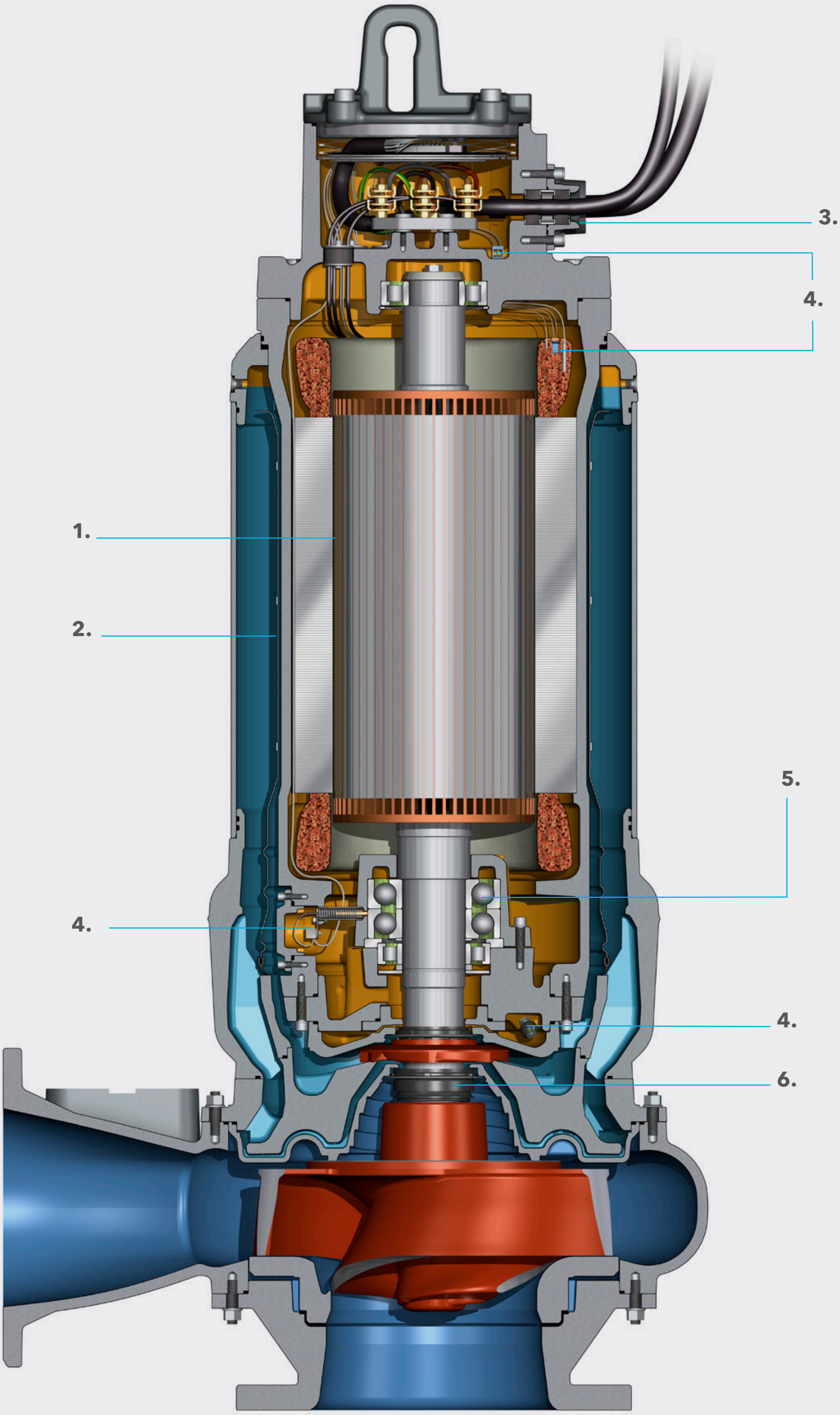
Compliance
Each pump is tested and approved in accordance with national and international standards, including IEC 60034-1 and CSA. Pumps are available in explosion-proof versions for use in hazardous environments, and are approved by the Factory Mutual, European Standard and IEC.

POWER RATINGS AND SIZE

Model	**3231**	**3306**	**3312**	**3356**	**3400**	**3531**	**3800**
Power rating - kW	70–215	58–100	55–250	45–140	40–310	40–680	225–550
Discharge size - mm	200	300	300	350	400	500	800

Performance, 50 Hz





3.
4.
1.
2.
5.
4.
4.
6.

Install it and control it just the way you want it

Regardless of the size or type of N-pump you require, we offer a wide range of modular installation concepts as well as purpose-built monitoring and control systems. Our modular installation concepts let you customize inlets and outlets to fit your needs exactly.

MAS 801

MAS 801 – the smart way to monitor performance

This new pump monitoring system offers powerful data management capabilities to ensure you are constantly updated on each pump's conditions and operational status. We've removed the traditional sensor cable, which means simplified handling, improved measurement quality and fewer callouts. Digital communication now takes place in the power cable, made possible by the new Flygt SUBCAB range with integrated signal leads.

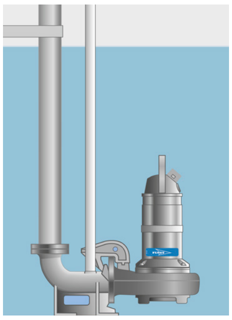
Flygt SmartRun

Flygt SmartRun® – optimal reliability

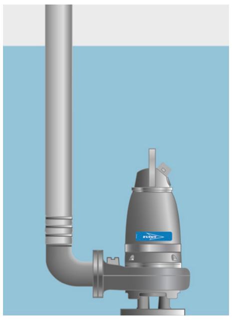
For pump stations with up to three alternating pumps, the Flygt SmartRun pump controller handles pump cleaning, pipe cleaning, sump cleaning, soft starts and stops.

The integrated intelligence and variable speed control make it the perfect match for Flygt N-pumps – a combination that potentially realizes energy savings of up to 50%.

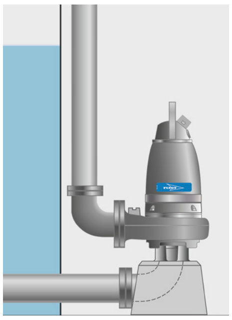
Flexible Installations



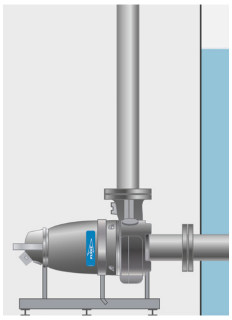
P-installation
For semi-permanent wet well installations. The pump is installed with twin guide bars on a discharge connection.



S-installation
A semi-permanent free standing installation. Transportable version with pipe or hose connection.



T-installation
A vertically-mounted, permanent dry well or in-line installation with flange connections for suction and discharge pipework.



Z-installation
A horizontally-mounted, permanent dry well or in-line installation with flange connections for suction and discharge pipework.

Take advantage of our design and engineering expertise

Are you getting the most out of your pump station designs? If you have questions regarding fluid dynamics, optimizing your sumps, water hammer calculations or even service, we can help. Flygt engineers have been researching and designing pump stations for over three decades to achieve the lowest life-cycle costs. And we have a strong service network to support you.

One of the biggest challenges in designing a pump station is to achieve a balance between efficiency and performance. Often times, pump stations are over-designed, resulting in higher costs. One key focus is to secure the best possible inlet conditions while minimizing sedimentation and pump station size.

Optimizing your flow rates

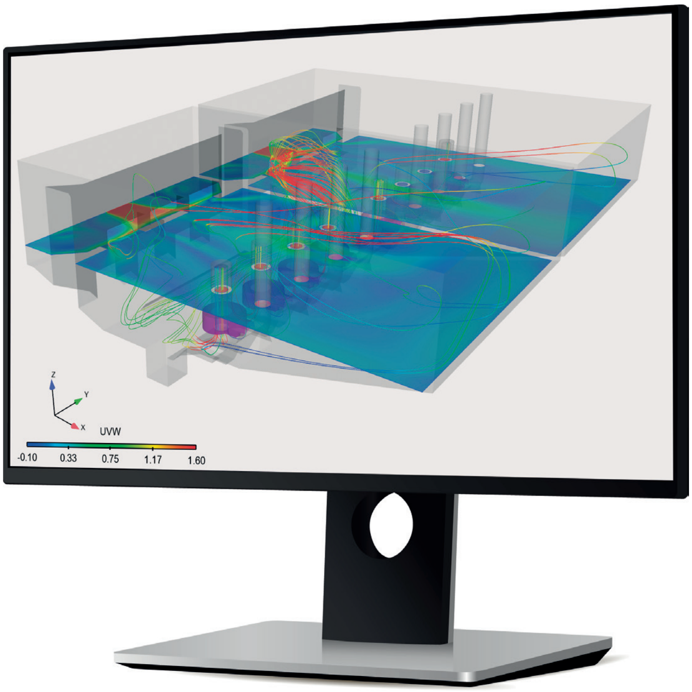
Other critical factors include the number, type and arrangement of pumps, variable flow conditions in the approach area, the geometry of the structure itself and other site-specific factors. It's also vital to consider and plan for operational concerns such as pump control schemes and access for equipment service.

Finding the optimal solution

Whether you're looking for a pre-engineered and packaged pump station, standardized design or a custom design, we can help you find the best solution for your project's needs.

Computational fluid dynamics

To verify a proposed new sump design, we use computational fluid dynamics (CFD), a mathematical modeling technology. It allows us to analyze flow patterns under different operating conditions. Flygt pioneered the use of CFD to verify sump design, and we have been using it for more than 30 years.



Extensive engineering know-how

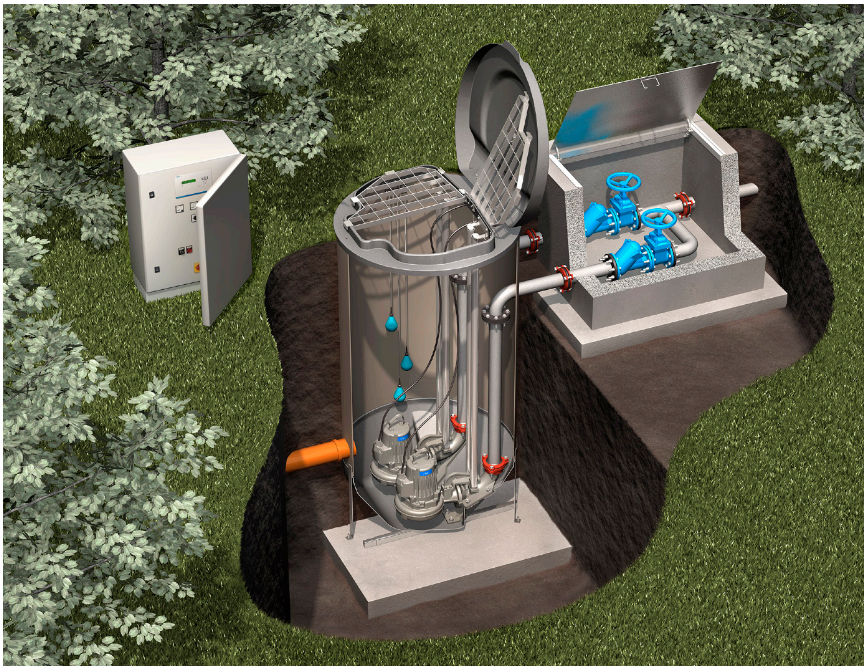
We provide a broad range of engineering services, including:

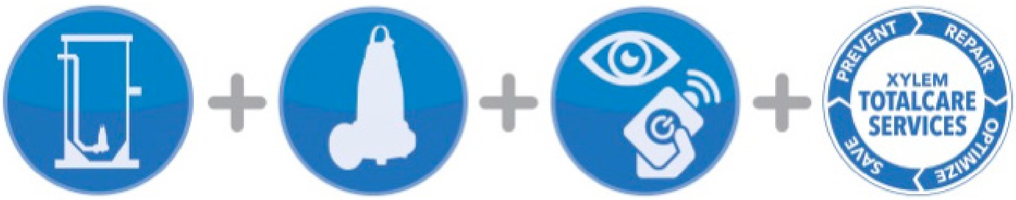
- System analysis and calculations
- Sump design
- Water hammer calculations
- Pump start analysis
- Transient analysis
- Computational Fluid Dynamics (CFD)
- Scale model testing

Complete pre-engineered solutions for all your needs

Even better together
Do you need a swift station rollout in a municipal or commercial area? You'll be happy to know that we offer a wide range of pre-engineered packaged pump stations that make the job easier and more cost-efficient. Ideally suited for our premium N-pumps, Flygt packaged pump stations come with piping and valve systems, all installation accessories and monitoring and control equipment.

Trouble-free pumping in a package
Available in a range of designs, sizes and materials, our packaged pump stations feature a common self-cleaning design, optimized for trouble-free and efficient pumping. One of the more popular versions is the Flygt TOP design pictured here.





Support for your Flygt pumps
Our global network of local service centers and service partners, provides integrated services to support safe, efficient and reliable operation. Count on us for a quick, professional response and quality maintenance services, using genuine Flygt spare parts.

Genuine Flygt spare parts and warranty
When downtime isn't an option, rely on our global service network to deliver genuine Flygt spare parts – quickly and efficiently. All Flygt spare parts are backed by a 15-year availability guarantee. With our higher-capacity pumps, we provide a 20-year availability guarantee.

The power to adapt

Options table

Customize your Flygt N-pump with optional equipment.

Flygt N-pump model	3069	3085	3102	3127	3153	3171	3202	3301	3315	3231	3306	3312	3356	3400	3531	3800
Option/Product																
Motor																
Premium efficiency (IE3)		⊂	○	○	○	○	○	○		⊂	⊂	⊂	⊂	⊂	⊂	
Hydraulics																
Guide pin	●	●	●	●	◐	◐	⊂									
Hard-Iron™	○	○	○	○	○	○	○	○	○	○	○	○	○		○	○
Chopper N				○	⊂	⊂	⊂									
Adaptive N™	●	●	●	●												
Seal system																
Griploc™ seal	●	●	●	●												
Plug-in™ seal					●	●	●	●	●							
Active Seal™					●	●	●	●	●	●	●	●	●	●	●	●
Spin-out™		●	●	●	●	●	●	●	●	◖	◖	◖	◖	◖	◖	◖
Seal flush										○	○	○	○	○	○	○
Cooling systems																
1. w/o cooling jacket	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
2. Closed-loop cooling	○				○	○	○	○	○	○	○	○	○	○	○	○
3. Pump media	○									○	○	○	○	○	○	○
4. External	○				○	○	○	○	○	○	○	○	○	○	○	○
Installation method																
P	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
S	●	●	●	●	●	●	●	◖	◖	●	●	●				
T		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
Z		●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
L			●	●												
Accessories																
Flush valve		○	○	○	⊂	⊂	⊂	⊂	⊂							
Pump monitor *Prepared for*																
– Mini CAS	●	●	●	●	●	●	●	●	●							
– MAS						○	○	○	○	●	●	●	●	●	●	●
Pump control																
– SmartRun™		○	○	○	○	○	○	○								
– MultiSmart		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○
– MyConnect		○	○	○	○	○	○	○	○							
– FGC	○	○	○													

● = Standard
○ = Optional
◐ = Standard but also optional, depending on model
◖ = Standard or not available, depending on model
⊂ = Optional or not available, depending on model

Xylem [ˈzīləm]

1) The tissue in plants that brings water upward from the roots
2) A leading global water technology company

We're 12,000 people unified in a common purpose: creating innovative solutions to meet our world's water needs. Developing new technologies that will improve the way water is used, conserved, and re-used in the future is central to our work. We move, treat, analyze, and return water to the environment, and we help people use water efficiently, in their homes, buildings, factories and farms. In more than 150 countries, we have strong, long-standing relationships with customers who know us for our powerful combination of leading product brands and applications expertise, backed by a legacy of innovation.

For more information on how Xylem can help you, go to xyleminc.com

896876 . Flygt N-pump series . 1 . Master . 5 . 20180611

xylem
Let's Solve Water

Flygt is a brand of Xylem. For the latest version of this document and more information about Flygt products visit **www.flygt.com**

Appendix D

PS220 Quick Guide

PS220 Modbus TCP/IP Quick Start Guide

13.02.19 Rev.: 1.4 M&C

Innholdsfortegnelse

1. Konfigurasjon

FSCA-modulen må installeres i "Slot 2" på PS220.

Parameter	Navn	Krevd verdi	Kommentar
50.1	FBA A ENABLE	[2] Option slot 2	Aktiverer/deaktiverer kommunikasjonen mellom drive og feltbussadapter A.
74.01	START/STOP	[6] Fieldbus	Velger hvor en skal sende start/stopp-signal fra.
74.05	SPEED OVERRIDE	[6] Fieldbus	Manuell fjernstyring.
74.06	SPD OVERRIDE REF	[3] Fieldbus	Manuell fjernstyring.
51.01	FBA A type	[128] Ethernet	Denne vil være valgt som standard for FENA-modulen.
51.02	Protocol/Profile	[1] MB/TCP ABB E	Velger ABB – Enhanced som profil for Modbus.
51.03	Commrate	[0] Auto	
51.04	IP configuration	[0] Static IP	
51.05	IP address 1		Første del av valgt IP-adresse (**123**.456.789.0)
51.06	IP address 2		Andre del av valgt IP-adresse (123.**456**.789.0)
51.07	IP address 3		Tredje del av valgt IP-adresse (123.456.**789**.0)
51.08	IP address 4		Fjerde del av valgt IP-adresse (123.456.789.**0**)
51.09	Subnet CIDR	24	Velger subnet: 24 = 255.255.255.0 (Standard) For andre valg, se FENA-manual side 55.
51.22	Word order		Velger rekkefølge på ordet. Avhenger av systemet som skal lese fra drive.
51.23	Address mode	[1] Mode 1	Denne modusen muliggjør lesing/skriving av 16-bit. *Se forøvrig begrensningene knyttet til denne modusen.
51.27	FBA A par refresh	[1] Refresh	*Lagrer* alle parameterendringer utført i gruppe 50. 51. 52 og 53. Vil gå tilbake til [0] Done etter å ha blitt satt til [1] Refresh.

*51.23 Address mode [1] Mode 1: Tillater **kun** 16-bits verdier. Om det er behov for å lese/skrive med 32-bit må velges et annet alternativ enn Mode 1. Se for øvrig FENA-manual side 58.

2. Kommunikasjon med ABB Enhanced Profile

Kontrollord

Bit	Navn
0	OFF1_CONTROL
1	OFF2_CONTROL
2	OFF3_CONTROL
3	INHIBIT_OPERATION
4	RAMP_OUT_ZERO
5	RAMP_HOLD
6	RAMP_IN_ZERO
7	RESET
8…9	RESERVED
10	REMOTE_CMD
11	EXT_CTRL_LOC
12…15	RESERVED

Registeradresse	Navn	Kommentar
400001	ABB Drives Profile Control	Kontrollord (W)
400002	ABB Drives Profile Reference 1	Referanse 1 (W)
400003	ABB Drives Profile Reference 2	Referanse 2 (W)
400004…400015	Data Out 1…12	Data Out (W)
400051	ABB Drives Profile Status	Statusord (R)
400052	ABB Drives Profile Actual 1	Prosessverdi 1 (R)
400053	ABB Drives Profile Actual 2	Prosessverdi 2 (R)
400054…400065	Data In 1…12	Data In (R)

Statusord

Bit	Navn
0	RDY_ON
1	RDY_RUN
2	RDY_REF
3	TRIPPED
4	OFF_2_STA
5	OFF_3_STA
6	SWC_ON_INHIB
7	ALARM
8	AT_SETPOINT
9	REMOTE
10	ABOVE_LIMIT
11	EXT_CTRL_LOC
12	EXT_RUN_ENABLE
13…14	RESERVED
15	FBA_ERROR

For kontrollordet brukes hovedsakelig integer-verdiene
STOPP/RDY – 1150
START – 1151
SPEED OVERRIDE – 1295

For vanlig oppsett brukes følgende:
Kontrollord – 40001
Referanse 1 – 40002
Data out – Se tabell
Statusord – 40051
Prosessverdi 1 – 40052
Data in – Se tabell

Pump Statusord

BIT	Navn
0	REMOTE
1	M/F READY
2	RUNNING
3	FAULT
4	MS SWITCH REQUEST
5	RESERVED
6	RESERVED
7	SLEEP
8	START ISSUED
9	SPEED OVERRIDE
10	RESERVED
11	RESERVED
12	START DELAY

Parameter **6.200** er **Pump statusword** og kan brukes for å hente ut tilbakemelding om blant annet drift og speed override, som vist i tabellen til venstre. I tillegg har man parameter **6.50**, **User Status Word**, hvor man kan sette opp eget statusord. **Alarmord 1**, **6.203**. **Alarmord 2**, **6.204**.
Alle parametre i PS220 kan leses med bruk av adresseringsmetoden på neste side. Det anbefales å bruke Data In/Out så lenge det lar seg gjøre.

Alarmord 1

BIT	Navn
0	Basic pump protection warning
1	Sleep warning
2	Dry run warning
3	Min flow warning
4	Runout flow warning
5	Tuning warning
6	VFD warning
7	Cond 1 warning
8	Cond 1 alarm/fault
9	Cond 2 warning
10	Cond 2 alarm/fault
11	Speed override
12	TX/AI error
13	Secondary protection A warning
14	Secondary protection B warning
15	Run disabled

Alarmord 2

BIT	Navn
0	Basic pump protection warning
1	Low demand fault
2	Dry run warning
3	Min flow warning
4	Pump clogged fault
5	Tuning warning
6	VFD fault
7	Multipump comm
10	Start delay

3. Parameteradressering

Kan velge å lese/skrive direkte til parametere, eller mappe dem via. «Data In/Out» og adressere dem der fra.
Data In brukes for parametere som skal leses og «Data Out» brukes for parametere som skal skrives til. Man må alltid oppdatere via. parameter «51.27 FBA A par refresh» om det er utført endringer i parametergruppene 50 t.o.m. 53. Både 5- og 6-bits adressering er støttet. Det er som regel fordelaktig å bruke «Data In/Out» ettersom dette er sammenhengende registre som gjør lesing og skriving med effektivt. Det anbefales derfor i første omgang å bruke disse, for så å legge til direkte parameteradressering om man skulle trenge flere parametere ut over de 24 som blir tilgjengelige på «Data In/Out».

Data In/Out
Når en parameter er knyttet til «Data Out» vil den alltid overskrive andre skrivekilder, som f.eks. fra keypad eller ved direkte parameteradressering. Det er derfor viktig å holde oversikt på hvilke parametere som ligger i «Data Out», og kun bruke «Data Out» om man først har knyttet opp en parameter her. Det samme gjelder «Data In». Om man har knyttet opp en parameter til en «Data In» må man lese av denne og ikke av direkte parameteradresse.

Direkte parameteradressering
For å adressere en parameter direkte må man bruke følgende formel:

400000 + 256 x parametergruppe + indeks

Eksempler
Data Out: Man mapper parameter «12.18 AI1 max» gjennom Data Out 1 ved å gå inn på parametergruppe 53. Så velger man Data Out, og deretter «Other». Her kan man velge hvilken parameter man ønsker å hente ut for skriving. Man må da bruke registeradressen for Data Out 1 (400004), og ikke 403090.

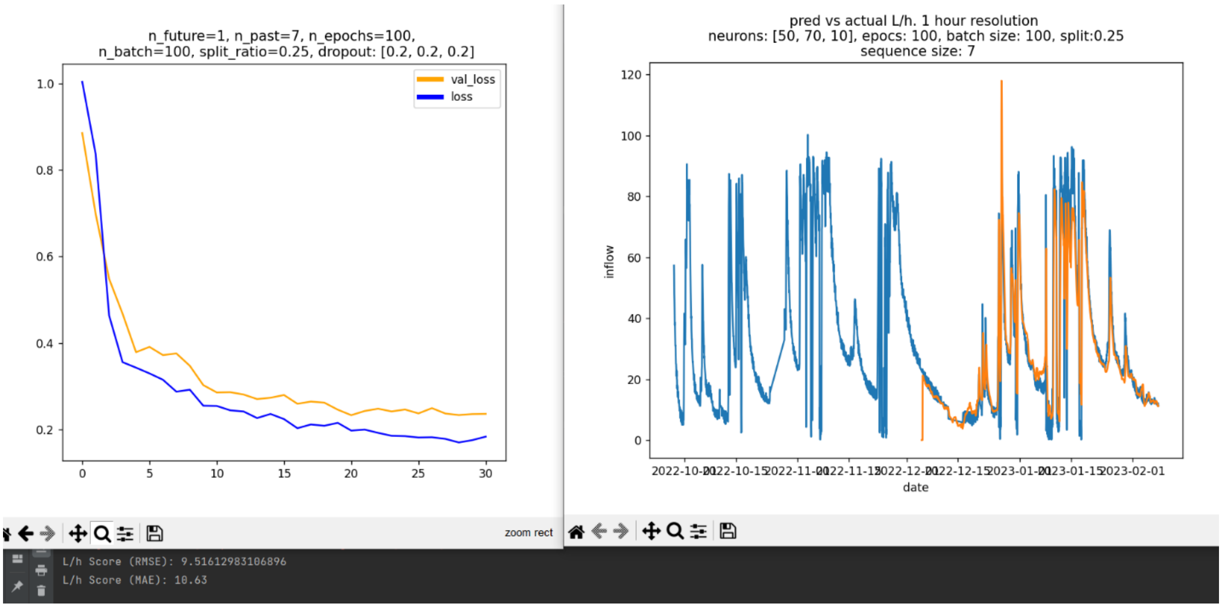
Direkte: For å skrive til parameter «12.18 AI1 max» bruker man formelen (400000 + 256 x 12 + 18) og skriver da til registeradresse 403090.

Appendix E

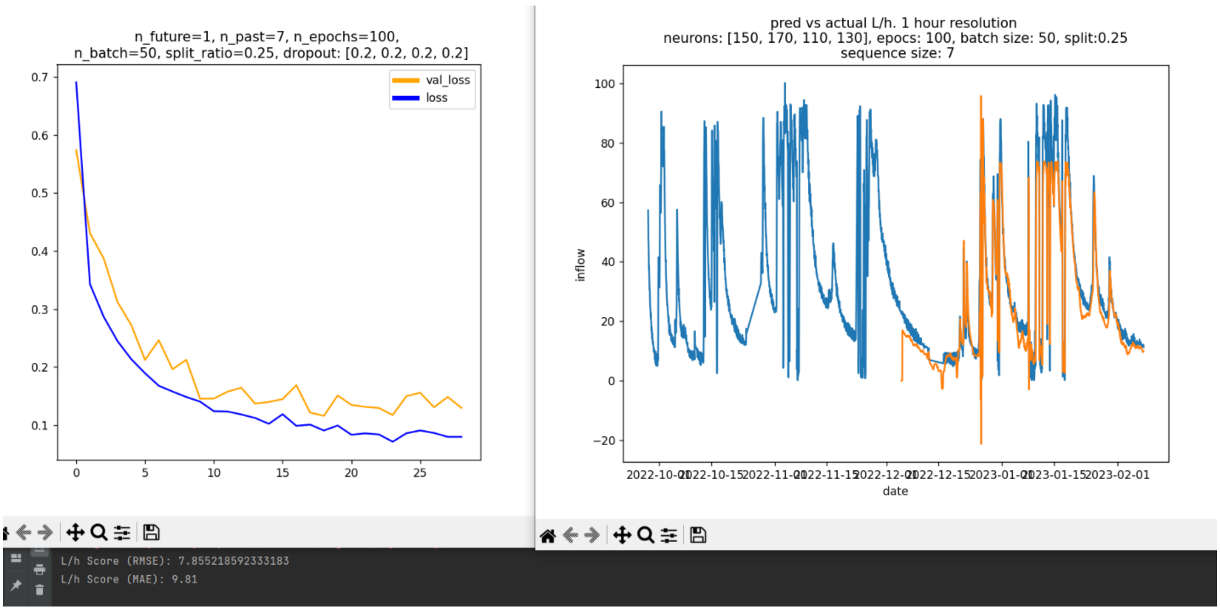
Tuning Transformers and LSTM results

LSTM Tuning

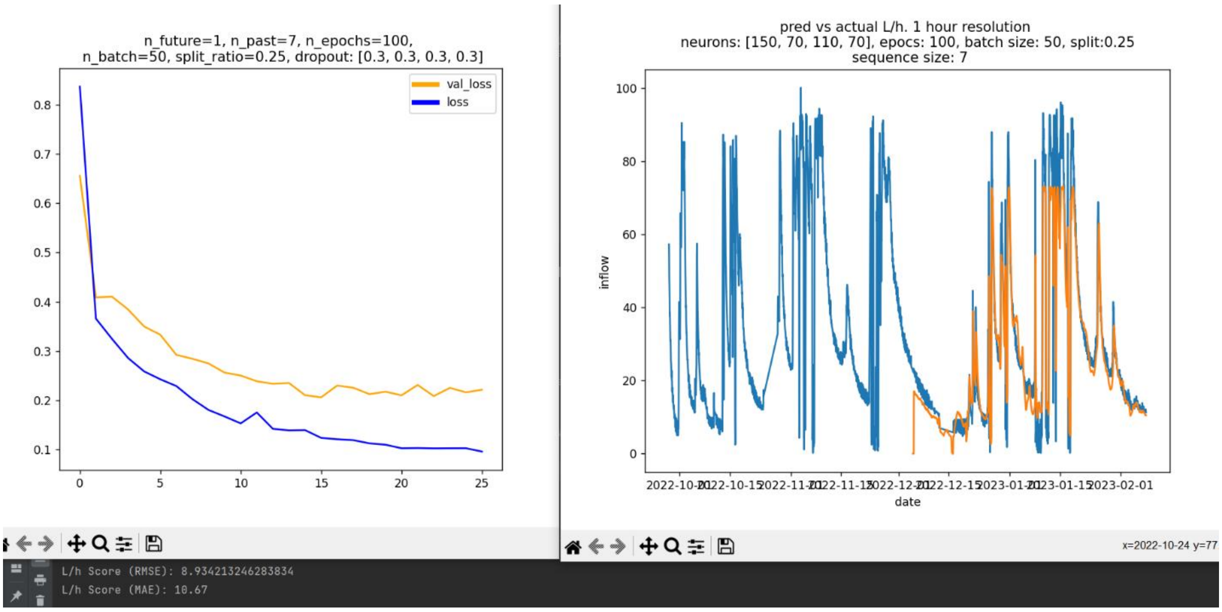
Underfitting:



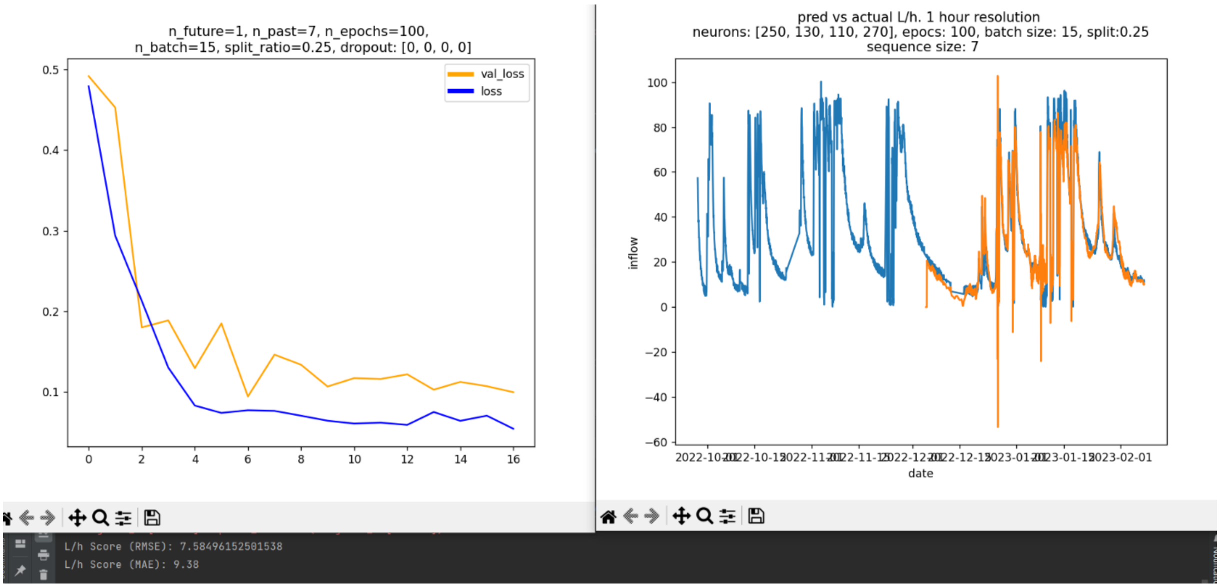
Less underfitting:



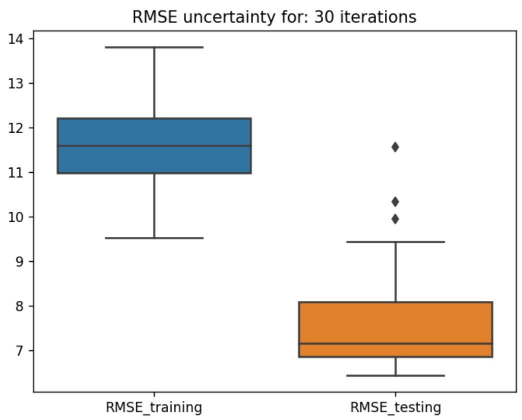
Reduced noise:

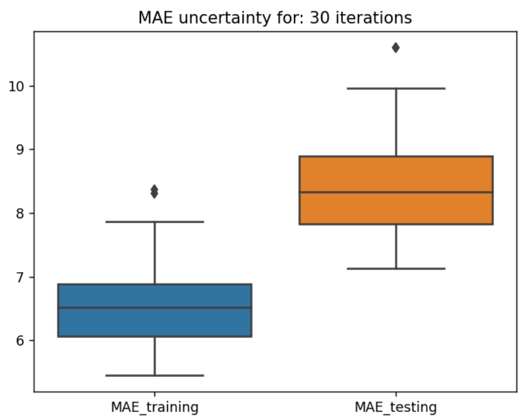


Best fit:



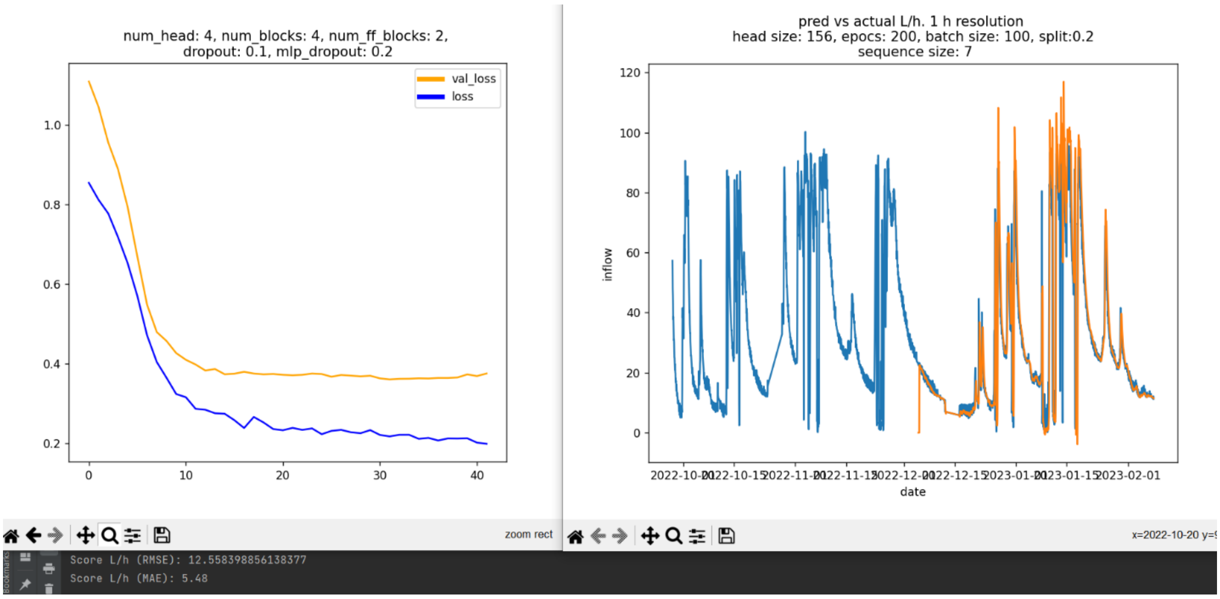
Uncertainty for 30 iterations of training and testing:



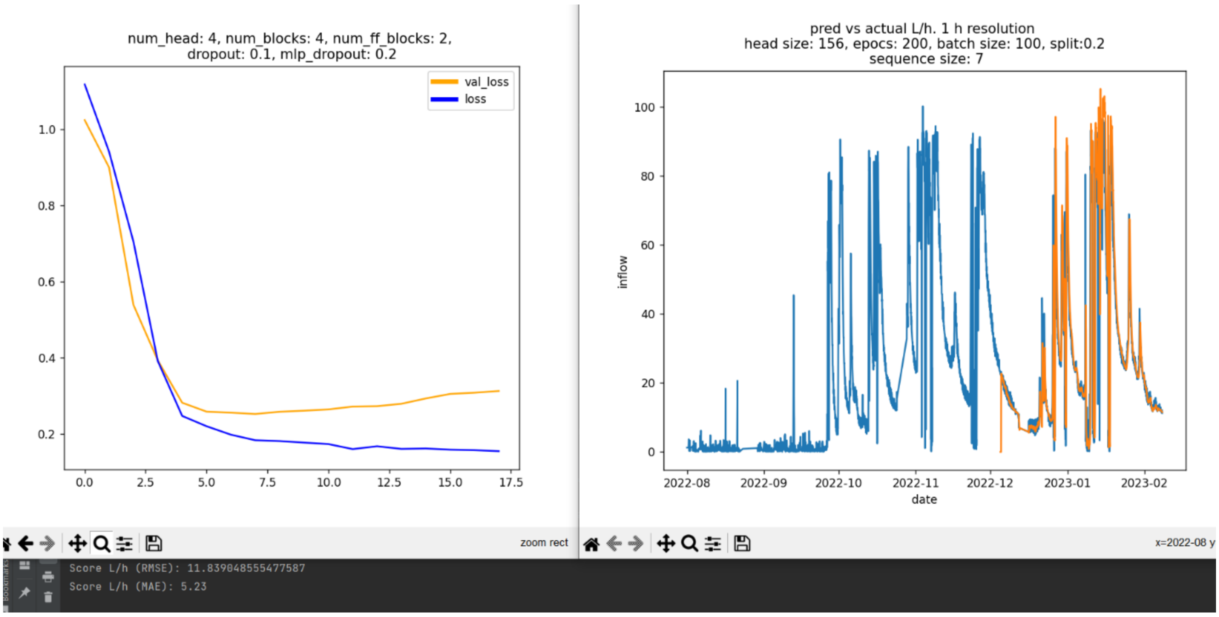


Transformers tuning

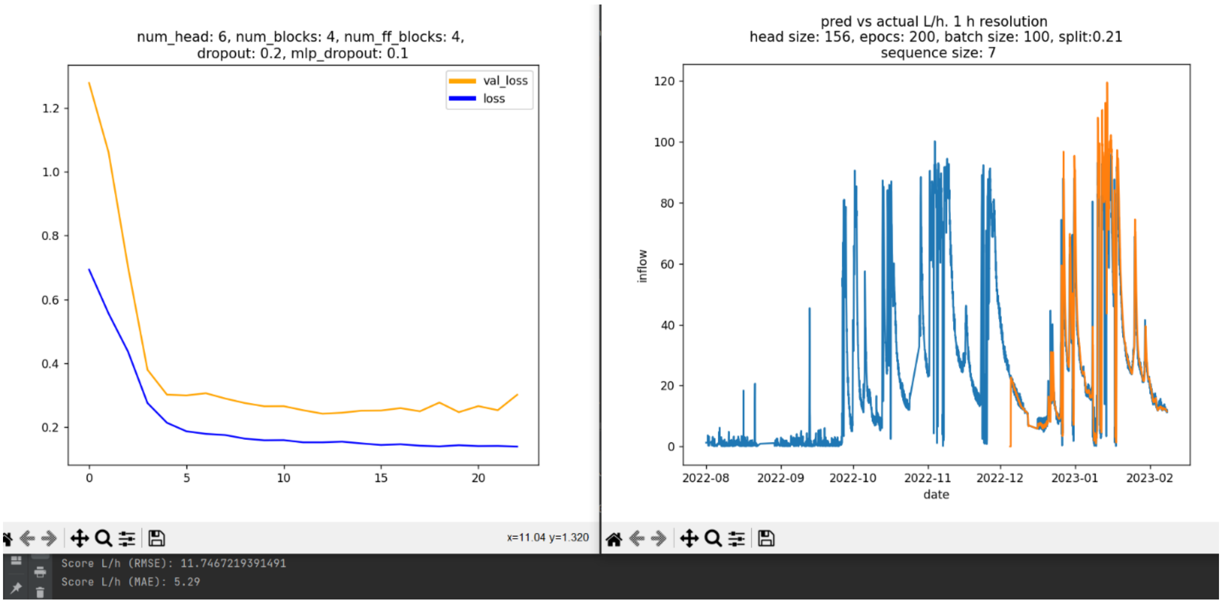
Underfitting:



Reduced underfitting (more data):



Best fit:



Uncertainty for 30 iterations of training and testing: