Challenges in connecting a wastewater treatment plant to a machine learning platform

Christian Wallin[1,2] Eva Nordlander[1]

[1]Future Energy Center, Mälardalen University, Sweden, {eva.nordlander}@mdh.se
[2]ABB AB, Sweden, christian.wallin@se.abb.com

Abstract

Treatment of wastewater is fundamental to protect the environment and to ensure a healthy water supply. Higher demands are put on the treatment of the effluent from wastewater treatment plants (WWTP) to reduce more pollutants as well as remove pharmaceutical residues. To be able to deliver better water quality monitoring and control is of importance but wastewater treatment is far behind many industrial processes when it comes to automation. Digital twins and machine learning could offer many benefits but not much work has been done in this field concerning wastewater treatment. How do you move from an existing traditional process automation system to an integrated machine learning platform?

This paper investigates the challenges of implementing an integrated machine learning platform for a wastewater treatment plant. The paper is based on experience from a project where a number of different processes, including a WWTP where integrated into a machine learning platform in an online cloud environment. In this paper we focus on the integration of the WWTP. On the platform a model is run in real-time using process data. Machine learning algorithms are used to treat the process data and for sensor fault detection. The challenges and considerations are many, such as cyber-security when it comes to data access and data transfer and how to convert the process data to a format that can be used by the model.

Multiple defining choices must be made along the way that can have a major impact on the final platform functionality. It is important not only to evaluate these choices but also to have enough knowledge and jurisdiction to make both the right decisions and to also make them in time. Many projects run out of time and/or money for different reasons and strategies will be discussed for how to mitigate risk factors.

Keywords: Wastewater Treatment, Machine Learning, Cloud Environment

1 Introduction

Many wastewater treatment plants today have sensors to measure water quality and also to control parts of the plants. A common control is the aeration of the biological treatment to send enough air in to the water to fulfill a set-point of the level of oxygen in the water. Other common measurements is the level of ammonia and nitrate in the water and some plants also use these measurements as control parameters. The wastewater treatment plant of Västerås, that was used as reference plant in this project, have all of the above mentioned measurements as well as water temperature, water flow-rate and phosphor concentration sensors. Data from these sensors have been stored in a historical database since at least 5 years back to produce graphs for operators of the plant and to create reports to authorities. The goal was to do more with all this data and use modern machine learning technologies to achieve better control and monitoring of the water and the treatment plant.

2 Method

Historical data at the WWTP were stored in a process database. A model had also been developed of the WWTP based on the BSM2G Matlab Simulink simulation model (Vrecko et al., 2006) but for other purposes than was intended within this project. The model was used for online simulations of the plant and needed to be adapted to connect to the data from sensors and analysis. It also needed to be stable and have a reasonable simulation time. In the end a BSM2 model (Jeppsson et al., 2007) adapted for the plant was used, but the ASM1 (Henze et al., 1987) and BSM2G models were also tested. Since sensor data would be used for the models, sensor fault detection was also of importance and multiple strategies were evaluated. In addition we wanted to add Model Predictive Control (MPC) as extra functionality to optimize the process.

2.1 Using a Cloud Platform

To ensure the existing data was not altered by mistake and to avoid unauthorized access the data needed for new models and analysis was cloned from the existing database. The cloned data and models could either be hosted on-site or using an online cloud platform. There are benefits with self-hosting since data never leaves the site but there are also drawbacks in terms of accessibly and scaling as well as needing new hardware to be purchased and maintained. For this project a cloud-based solution was selected. With a cloud-based solution initial costs for investing in infrastructure could be avoided and it made it possible to get started before the full benefit and return value of an investment could determined.

2.2 Models

Throughout the project numerous different models and model interfaces have been used and evaluated. To build a scale-able platform that is able to support effortless integration of as many different models as possible it is important to choose a platform that is not built only to support a specific model vendor. As the project was founded by the EU Research and Innovation program ”Horizon 2020” all models were required to be open source models. This does not limit the use of commercial software such as MATLAB but requires the final models created to be exported to an open-source format. One of the benefits with open-source models is that they allow for better sharing of experiences between organisations. Another benefit is that the open-source models are often less platform-dependent which also gives more options to the final user.

The models used in this project was built using MATLAB Simulink. MATLAB models can be run outside of MATLAB using runtime dll-file but it still requires a MATLAB runtime license. This could be a working solution for many but with our requirement to be fully open-source it was not possible to use this in our project. Another possibility is to convert an existing model to an open-source model, but the resulting model would essentially be a new model which would require time to build and test it. In this project it was solved by using the Simulink model but compile it is an open-source model using the Modelon "FMI TOOLBOX FOR MATLAB/SIMULINK" (Modelon AB, 2021). This toolbox makes it possible to compile a Simulink model as a Functional Mock-up Unit (FMU) which creates a file with the simulation model based on the Functional Mock-up Interface (FMI) which is an open standard for simulation models.

2.3 Data

Data was extracted in two ways: Initially historical data of interest was extracted and then followed by a continuous extraction of new data. The data was extracted using Excel-scripts and the two different types of extractions had to be treated separately. As Excel has limitations in how much data that can be stored in one file and the amount of data from certain sensors that had been storing data for several years where extensive. The historical data had to be extracted for one sensor at a time and in some cases also in batches of as little as a few months while the continuous extraction of new for all sensors could fit in a single file. Even though the time-resolution varied among the different sensors, the extracted files was created using batches of data based on one-minute averages for each sensor. Intervals of 15-minutes were used between extractions, meaning each extracted file contained 15 lines, one for each minute, of the data for each sensor for the last 15-minute period.

The chosen time resolution of 1 minute and 15 minutes between extractions impacted the amount of storage space required to store the extracted data as well as what models can be run using the data. The time resolution was chosen to work well with all models used and could be increased if too much storage would be consumed. 15 minutes between each batch of new data was also considered enough since the models would not be used for direct control. A delay of up to 15 minutes would be sufficient to decide the set-points of faster controllers or to detect sensor failing over time. If the models should instead be used for direct control of for example valves or pumps the time resolution and time between extractions would need to be reconsidered.

2.4 Integration

The models need to be connected to the data storage. Between the storage of data and the model the data needs to be checked as well as translated into a form that can be used as input by the model. The model results also need to be stored as well as the model state in case of state-based models. An alternative to storing the model state is to have a default state and an initialization period before each batch of new data feed into the model.

The data and model integration framework used to handle management of data between the data-store and the models used in this project was Node-RED (JS Foundation, 2021). In addition to data management the framework also contains the functionality to create websites to visualize data and model output as well as providing the possibility to get user input to models. Having such a framework is essential to prepare raw data from the database storage in a suitable input format for each of the models and to sort through the model output to format and store essential model output. We decided to have a full separation of stored raw data from the treatment plant and the data stored from model output to never risk losing track of the origin of the stored data.

Some of the models that were used required data from chemical analysis as well as sensor data for model input. The chemical analyses are made at the plant manually about once every week. The data from the analysis were stored separately from the sensor data. Each of the analysed values was assumed to remain the same until a new analysis was made. The load on the plant varies over the day but since the analysis are made as collected samples each week the values does not reflect daily variations. Instead to simulate daily variations, a normal daily variation curve was used. The resolution was 15 minutes and the average for each day was the chemical analysis value for the week the day belonged to.

2.5 Output and presentation

One of the most important parts for a project like this to be considered successful is to make good use of the produced outputs. The performance of the models does not matter much if the outputs cannot be visualized to an end user to make better or more informed decisions or to optimize actions as plant control or maintenance planning. Writing information back to the control system can be a real chal-

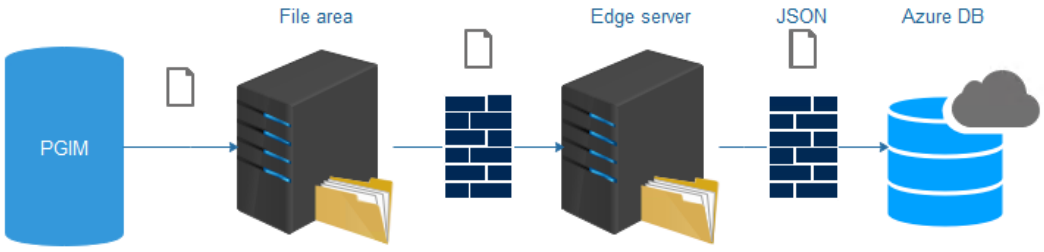

Figure 1. File transfer from historian database to cloud storage

lenge from a security perspective and it should be handled with care. Even if the write-back of data to the system is secured, the models also need to be thoroughly tested before actual control actions can be taken based on the model results. It would also be advisable to have model input and output validation to make sure that all variables are within reasonable levels. If any of the variables would deviate a default fallback option ignoring the models could be used as a fail-safe.

In this project nothing was written back to the system, results were instead presented on a webpage for operators to read to take actions, with a security air-gap between the new cloud models and the control system.

3 Results

3.1 Implemented models

Several physical simulation models both of generic and plant specific tuned models have been integrated in the platform. All physical plant simulation models used originates form MATLAB Simulink models that was wrapped as Function Mockup Units (FMU:s) so they could be run in any environment without the requirement of a Simulink runtime. First the generic benchmark model BSM1 (Alex et al., 2008) got implemented as a proof of concept to validate that the FMU models would produce the same results as when they were run inside Simulink which was successful. Following that the more complex models BSM2 (Jeppsson et al., 2007) and BSM2G (Vrecko et al., 2006) modified and tuned as digital twins of the Västerås wastewater treatment plant named Kungsängsverket were also integrated and tested in the platform.

Some live sensors have been connected to these models and the current ongoing task is to create the required influent datapoints from live sensors together with information gather from recent lab measurements. This is needed as the models requires a large set of input parameters which is not directly measured with live sensors as the contamination in the incoming water make continuous measurements very challenging.

Models for Model Predictive Control (MPC) was also developed in Matlab Simulink using the MPC Toolbox and like the full plant model they were exported to FMU. Several different FMU:s have been tested to run the BSM1 and the BSM2 models. For security reasons no write-back to plant is currently allowed and would have to be solved in future project which make closed-loop MPC impossible but the possibility to manually read MPC output and use this as manual input to the control system is still useful whilst not as manageable as a closed-loop solution. The output from the MPC in all cases have been supervisory control as in set-points for existing PI controllers. The main improvement observed is in the possibility to lower the amount of aeration during lower plant load to save energy but a small improvement to water quality has also been seen in some scenarios.

To compare and analyze the correlation between multiple oxygen and air-flow sensors a Bayesian Network model was developed using the Hugin tool (HUGIN EXPERT A/S, 2021). However, this model required a lot of training data to provide accurate results which was not available in the quantities needed. Instead the fault detection was moved to Python models. A model to classify normal data from disturbance data from event of sensor cleaning was developed and tested with good result using historical data. This was not suitable to classify new datapoints from continuous sensor measurements and instead focus was moved an Auto-regressive Integrated Moving Average Model (ARIMA) model to look at multiple airflow sensors together with oxygen sensor readings to determine normal conditions and detect deviations from this.

3.2 Data Extraction

A file is cyclically exported from PGIM database and put in an internal file area and a scheduled job is moving this to an area where an edge server can consume this file. This flow is illustrated in Figure 1.

The edge server is a virtual machine placed inside Mälarenergi network which runs on Ubuntu operating system. Edge server ingest data from a share drive located in Mälarenergi network and then performing data processing, anonymization and transfer to JavaScript Object Notation (JSON) format, and sending to cloud will be managed by scheduled data collecting modules. JSON files are processed directly to database by scripts that runs in Azure functions. Functions are triggered on incoming files at Azure BLOB Storage. Data is continuously transferred from history database to Azure Cloud MS SQL Server and

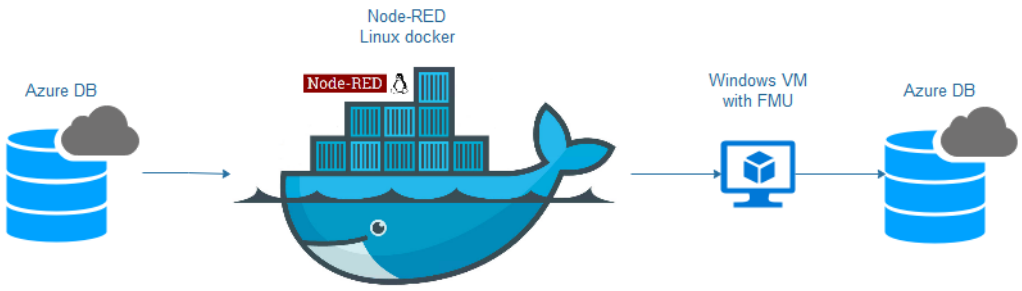

Figure 2. Data flow in cloud environment

the data is available in the cloud storage less than 5 minutes after the data has been written from the control system to the history database.

3.3 Data and Model Integration

Using a variety of Microsoft Azure resources; virtual machines, storage, databases and application-services, it is possible to use the data originating from the customer site database to call i.e. a FMU-model and store back the model results. The results could later be used to control the plant or give suggestions to plant staff of how to optimize the process of running the plant or optimize maintenance. Data flow from database through data preparation stages to model execution, retrieving model results and finally writing results back to a database is illustrated in Figure 2.

The process of running the models inside the cloud platform was built in the flow-based programming tool named Node-RED which is running inside a Linux Docker container in the MS Azure cloud. Inside Node-RED a flow is created which retrieves the necessary data from the MS Azure Database pre-process the data to a format suitable for the intended model to be run and send a http request to a Windows virtual machine also running in the MS Azure environment. This virtual machine, which has the necessary run-times required to run the models, run a web-service that listens for request on different ports for different models to know what model should run with what data. Once data is sent to a web-socket a program execute the corresponding model based and feed it the input data sent to the virtual machine. Once the model has finished running the results are sent back the Node-RED flow which then take care of the model output and make it possible to store it back in the cloud database. An example of how such a Node-RED data-flow is created within the Node-RED user interface has been illustrated in Figure 3. Here each node i.e. SQL code to retrieve or store data to a database or JavaScript code to prepare data for model execution.

4 Discussions

Making sensor data available to be processed and accessed by modern machine learning algorithms and simulation models is the start of a new era of process control. There are numerous studies simulating the WWTP and different possible advanced control strategies. This implementation and evaluation framework is a step to bring advanced control closer toward full-scale testing and final implementation. Having a framework in place where data can very easy be made accessible to new model developers also reduce both the complexity and the effort required in any future project looking to further improve the process. Adding feed-forward and feedback MPC to the existing aeration control strategy give both the possibility to reduce effluent violations and to reduce the energy used during periods when the aeration demand is lower.

When adding more algorithms and models to a system the quality of input data become increasingly important. Sensor maintenance already requires a lot of work for plant owners and maintenance personnel. When this maintenance can be planned in according to where and when a sensor is in most need of maintenance such as cleaning with addition of added sensor diagnosis tools both the cost of maintenance and the accuracy of sensor data could be improved. Today every sensor is cleaned in intervals based on experience but with the possibility of analyzing actual sensor performance it could be both better and without requiring experienced staff to plan the work.

Digital Twin(s) give endless comparison opportunities both for process control and sensor maintenance. Future control strategies should be evaluated by numerous clones with different settings and models can also be made accessible for experienced operators to try new strategies without the risk of affecting the process. The platform can also be a valuable tool for new personnel to train, learn and experiment with the process in a safe environment. It would also be possible to record special scenarios to be replayed either to try strategies or to teach already know strategies.

The biggest achievement is the implementation of the platform itself and the endless possibilities for further development of new models and possible plant control. It has been shown that it is possible to achieve better water quality and reduce energy consumption by applying MPC to the aeration process of a WWTP. Having a full plant model implemented in a cloud environment connected to live data, makes it possible to evaluate strategies in a close to live environment which further increase the possibility to develop and test new strategies.

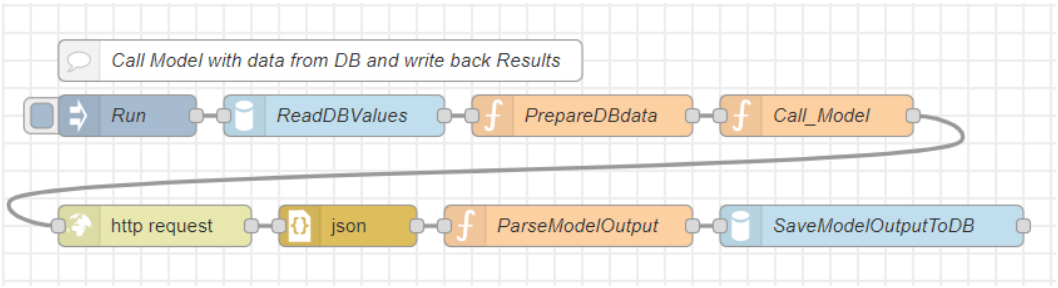

Figure 3. Example of Node-RED data flow editor

Numerous papers and research project already exit outside of this project at a theoretical level and having this implementation and evaluation framework is a step to take the theories toward further testing and finally implementing actual control of treatment plants using new technologies.

With many models to integrate, strategies to evaluate and a complex platform it is important to start with getting the simplest model (or simply just present data) to fully work as a proof of concept before moving to more complex tasks. Hopefully the information within this paper can serve as a guide for others to more easily accomplish this.

5 Conclusions

In this study an integrated machine learning platform was implemented for a WWTP. Data was extracted every 15 minutes from the plant's historical database to a cloud storage. In the cloud storage data cleaning and fault detection was performed. The data was used to run models of the WWTP and create suggestions of control actions for the operators. It shows an important step towards implementation of advanced control in wastewater treatment. Further work is needed but it is possible in the future that advanced models could be run in the cloud to directly control wastewater treatment plants and optimize their performance.

6 Acknowledgment

The authors gratefully acknowledge the financial support from European Union's Horizon 2020 research and innovation program under grant agreement No 723523 through FUDIPO project (http://fudipo.eu/).

References

Jens Alex, Lorenzo Benedetti, Jb Copp, Krist Gernaey, Ulf Jeppsson, Ingmar Nopens, MN Pons, Leiv Rieger, Christian Rosen, and J-P Steyer. Benchmark simulation model no. 1 (bsm1). *Report by the IWA Taskgroup on Benchmarking of Control Strategies for WWTPs*, 01 2008.

Mogens Henze, Leslie Grady Jr, W Gujer, G. Marais, and T Matsuo. Activated sludge model no 1. *Wat Sci Technol*, 29, 01 1987.

HUGIN EXPERT A/S. Hugin. `https://www.hugin.com/`, 2021. Accessed: 2021-07-12.

Ulf Jeppsson, Marie-Noëlle Pons, Ingmar Nopens, Jens Alex, Jb Copp, Krist Gernaey, Christian Rosen, J-P Steyer, and Peter Vanrolleghem. Benchmark simulation model no 2—general protocol and exploratory case studies. *Water science and technology : a journal of the International Association on Water Pollution Research*, 56:67–78, 02 2007. doi:10.2166/wst.2007.604.

JS Foundation. Node-red. `https://nodered.org/`, 2021. Accessed: 2021-07-12.

Modelon AB. Modelon functional mock-up interface. `https://www.modelon.com/functional-mock-up-interface-fmi/`, 2021. Accessed: 2021-07-12.

Darko Vrecko, Krist Gernaey, Christian Rosen, and Ulf Jeppsson. Benchmark simulation model no 2 in matlab-simulink: Towards plant-wide wwtp control strategy evaluation. *Water science and technology : a journal of the International Association on Water Pollution Research*, 54:65–72, 02 2006. doi:10.2166/wst.2006.773.