SCOREwater

Smart City Observatories implement REsilient Water Management

DELIVERABLE D2.7

FINAL VERSION OF STREAMLINED MODEL EVALUATION ENVIRONMENT

Dissemination level	Public
Type	Report
Issued by	IVL
Contributing project partners	IVL, EURCAT
Author(s)	Wilhelmsson, J., Lindblom, E.U.
Reviewed by	Rubión, E., de Rover, S., Samulesson, O.
Keywords	Model, Evaluation, Performance criteria
Number of pages	42
Number of annexes	1
Date:	2021-10-28
Version:	V 1
Deliverable number	D2.7
Work Package number:	WP 2
Status:	Delivered
Approved by coordinator (IVL)	2021-10-28

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

Copyright notices

The information contained in this document represents the views of SCOREwater members as of the date they are published. The SCOREwater consortium does not guarantee that any information contained herein is error-free, or up to date, nor makes warranties, express, implied, or statutory, by publishing this document. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The document reflects only the author’s views and the European Union is not liable for any use that may be made of the information contained therein.

REVISION HISTORY

Version	Reason for changes	Name	Date
1	Original release to EU	Jens Wilhelmsson and Erik Lindblom, IVL	2021-10-28

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

CONTENT

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

LIST OF FIGURES

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

LIST OF TABLES

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

ABBREVIATIONS

Abbreviation	Definition
AI	Artificial Intelligence
AUC	Area Under Curve
FN	False Negative
FP	False Positive
ICT	Information and Communications Technology
IoT	Internet of Things
KRI	Key Reference Indicator
MAE	Mean Average Error
MARE	Mean Absolute Relative Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
PI	Persistence Index
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
SDG	Sustainable Development Goals
SME	Small and Medium-sized Enterprise
TN	True Negative
TP	True Positive
TTF	Time To Flooding

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

PROJECT ABSTRACT

SCOREwater focuses on enhancing the resilience of cities against climate change and urbanization by enabling a water smart society that fulfils SDGs 3, 6, 11, 12 and 13 and secures future ecosystem services. We introduce digital services to improve management of wastewater, stormwater and flooding events. These services are provided by an adaptive digital platform, developed and verified by relevant stakeholders (communities, municipalities, businesses, and civil society) in iterative collaboration with developers, thus tailoring to stakeholders' needs. Existing technical platforms and services (e.g. FIWARE, CKAN) are extended to the water domain by integrating relevant standards, ontologies and vocabularies, and provide an interoperable open-source platform for smart water management. Emerging digital technologies such as IoT, Artificial Intelligence, and Big Data is used to provide accurate real-time predictions and refined information.

We implement three large-scale, cross-cutting innovation demonstrators and enable transfer and upscale by providing harmonized data and services. We initiate a new domain "sewage sociology" mining biomarkers of community-wide lifestyle habits from sewage. We develop new water monitoring techniques and data-adaptive storm water treatment and apply to water resource protection and legal compliance for construction projects. We enhance resilience against flooding by sensing and hydrological modelling coupled to urban water engineering. We will identify best practices for developing and using the digital services, thus addressing water stakeholders beyond the project partners. The project will also develop technologies to increase public engagement in water management.

Moreover, SCOREwater will deliver an innovation ecosystem driven by the financial savings in both maintenance and operation of water systems that are offered using the SCOREwater digital services, providing new business opportunities for water and ICT SMEs.

EXECUTIVE SUMMARY

In SCOREwater a set of Machine Learning (ML) and Artificial Intelligence (AI) models to build smart water infrastructure supporting urban resilience are developed. This deliverable describes tools for evaluating these models, thus acting as quality assurance along with ensuring that the models are unbiased.

Model evaluation deals with answering the difficult question “to what extent can I trust the model output?”. There is no standard way of evaluating all ML/AI models, therefore the objective of this deliverable is to present a general framework for performing a streamlined model evaluation. Along with a general framework, three examples of model evaluation are presented based on preliminary models from three cases within the project. Each case is presented along with examples of suiting key reference indicators.

Most importantly a clear definition of the model objective is mandatory, e.g., what are the questions the model tries to answer? This is important not only for evaluation but for the entire modelling process including the design and training phases (reported separately in D2.5 (Final version of data-driven models report for a water-smart society), to be delivered in April 2022).

For the evaluation specifically, the objective has a direct impact on the definition of the *target variable*, i.e., the (sometimes reformulated) model output variable(s) that is of particular interest. By comparing the modelled target variables with experimental evidence (observations) the model *residuals* are obtained. These are analyzed visually, and selected *metrics* are then used to weight them to quantify the “goodness” of the model.

The residuals considered in this deliverable result from the test data set, i.e., data that was not considered or seen during the previous design and training phases. By using the methods shown in this report, it must be assumed that the test data set is representative for the future model objectives. We cannot evaluate the models for scenarios that are not included in the test data.

Sensitivity and uncertainty analysis are identified as research areas that can increase the understanding and applicability of the AI/ML models. By calculating *Shapley values*, the significance of the model input data to the predicted response is highlighted. For the model developer, such knowledge might motivate a reformulation/training of the studied model. For the end-user, having a better understanding of how the inputs affect the outputs might increase the trust in the model. It also helps in prioritizing which future data that should be collected more frequently or with higher precision. *Monte Carlo dropout* is one method that allows the uncertainty of ML predictions to be predicted and can automatically reflect if a set of input data differ significantly from the training data. The technique has not yet been used to evaluate the SCOREwater models but is judged interesting if the wrap-up of the project is prioritized to focus on uncertainty.

The result of the deliverable is a recommended set of key reference indicators/metrics to use for model evaluation along with how to interpret the results. The results are exemplified with data from one case from each participating city in the project.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

1. INTRODUCTION

In SCOREwater a set of data-driven models to build smart water infrastructure supporting urban resilience are developed. For example, in the Barcelona case, a model to predict the sediment level in various parts of the sewer system is developed. The goal is that this can be used by the system owner to optimize maintenance measures, e.g., cleaning of pipes. In the Amersfoort case, a model to predict the risk of flooding in certain locations of the system is in the focus. This model could be used within a warning system to give an alarm when a flooding is about to happen. Also, in the Göteborg case, a model for early warning is developed. In this case the objective is to predict sudden changes in the effluent pH of a water treatment plant. This warning could guide operational staff in taking appropriate measures.

As part of the streamlined model evaluation process, it is necessary that the models provide sufficiently good results and that the end-users are aware of their limitations. Contained in this deliverable is a description of the tools that will be used to evaluate the models and provide this information. They involve plotting tools for visual performance analysis, metrics for quantification and suggested methods for sensitivity and uncertainty analysis.

2. OVERVIEW OF MODEL EVALUATION

Model evaluation deals with answering the question “to what extent can I trust the model output?”. Since all models have unique goals and challenges, there is no ideal or standard technique for evaluation that can be applied for all models (Bennett, N.D. et al., 2013). However, there are tools in the evaluation procedure that are of more general nature, which this report aims at documenting.

2.1. MODEL OBJECTIVES

The model evaluation environment of SCOREwater use as a basis each of the implemented and trained Machine Learning models $f()$, including the equations of the model and its calibrated parameter values (e.g. weights and thresholds). The models transform the input data *x* (also called features) to predictions $\hat{y}$ of interest:

$$\hat{y} = f(x) \tag{1}$$

The observations (measurements) of the predicted variable are generally denoted y. For the entire modelling process, a clear definition of the model objective is mandatory. This will highly affect the design of the ML model as explained in D2.5 (Final version of data-driven models report for a water-smart society). For the evaluation, the objective is of particular interest while deciding on a suitable reference indicator (Section 2.3).

The general objective of the SCOREwater models is to enhance the resilience of cities against climate change and urbanization. In this deliverable the following models with associated objectives were evaluated as examples of the model evaluation methodology:

- Estimate the sediment levels in specific parts of a sewer system in Barcelona to streamline maintenance efforts
- Predict the risk of flooding in specific parts of a combined sewer system in Amersfoort to enable early warning
- Predict the effluent pH of a construction site water treatment plant to enable early warning

The models are further explained in the case studies (Section 6). Several other models are being developed within the project. It should be noted that the purpose of the model evaluations in this deliverable is to present a streamlined model evaluation process. The examples from the cases are included to contextualize the methodology.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

2.2. DATA FOR PERFORMANCE EVALUATION

As a part of the ML model design phase, methods are used to divide the available data (experimental evidence) in different subsets. During the ML model development phase, historical data is typically divided in groups such as training, validation, and test data.

In the evaluation phase, focus is on the test part of the data. Especially, it is quantified how well the model predict the observations. Thus, the residuals *e* are analyzed:

$$e = y_{test} - f(x_{test}) \qquad (2)$$

If not else mentioned, data in this report refers to the test data set. It is presupposed that this is representative considering the objectives of the model and that it was not seen during the model training.

2.3. DECIDING ON THE KEY REFERENCE INDICATORS

The goal of defining key reference indicators (KRI:s) is to allow for calculating single numbers that determine how “good” a model is. With the KRI:s, the performance of various models can be compared to each other. Given the quality of the test data and the objectives of the model, KRI thresholds can also be defined to separate useful models from rejected ones. In this deliverable, the KRI:s are derived by defining *target variables* for each model and by selecting *metrics*.

The target variable represents the “prediction of interest”. Recall that while a ML model most often is trained using one (1) defined loss function, it can be used to predict several quantities. For example, a hydraulic sewer system model might be trained to minimize the deviation between observed and predicted time series data of flow rates. The end-user, on the other hand, might be interested in using the model to estimate the *target variables* “average dry weather flow” or “peak dry weather flow”. The target variable is thus reformulated with some function *g()* of the original model output variable *y* and the errors/residuals if compared to observations become:

$$e = g(y_{test}) - g(f(x_{test}) \qquad (3)$$

The definition of target variables is highly case-dependent and will be exemplified in the case studies (Chapter 6).

By the *metric*, it is selected how to weight the residuals, e.g., by the commonly applied metric root mean squared error. The applied metrics will be shown in Section 4. To ensure the unbiased estimation of predictive power of the developed models it is normally required to evaluate several metrics for each target variable.

3. INITIAL VISUAL PERFORMANCE ANALYSIS

An initial visual performance analysis should be an early activity of model evaluation. By using trend, scatter and residual plots comparing observed and modelled outputs and target variables the overall performance of the model is visualized. Any bias (over- or underpredictions) in certain regions can also be identified.

3.1. DIRECT VALUE COMPARISON

With the plots in this section, the raw observations and predictions of the target variable are visualized.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

3.1.1. TREND LINE

The most basic plot is the trend line – the values plotted over time. When looking at predictions and observations, it is often useful to plot both in the same figure. In Figure 1, the effluent pH from a water treatment plant in the Göteborg case is shown. This type of plot is useful for getting an overview of the data and results, if there are any obvious trends or differences in the predicted and the true data.

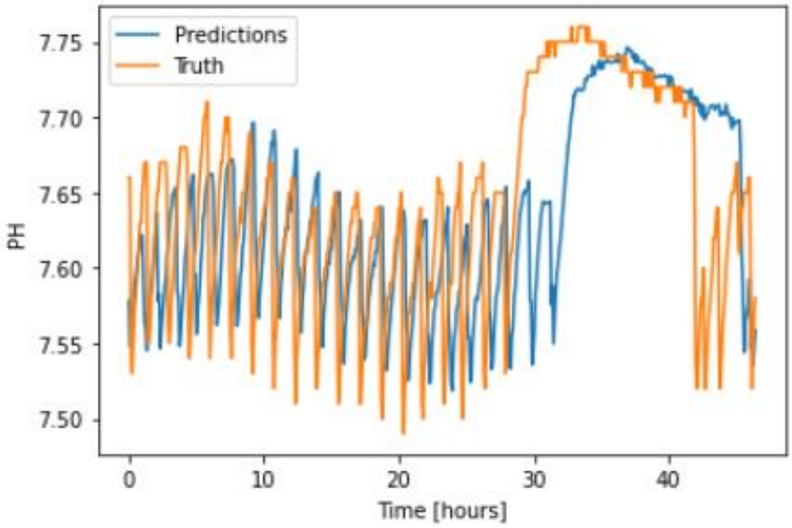

Figure 1: Trend line plot. The plotted signals are the effluent pH at a water treatment plant from the Göteburg case.

3.1.2. SCATTER PLOT

In a scatter plot, the position of the dots represents the value of two numerical variables, one for each axis. It is a convenient plot to use when comparing predictions and observations, where the optimal model would, for a scatter plot with equal axes, put all values along the line going straight from the origin to the top right corner.

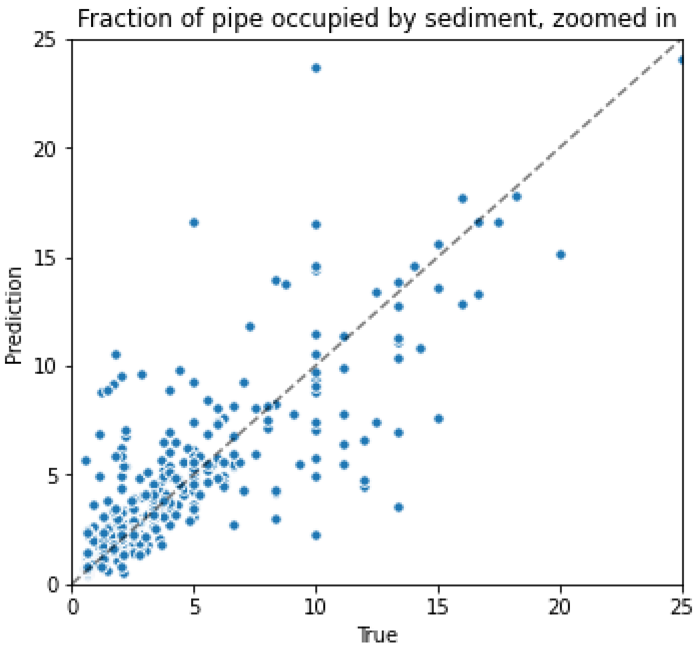

Figure 2: Example of a scatter plot. Each point represents the prediction and the true value of a data point in question. This example is from the Barcelona sediment level prediction model.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

The scatter plot is a useful tool for understanding how well the model performs on different regions of data, for example if the model is better at predicting low values or high values. It can also indicate if the model tends to predict higher or lower than the actual values, which would be indicated if the dots have a tendency above or below the dotted line.

3.1.3. RESIDUAL PLOT

The residuals are what remains when removing the predicted values from the observations. For a perfect model, the residuals should be zero. This is generally impossible, but the residual plot is a very useful tool to see whether there is more information that can be captured by the model. This is indicated if there is any kind of non-random pattern in the residual plot. In Figure 3, an example from the Göteborg case where the residuals of effluent pH in a water treatment facility is shown.

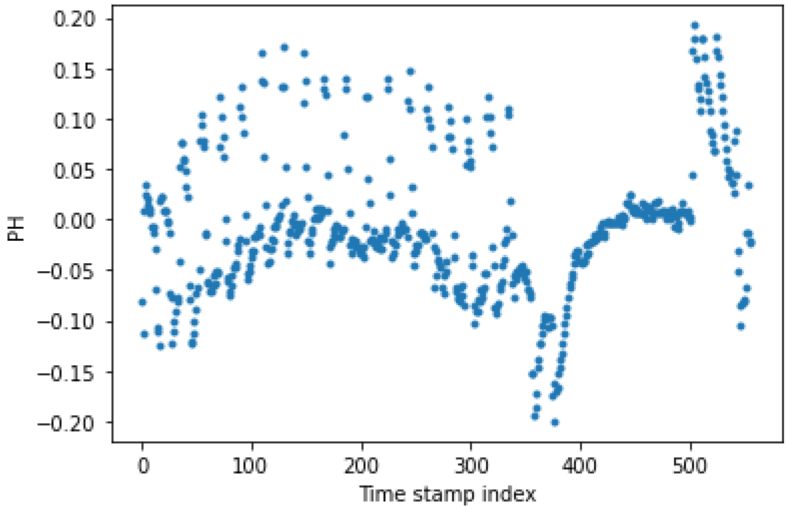

Figure 3: Example of a residual plot. For an extensive model, there should not be any visible pattern in the plot and the values should be centered around 0. This is not the case in this example from the Göteborg case where the effluent pH is predicted.

3.2. AUTOCORRELATION

To make sure that all data has been captured by the model, no information should remain in the residuals. The residuals should not show any correlation with itself – it should only be white noise remaining when correlated with a lagged copy of itself.

To verify that this is the case, the autocorrelation function of the residuals can be plotted. It shows the correlation on the vertical axis and lags on the horizontal axis. In Figure 4 is shown an example of an autocorrelation plot of the residuals from the Göteborg case where the outgoing pH of a water treatment facility is predicted.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

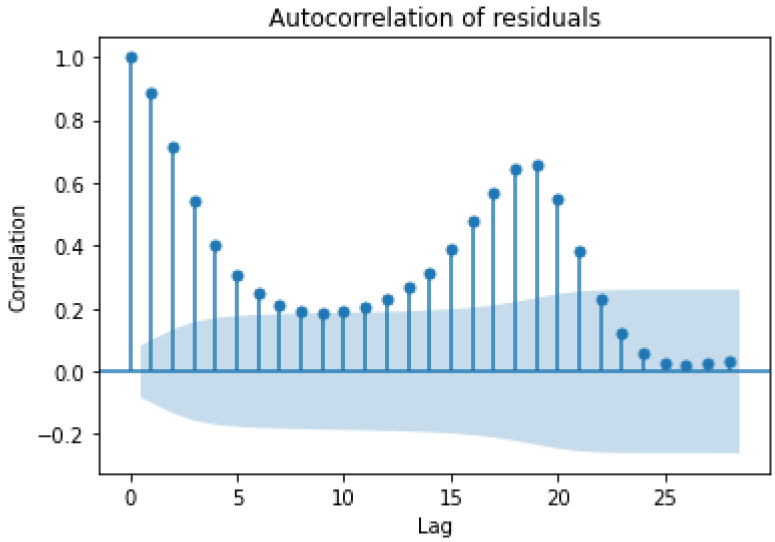

Figure 4: Example of autocorrelation plot. The blue shaded region represents the 95% confidence interval, values within this range has no significant correlation. The data in this figure is from the regression model in the Göteborg case which predicts the effluent pH in a water treatment facility.

4. METRICS

In SCOREwater, both classification and regression models are developed and applied. Classification can for example be whether a sewer is clogged or not, based on measurement data surrounding the sewer. Binary classification is a common classification type, where the number 1 in the mentioned example would represent a clogged sewer and the number 0 would represent a sewer that is not clogged.

The aim of regression is to find a function that best fits the observed data. It can be used to explain relationships between different variables along with predicting a future data point based on historical data.

A regression model can be converted into a classification model if a threshold is set for the target variable. In the Amersfoort flooding model, this was done by setting the threshold to zero, meaning that the new logical value will be 1 if the flood volume exceeds zero and 0 if the flood volume is lower than zero.

The metrics used for evaluating the two model types are listed in this section. To exemplify, observations and results of the flooding ML model (Section 6.2) is used.

4.1. METRICS FOR CLASSIFICATION MODELS

4.1.1. CONFUSION MATRIX

The confusion matrix is the collection of true positives, false positives, false negatives, and true negatives. An example of a confusion matrix can be seen in Figure 5 below.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

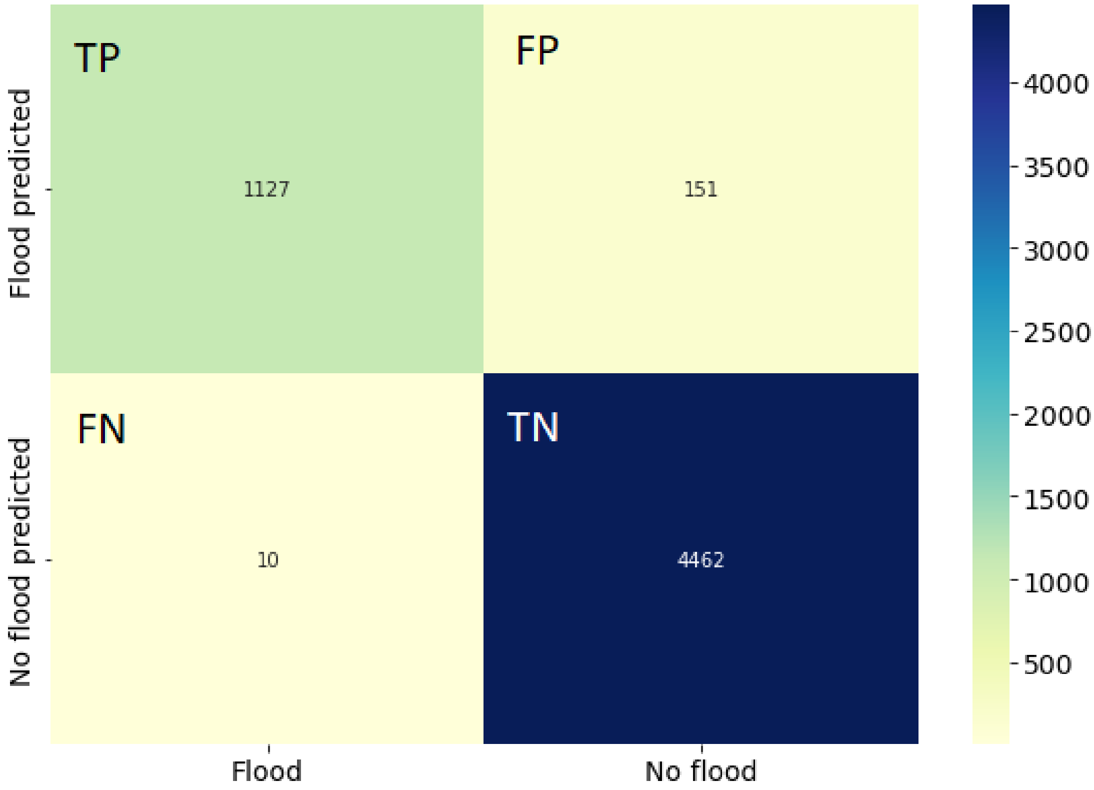

Figure 5: Confusion matrix with true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN). Classification metrics such as accuracy, precision and recall can be derived from the confusion matrix.

This confusion matrix is the result of a model which predicts flooding in several sections in the Amersfoort case. True values (true positives and true negatives) means that the model does a correct prediction. The difference between true positives and true negatives is that true positives are correct predictions of when there is a flood and true negatives is the correct prediction that there is no flood.

Similarly, the false values represent incorrect predictions, so that a false positive is when the model predicts a flood although there is no flood.

The *accuracy* of a model is the rate of correct predictions, which for the example above is calculated as

$$\frac{TP + TN}{TP + FP + FN + TN} = \frac{1127 + 4462}{1127 + 151 + 10 + 4462} = 0.972$$

Depending on the application of the model, it can be important to consider which kind of mistakes the model does when doing inaccurate classifications. For example, the purpose of the model above is to warn if a flooding is about to occur. In this scenario, it might be important that all possible floods are warned for so that the people affected by a flood can be prepared since a flood can cause big harm. So, for this case, it might be better for the model to warn for flood one time too many instead of risking to miss a flood. This is quantified as the *recall* of a model and is calculated as

$$\frac{TP}{TP + FN} = \frac{1127}{1127 + 10} = 0.99$$

An alternative to recall is the *precision* of a model, which should be valued higher if it is more important to not warn for a flood in vain than to warn often. It is calculated as

$$\frac{TP}{TP + FP} = \frac{1127}{1127 + 151} = 0.88$$

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

4.1.2. PRECISION-RECALL CURVE AND RECEIVER OPERATING CHARACTERISTIC CURVE

The classification model mentioned in the previous section only predicts either flooding or no flooding, a binary classification model. But, if instead, it would have been a model with a continuous prediction from 0 to 1, where 0 would be no flooding and 1 would be flooding, it opens for more ways of interpreting the results. The intuitive threshold for the model would be if the prediction is over 0.5, it would be interpreted as flooding and if the prediction is lower than 0.5, it would be interpreted as no flooding. But depending on if precision or recall is the most important metric, the threshold can be adjusted to fit the needs of the beneficiary of the model.

The *Precision-Recall curve* is a useful tool for finding the optimal threshold depending on the purpose of the classification model. It is created based on the probabilities mentioned above, but instead of having 0.5 as the threshold of flood or no flood, it tests all possible thresholds from 0 to 1 and calculates the precision and recall for each. In Figure 6, the color of the markers represents the different threshold used. This figure can be useful to acquire the best possible recall or precision, depending on the area of use for the model. Note that the data in this figure is based on the real flooding case from Amersfoort, but with a random noise added.

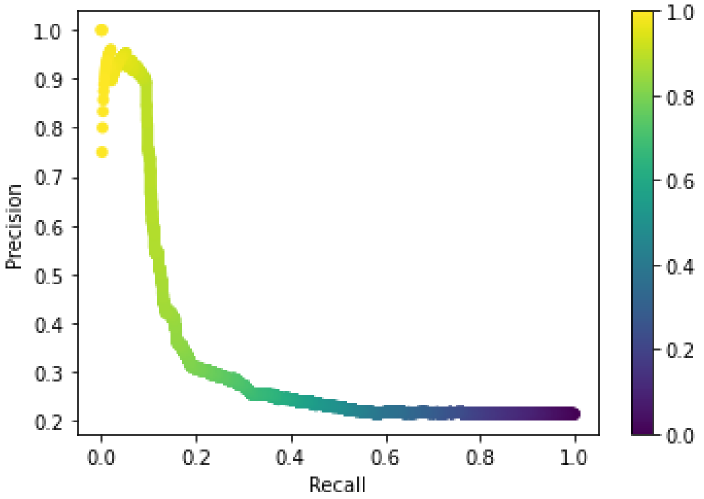

Figure 6: Precision-Recall curve. Each marker stands for a threshold between 0 and 1 which is depicted in the colorbar. For each threshold, the precision and recall are calculated so that a specific threshold can be chosen for the model if a specific rate of precision or recall are sought. Note that the data in this figure is random.

Another similar trade-off is the true positive rate (recall) compared to the false positive rate. The so-called *ROC curve* (receiver operating characteristic curve) is a plot that visualizes this trade-off. As with the Precision-Recall curve, every point in the graph is the true positive rate and false positive rate for a specific prediction threshold. See Figure 7 for an example of a ROC curve. The ROC curve is another tool for assisting in deciding upon a decision threshold in compliment to the precision-recall curve.

Based on the ROC curve, the measure *AUC* (area under the curve) can be calculated. A classifier which, independent of prediction threshold, classifies all events correctly has an AUC of 1.0 and a classifier which classifies no events correctly has an AUC of 0.0. It is a good rough measure of a classifiers performance since it is independent of the threshold. But on the other hand, for cases when it is relevant with a specific classification threshold, such as when attempting to minimize the number of false positives, it is not a relevant metric.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

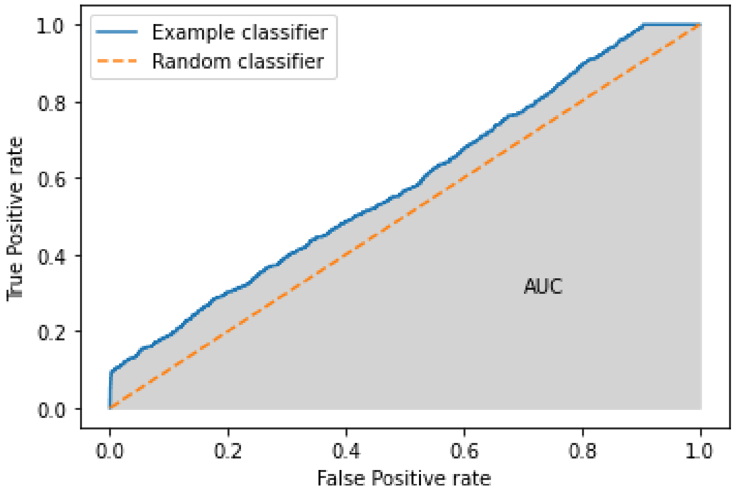

Figure 7: ROC curve (receiver operating characteristics curve) and the associated AUC (area under curve). Each point in the curve is the true positive and false positive rate of a specific classification threshold. The curve is useful when the task for example is to minimize the false positive rate.

The area under the curve can be calculated also for the Precision-Recall curve in Figure 6, giving a measure called *AUC*. While interpreting the AUC of the ROC curve is quite straightforward, since 0.5 is the baseline – a random classifier, interpreting the AUC of the Precision-Recall curve is different. The baseline for the AUC is not 0.5 but instead the fraction of positive samples which for our example case is 0.2. The area under the Precision-Recall curve in Figure 6 is 0.32. The AUC for the Precision-Recall curve is a measure to be considered when finding the positive samples is important. A perfect AUC score for the Precision-Recall curve of 1.0 means that all positive samples have been identified (perfect recall) while none of the negative samples has been marked as positive (perfect precision).

4.2. METRICS FOR REGRESSION MODELS

As a basis for quantifying the performance of regression models, the most common metrics analyze and weight the model errors or residuals (Equations 2, 3) in different ways. The metrics used for evaluating the SCOREwater models are defined and discussed below. In the shown equations, y and $\hat{y}$ refer to the variable of interest, e.g., the original model output *or* the target variable ($g(y)$ and $g(\hat{y})$). The subscript *i* is used to indicate the *i*:th observation/prediction of the test dataset containing in total *n* recordings.

4.2.1. MEAN ABSOLUTE ERROR (MAE)

A commonly used metric is the mean absolute error (MAE), it evaluates the absolute values of the residuals (Equation 5) involving that residuals are weighted equally, although positive or not:

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

$$\text{MAE}= \frac{1}{n}\sum_{i=1}^{N}|y_i - \hat{y}_i| \qquad (5)$$

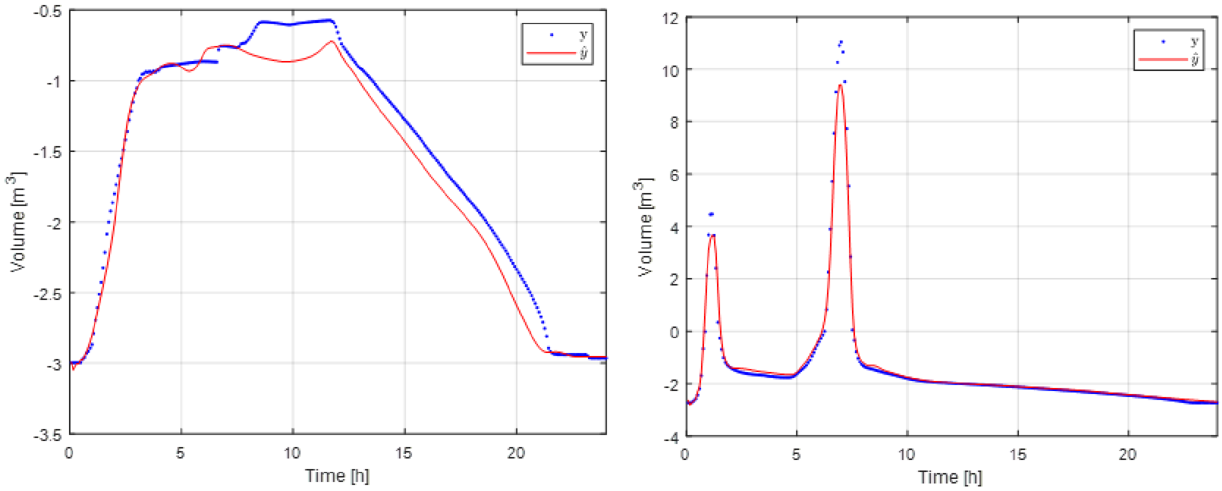

Figure 8: Two sections (#4 and #48) and events (#4 and #6) of the flooding model used to compare the metrics BIAS, MAE and RMSE.

4.2.2. ROOT MEAN SQUARED ERROR (RMSE)

The commonly used metric root mean square error (RMSE) is like MAE but more sensitive to large errors as the residuals are squared:

$$\text{RMSE}= \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}} \qquad (6)$$

RMSE can never be lower than MAE and a high RMSE/MAE quota indicates that there are outliers. Although the scenarios in Figure 8 have a similar MAE, the RMSE of the right panel is higher (0.15 m^3 vs 0.31 m^3) because of the not perfectly predicted peaks at t=1 h and t=7 h.

4.2.3. MEAN AVERAGE ERROR (BIAS)

The perhaps most simple way to evaluate a model is to calculate the mean values of all *n* errors:

$$\text{BIAS}= \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \qquad (4)$$

A high or low value of this metric (denoted BIAS) can indicate that there is a systematic difference between the observations and predictions while a value close to zero indicate a good model. A drawback is that errors can compensate for each other. In Figure 8, this is exemplified by evaluating the flooding model with data from one event and for one section separately. To the left, BIAS is positive (0.11 m^3) since the model in general predicts lower values compared to the observations. For the scenario of the right panel, BIAS is -0.01 m^3, indicating an almost perfect model. However, at t=1 h and t=7 h, periods where the model underpredicts the volume can be clearly seen.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

For the scenarios in Figure 8, the MAE of the left and right panels are similar (0.12 m^3 and 0.13 m^3). Thus, for the left panel the metric BIAS is necessary to evaluate that the model underpredicts the observations while MAE is required to detect the errors in the right panel.

4.2.4. COEFFICIENT OF DETERMINATION (R^2)

The metric R^2, sometimes called coefficient of determination or Nash-Sutcliffe model efficiency, is an example of a group performance measures that evaluate the model by comparing it with another, often simpler, model. In the case of R^2, the performance of the model is compared to simply using the mean ($\bar{y}$) of the observations:

$$\mathrm{R}^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2} \qquad (7)$$

A R^2-value of 1 indicates a perfect model while a value of 0 indicates that the model could be replaced by the mean. A negative number means that the model performs worse compared to the mean.

As an example, The R^2-metric was used to detect a few sections of the Amersfoort model that had constant observed volumes. The trained ML model, however, respond to the rain input (Figure 9). Although the MAE is close to zero (0.01 m^3), R^2 is a large negative number since the denominator in Equation 7 is almost zero.

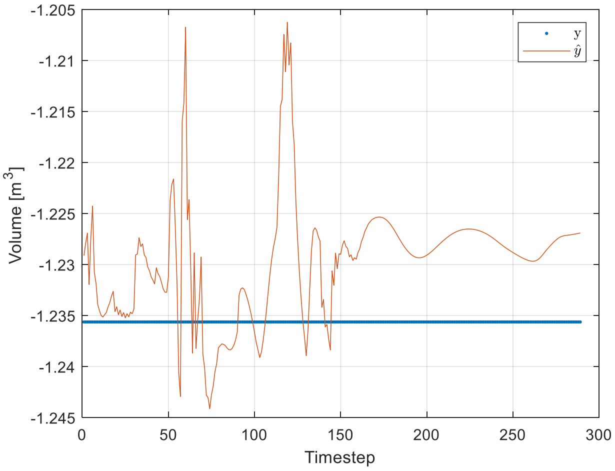

Figure 9: Example of an event with constant observations and very low R^2.

4.2.5. PERSISTENCE INDEX (PI)

This metric is like R^2, but instead of comparing the model with the mean, it is compared to a forecast using simply the previously observed value:

$$\mathrm{PI} = 1 - \frac{\frac{1}{n}\sum_{i=2}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n}\sum_{i=2}^{n}(y_i - y_{i-1})^2} \qquad (8)$$

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

This measure is thus relevant for measuring how much added value the ML model gives in a prediction context where the model is continuously updates with new observational data.

4.2.6. MEAN ABSOLUTE RELATIVE ERROR (MARE)

This metric is like MAE but replaces the residuals with the relative residuals:

$$MARE = \frac{y_i - \hat{y}_i}{y_i + \varepsilon} \qquad (9)$$

where ε is a small number to avoid division by zero. The result of this metric is that errors in the predictions of small numbers are given a large weight. For the event in Figure 10 below, the MAE (0.43 m^3) and RMSE (0.90 m^3) are relatively high, while the MARE (0.06) is low, and contradictory suggests that the model is good. For the flooding model of the Amersfoort case, it might be of interest that the model predicts values close to zero well (at this point a flooding occur) and MARE is then a relevant measure.

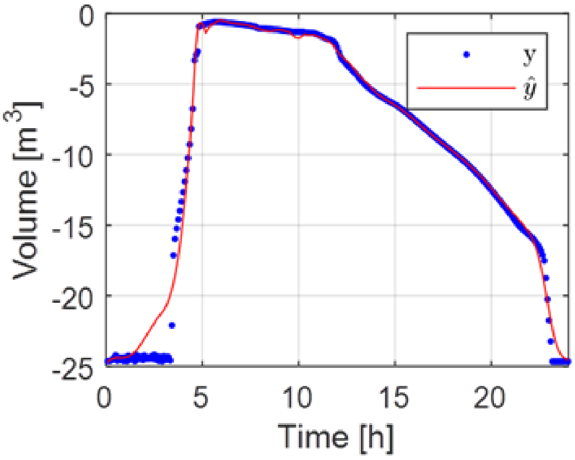

Figure 10: Example of an event with large errors but a low mean absolute relative error (MARE).

MARE has not been applied in the case studie section of this deliverable but might be useful for the remaining evaluation work within the project.

5. EXPLAINING BLACK BOX PREDICTIONS

The question of interpreting how certain a model is when giving predictions is often straight forward when looking at for example simple univariate linear regressions. A linear regression model is transparent, the coefficients tell exactly what the model base its predictions on, and the model is completely deterministic.

But when moving to Machine Learning models, the transparency of the models tend to get lost, and it gets harder to know what features that are important to the model outputs. Besides the metrics mentioned previously, it requires some extra work to understand the output of more complex Machine Learning models. Interpreting Machine Learning models is a hot topic and after researching the field, two techniques related to explaining black box predictions have been focused on.

The first is Monte Carlo dropout, which can be used to estimate uncertainty of neural network models which has so-called dropout layers integrated.

The second is calculation of Shapley values which is a game-theoretic approach to explain which features contributes the most to each single prediction, giving a picture of what features are important and which are not. It can be used with any model.

5.1. ARTIFICIAL NEURAL NETWORKS AND MONTE CARLO DROPOUT

A neural network consists of neurons and connections between them, so called weights. A simple neural network is shown in Figure 11.

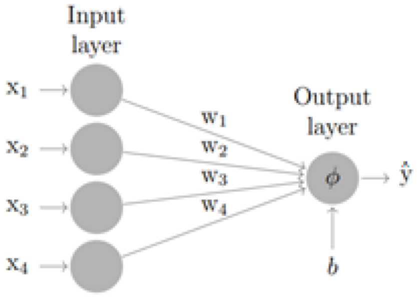

Figure 11: Example of simple neural network. The grey circles are called neurons and the connections between neurons are the weights (w_i) which is the parameters that during neural network training are modified until the network gives the desired output.

The output from this example neural network is calculated using the following formula

$$\hat{y} = \phi(\sum_{i=1}^{4} w_i \, x_i + b) \qquad (10)$$

Where $\hat{y}$ is the output, w_i is the weight, x_i is the input, b is the threshold and ϕ is the transfer function.

The goal of an artificial neural network with multiple inputs and a single output is to approximate some function *f* so that it maps the input vector *x* to the desired output category *y*. To make the neural network map the input to the desired output, the weights are updated until the output is close enough to the desired output.

When applied to advanced tasks, the number of layers and the number of neurons in each layer can be very large. When training, there is always a risk of overfitting the weights connected to the neurons to the specific data of the training data set. Overfitting causes the neural network to perform very good on the training data but poorly on new, previously unseen, data such as the test data used for evaluation. To avoid this, dropout (Srivastava et al., 2014) is commonly applied between layers. It causes a number of randomly selected neurons to be set to 0 during training which forces the neural network to be more generalized. During model training, the dropout rate (the fraction of neurons set to 0 randomly) is often derived empirically. A good rule of thumb is however to start at 0.5 (50%) for dropout layers within hidden layers and 0.2 or lower for dropout layers at input layers (Baldi, 2013).

When the trained network is subsequently used for making predictions, the dropout rate is usually set to 0, which makes the network deterministic. But if the dropout rate is set to something bigger than 0 also while making predictions, the model will become probabilistic. By doing multiple (Monte Carlo) predictions with the same input, the network will produce a distribution of outputs. The variance of the outputs will the give an indication of the certainty of the network (Gal and Ghahramani, 2015).

This method is called Monte Carlo dropout and an example of how it can be applied is seen in Figure 12. The data in the figure is flooded volume. Values below zero means that no flood is occurring. The dropout rate was arbitrarily set to 0.2 in the example.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

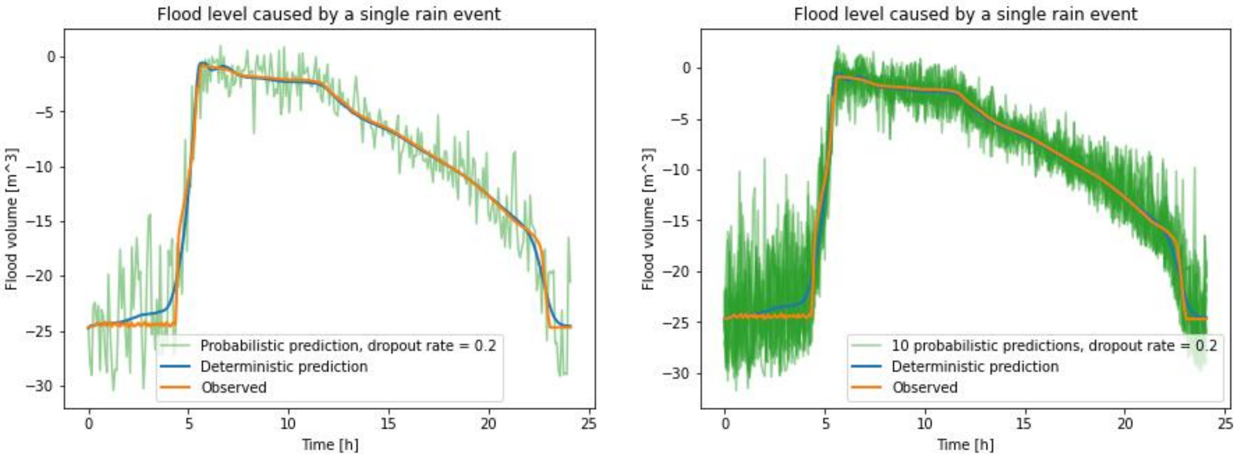

Figure 12: *Left*: Example of a single probabilistic prediction from the neural network with dropout activated. A single prediction can be used to give an indication of in what times or which value regions the highest uncertainty occur. *Right*: Same as left but with 10 consecutive predicted flood levels, giving a picture of which regions are uncertain.

The dropout rate applied during prediction is commonly chosen to correspond with the training dropout rates and, as noted by Loquercio et al. (2020), it hinders the computation of model uncertainty for networks trained without dropout. Therefore, they presented a method for how the dropout rate can be chosen in such cases. In the SCOREwater project so far, dropout (although included with rate=0 in the flooding model of Amersfoort) layers have not been activated during training. Despite the results of Figure 12, Monte Carlo dropout has not been used for model evaluation. From the work of T2.4, the technique is however considered interesting if uncertainty analysis of the SCOREwater model outputs is prioritized for the wrap-up of the project.

5.2. SHAPLEY VALUES – THE IMPORTANCE OF INPUT FEATURES

A common practice when developing Machine Learning models is to use all possible data and let the training process decide which features to value higher and which to practically ignore. This process can be useful but the many input features tend to make models act as black boxes with very little insight into how the models do the predictions and based on what information.

Shapley value calculation is a game theoretic approach that can be used to explain the output of machine learning models (Lundberg et al., 2017). From the beginning, it was used to reward people in a group individually based on their performance in a common task.

$$\varphi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(n-|S|-1)!}{n!}(v(S \cup \{i\}) - v(S)) \tag{11}$$

In equation 11, $\varphi_i(v)$ is the Shapley value for individual *i* based on the payout function *v(S)* where S is a subset of people contributing to the task. For an individual *i*, the equation calculates the marginal addition to the total value created for a subset *S* of contributors when the individual *i* has been removed. Within the subset *S*, all possible combinations of contributors are considered (Sundararajan, 2019).

In the Python package SHAP (Lundberg et al., 2017), this theory is implemented along with a couple of extensions and some practical visualization functions. When using SHAP, the individuals are replaced by input features in a machine learning model and the task is replaced by the prediction of the model. It calculates how much each feature contributes to each prediction. This results in plots such as the one in Figure 16.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

6. CASE STUDIES

Three models, one for each city, has been evaluated as examples of using the techniques described in the deliverable. In Barcelona and Gothenburg, regression models have been evaluated. In Amersfoort, the model has been evaluated both as a regression model and a classification model.

6.1. SEDIMENT LEVEL PREDICTION IN BARCELONA

6.1.1. MODEL OBJECTIVES

The objective of this specific model is to predict the degree of sediment accumulation in certain nodes of the sewer system. The predictions will guide operational staff and managers in prioritizing in what parts of the system maintenance actions will have the greatest effect. This with a decreased need for collecting real measurements. The model has been presented in Ribalta et al. (2021) and will be further described in Deliverable 2.5.

6.1.2. TARGET VARIABLE

Originally, the model was formulated to predict the height at time *t* of accumulated sediments relative to the bottom level of the sewer at a certain node *j* ($h_{j,t}$). The time *t* is constructed so that *t=0* is the current time and *t=1,2,3…* are the following. This also means that a time step not always have the same length. It was recognized that a more important target was the fill factor, i.e., the sediment height relative to the total height of the sewer pipe at the specific node ($height_j$), see Figure 13. This is because a certain level of sediment accumulation might require maintenance actions if it is a small sewer pipe while if it is a bigger pipe, this sediment level might be unproblematic. The target variable evaluated thus is:

$$y = \frac{h_{t,j}}{height_j} \times 100 \qquad (12)$$

A fill factor of 100 indicates that the sewer is completely full of sediments while a value of 0 indicates that there are no sediments at that specific node.

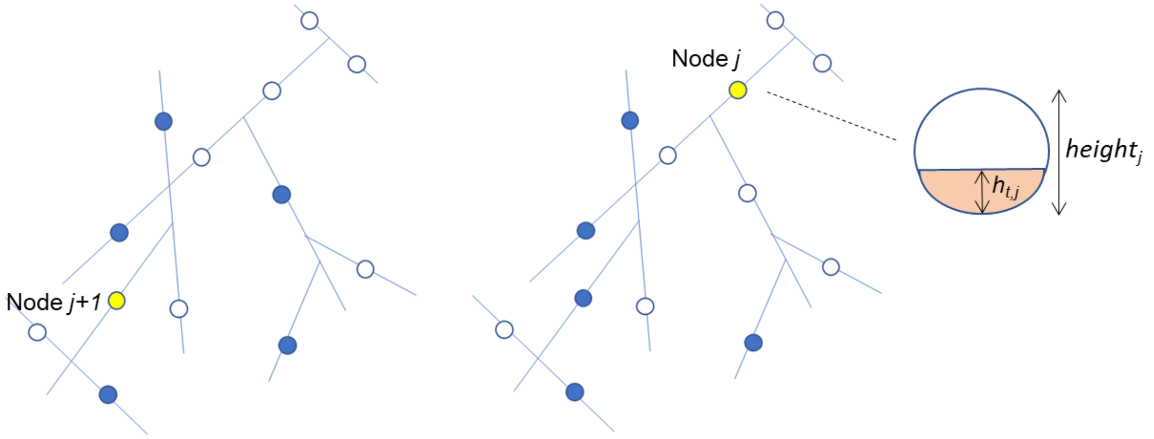

Figure 13: Sediment level prediction model. Example of predicted nodes (yellow), similar nodes (blue) used as input features and definition of the target variable.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

6.1.3. INPUT FEATURES

The required input features are three previous observations of the sediment levels in the node that is being predicted. Additionally, four previous observations from each of five “similar” nodes are required as inputs. Thus, to predict the sediment fill factor in a specific node, 23 previous sediment accumulation registers need to be inserted.

A particular aspect of the model is the concept of “similar” nodes. These have been established during pre-processing of the input data by identifying sets of sewer system nodes that show similar characteristics considering the sediment build up process. The approach involves that each node in the studied system is associated with a unique set of (5) similar nodes. This is illustrated in Figure 13 where the yellow dots indicate two examples of nodes that are being predicted. Each have their own “similar” nodes (blue dots) that might, or might not, coincide.

In addition to sediment level measurements, the model uses as inputs four previous recordings indicating weather cleaning has been applied in the node of interest.

Also, for the predicted node, the sewer pipe *height* and dry weather average wastewater flow and velocity are required as inputs.

In total, the model has 30 input features.

6.1.4. MODEL EVALUATION

The model is of the type Multi-Layer Perceptron (MLP) with 3 hidden layers, each consisting of 30 neurons. It was trained with 1668 samples and evaluated using the remaining 417 recordings (the test data).

The scatter plot of the predictions vs the observations as well as the histogram with all residuals can be seen in Figure 14. The testdata is quite concentrated to values between 0-20% sediment occupation. Only about 1.7% of the data are observations where the fill factor exceeds 20%. For the model development, training and evaluation, this is problematic since there is a risk that the MLP regressor learns that it can minimize the error by always predicting values in the range where a majority of the observations are (0-20%). The predictions of the four observations with a higher sediment fill factor of 30-40% are too low, which indicates that this might be the case here. On the other hand, one single observation with a fill factor in the range 40-80% is accurately predicted and contradicts this hypothesis. Since data is scarce, it is difficult to evaluate the certainty with which the model can predict fill factors higher than 20% . To improve the model it is suggested that data that better covers the possible range is collected. Since high sediment levels are a problem and therefore avoided (by cleaning), this type of data might not be easily obtained.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

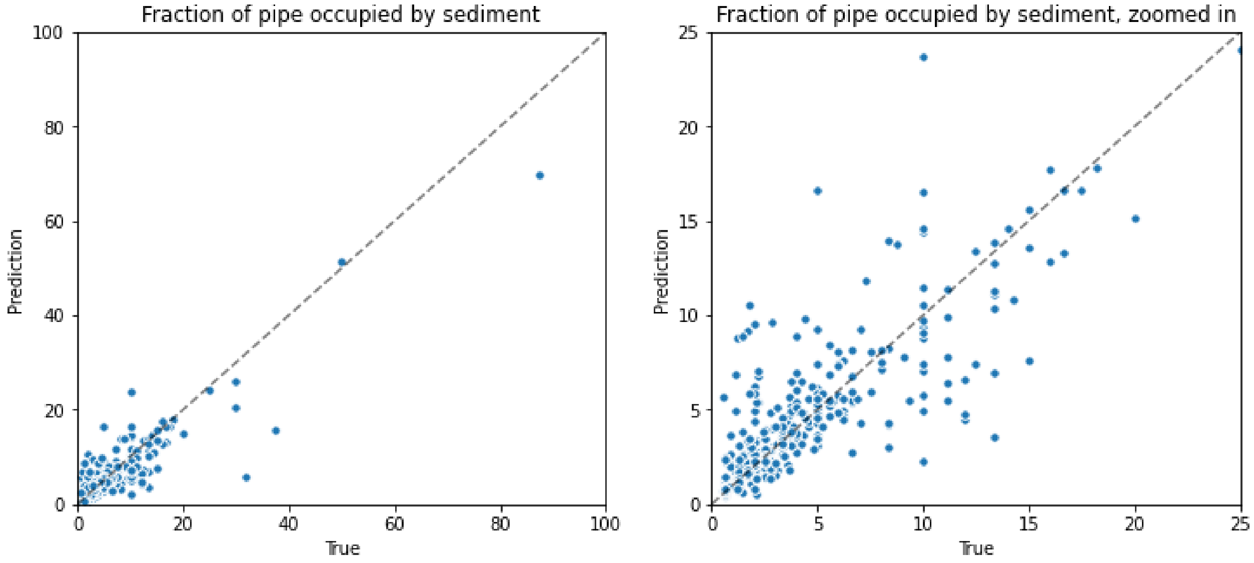

Figure 14: Scatter plot of the predictions and true values from the MLP regressor in the Barcelona case, predicting how much (%) of the pipes at each node that are occupied by sediment.

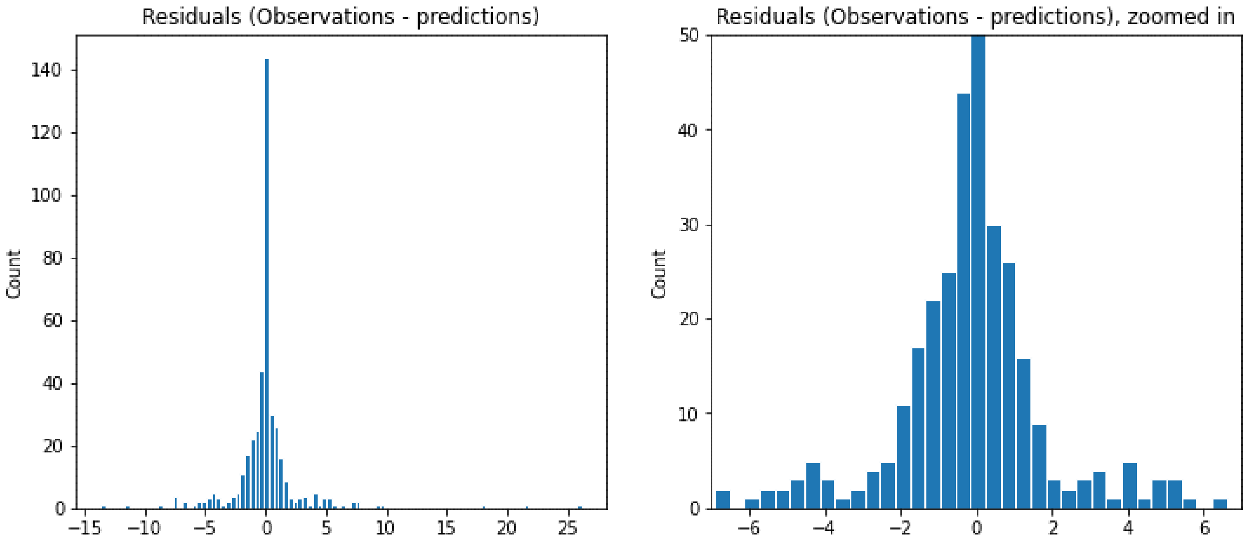

Figure 15: Residuals of the predictions and observations also presented in Figure 14. The right figure is the same as the left one but zoomed in.

The corresponding metrics used for evaluating the model's capability of predicting the fill factor follows in Table 1. The BIAS of practically 0 means that the studied model predictions were not biased in any direction. The mean average error (MAE) of 1.42% can be judged to be low since the measurement uncertainty is likely higher since it is a matter of manual measurements. However, if the model predictions are to be used for predictive maintenance and planning, it must be taken into account that the model performs rather bad for part of the input feature space. The higher value of the RMSE (2.97%) compared to the MAE reflects this.

As presented in 4.2.4, the R^2-score evaluates the predictive power of the model by comparing it with the performance of a model using only the mean of the observations. In general terms, the R^2 value of 0.8 is good (recall that 1 indicates a perfect model).

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

Table 1: Metrics corresponding to the scatter plot in Figure 14. The R^2 value of 0.8 is quite high although the test set data is not uniform which makes it little less impressive. The bias is practically 0 of the model.

Metric	
BIAS (%)	-0.02
MAE (%)	1.42
RMSE (%)	2.97
R^2	0.80
PI	0.54

The persistence index (Section 4.2.5) in Table 1 is 0.54. Similarly to R^2, the persistence index is a way of measuring the predictive power of the model. But instead of comparing to the mean of the observations, the persistence index evaluates if the model's predictions is more accurate than the previous measurement. For this specific case, the persistence index is calculated using the time index *t=1* instead of the prediction to see whether the prediction is better than just using the last measurement.

6.1.5. SENSITIVITY OF INPUT FEATURES

Concerning the features used when doing the predictions, there are differences in how much each feature contribute to each prediction. In Figure 16, these are shown using the python package SHAP which has been introduced earlier (Section 5.2).

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

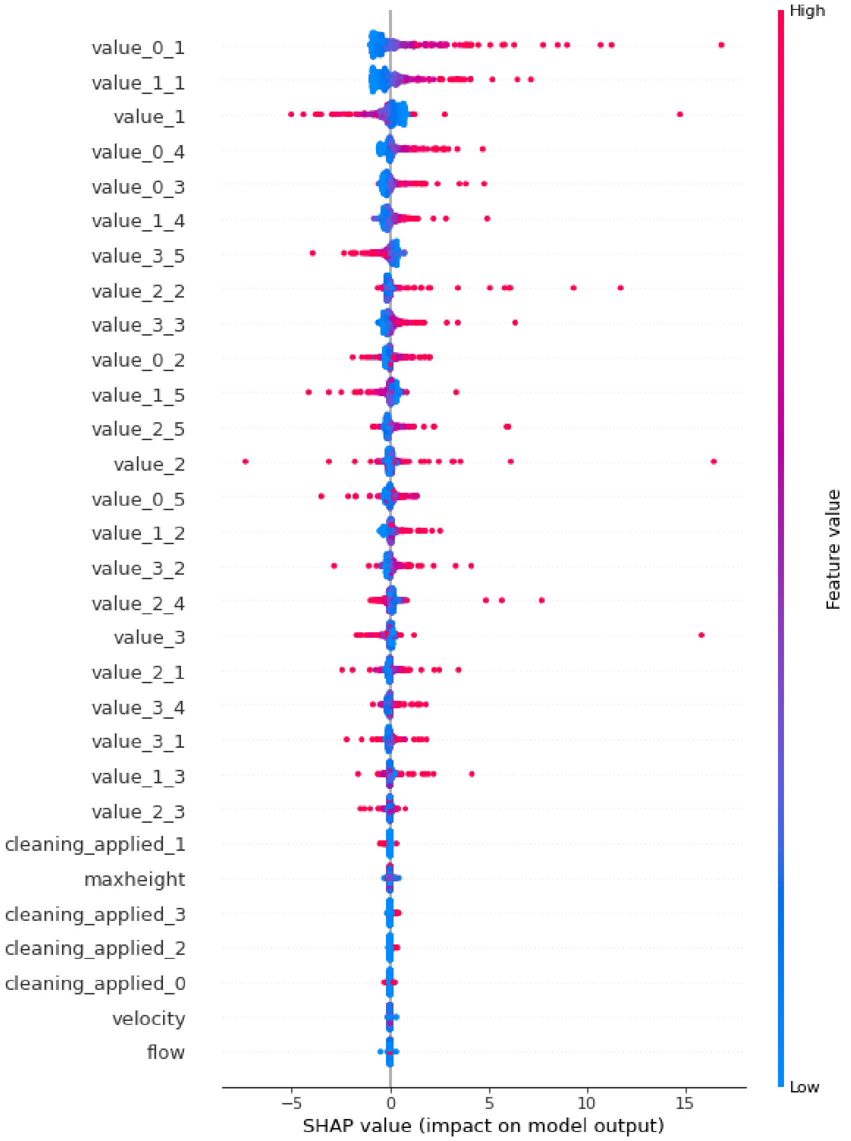

Figure 16: The SHAP values from the python package with the same name. The position of each dot indicates the impact of each feature for one prediction. The color of the dot shows the value of each feature.

The input feature names ”value” refer to the measured sediment levels (*h* in Section 6.1). Features formatted as ”value_*t*” represents the previous measurements in the node that is being predicted. Input features named value_*t*_*j* is the measured sediment level at time *t* in node *y*. The time *t*=0 means the current time, *t*=1 the previous measurement and so on. Each dot represent a prediction and the position of the dot is the impact on the model output. The color of the dot reflects the value of each feature.

Starting from the bottom of the figure, it is clear that the dimension (maxheight), the hydraulic condition (velocity, flow) and information about previous maintenance activities (cleaning_applied_*t*) in the node that is being predicted do not have a high impact on the modelled value of the target variable (the filling factor, Equation 12). Perhaps a bit surprising is that the features indicating cleaning is not important to the model. However, it could be argued that this is redundant information since it will show in the sediment level features if cleaning has been applied or not.

The top 10 input features with highest impact include the most recent sediment measurements (value_1 and value_0_*j*, *j*=1,2,3,4), which is intuitive. Some correlations are however unexpected. For example, according to the SHAP values, high values of ”value_1” (the previous measurement in the predicted node) in many cases seem to have a negative impact on the predicted filling factor.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

As an alternative to Figure 16, the average (absolute) SHAP value of each feature can also be calculated as shown in Figure 17.

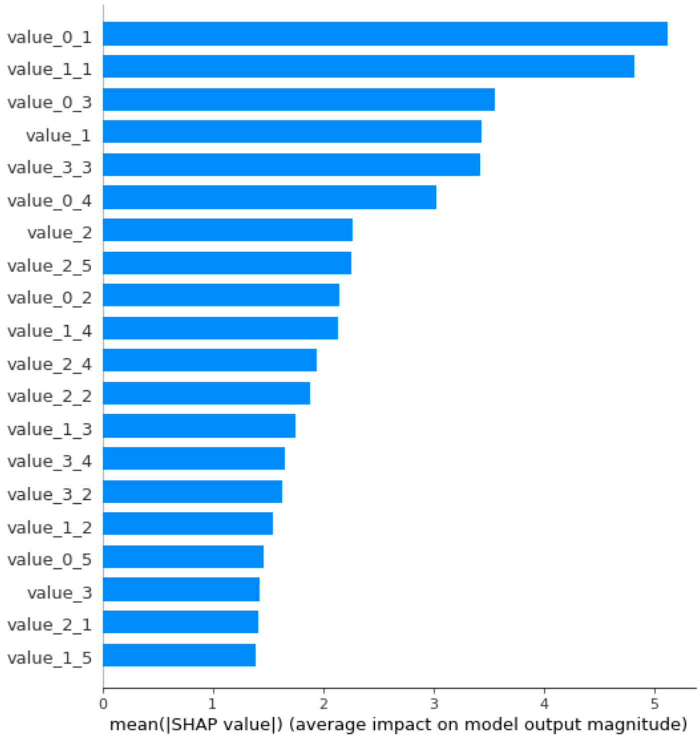

Figure 17: The top 20 features with most impact in terms of mean absolute values of SHAP values. Same data as in Figure 16.

6.1.6. DISCUSSION

The evaluation of the sediment level prediction model shows good results. An R^2 of 0.8 is very good, there seems to be no bias in the model and several expected features are ranked high in the analysis using SHAP. This means that for the test data set, the model performs well. The persistence index (0.54) also indicates that the model contains some useful information for the prediction. However, it must be noted that only 1.7 % of the observations of this involve fill factors above 20%. Thus, it is difficult to judge how good the model is at predicting high sediment accumulation levels. If this is an important part of the model objective more measured data in the higher ranges need to be collected.

Some limitations are also present in the model design. Since the previous measurement is included in the input data, and is also a prominent feature, the model can only be used once for each sewer pipe. The model also requires the current sediment measurement of 4 similar nodes to predict another one. This means that to predict the sediment level in a pipe, identifying and having access to data from 4 similar nodes is required.

6.2. FLOODING MODEL IN AMERSFOORT

6.2.1. MODEL OBJECTIVES

The objective of the flooding model is to assess the risk of flooding in a combined sewer system in the city of Amersfoort. The model will be part of a warning system that, given a precipitation prognosis, can warn if a flooding in any of 230 sewer system sections is about to occur.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

In opposite of the other SCOREwater ML models, the flooding model is trained with results from a mechanistic hydrological model. Thus, the results of the evaluation presented here is conditional on the performance of the hydrological model. The upside of using a ML model instead of the hydrological model is the computational time, which in this case is significantly reduced. Also, a ML-model can be implemented in open-source software such as Python, while the type of hydrological model used in this case study usually requires a software license.

6.2.2. TARGET VARIABLES

The model was developed and trained to predict, given precipitation data, an average volume for each section the previous five-minute period. A negative volume indicates that there is spare volume (volume that could be filled with water before flooding) in the section.

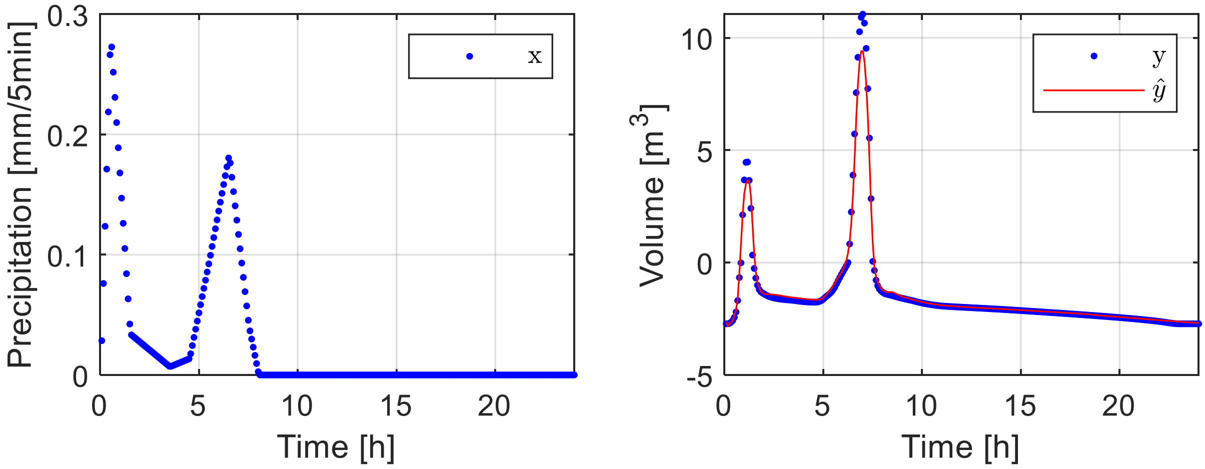

Figure 18: Example of input and output data of the flooding model for one rain event and one sewer section. *Left:* Precipitation data. *Right:* Observed (hydrological model) and predicted (ML model) volume. See text for further information.

A positive value indicates the volume of flooded water that is present above the ground level of the section area.

Three target variables (illustrated in Figure 19) were defined to evaluate the performance of the ML flooding model.

- y_F (-): Logical output indicating if flooding occurs during an event
- y_{TTF} (minutes): Time To Flooding, the duration from the start of the rain event to start of flooding. This measure is relevant since it contains information regarding how well the ML model can predict the dynamic response and the time it takes for a rain event to yield a flooding for each sewer section.
- y_{VF} (m^3): The maximum accumulated Volume of Flooded water in the streets, measuring the capability of the model to predict the overall extent of the event.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

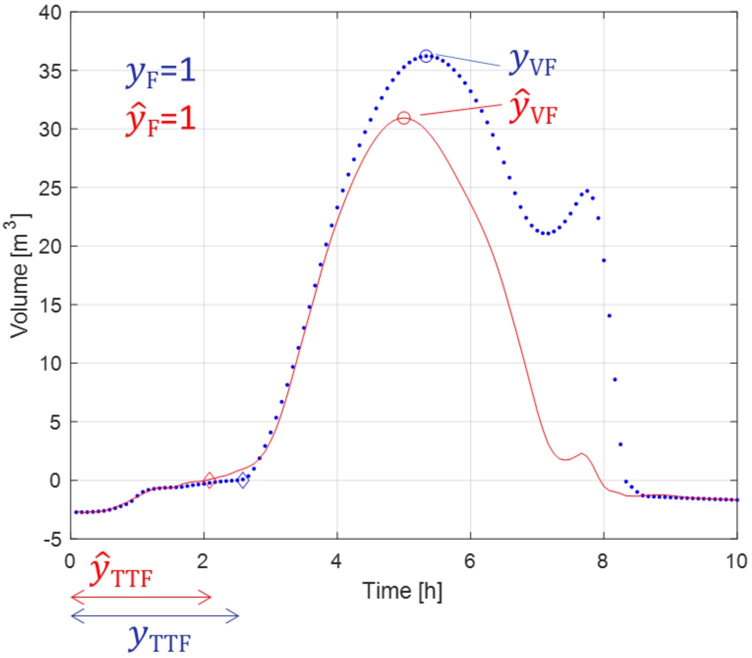

Figure 19: Illustration of the target variables used to evaluate the Amersfoort model. Blue symbols: observations (hydrologic model). Red symbols: (ML) model predictions.

6.2.3. INPUT DATA

As previously shown (Figure 18, left panel), the input data to the flooding model consists of several rainfall events with precipitation data having a resolution of 5 min. The events were generated with the hydrological model and varied in intensity and duration.

6.2.4. MODEL EVALUATION

The ML model is a neural network tasked to perform regression. It was trained with 101 rain events. 25 events were kept as test data for model evaluation.

For the model evaluation, it was assumed that, at the start of each rain event, the warning system has access to a precipitation prognosis for that event. Uncertainties in this prognosis is not considered and the rain events in the test data are assumed to be representative for events that will be tested in future.

Is flooding about to occur?

In general, the ML model is very good at mimicking the hydrological model considering if a flooding will occur or not. From the confusion matrix (Section 4.1.1), it is seen that the serious fault “false negative” (e.g. predicting no flood when a flood actually occurs) only happens 10 times out of totally 1137 observed floods. Moreover, the false negatives involve events with very small flooding volumes (<0.06 m^3). Looking at the false positives, the ML model incorrectly predicts 151 floods out of totally 4613 observations without flooding. Neither of these faulty predictions contain big flood volumes (<0.8 m^3).

When can we expect the flooding to start?

In Figure 20, histograms of the metrics BIAS, MAE and RMSE for each of the 230 sections and the model target y_{TTF} (time to flooding) is shown. The mean average error when the ML model is compared to the hydrological model varies between 0-60 minutes for most of the sections. A majority have an MAE of only 0-20 min. The BIAS histogram reveals that the errors are rather distributed around zero (no bias). Only for one section the bias is below -20 min. For this, the ML model is too slow which could be included as information in the warning system or for future model enhancements. In some sections the bias is positive, which, however, could be argued to be less serious since the ML model in these cases will indicate that the flooding will occur ahead of the time predicted by the hydrological model.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

For the target variable y_{TTF}, the RMSE values for each section is not very different from the corresponding MAE. This indicates that the spread is not significant. E.g. for each section, the ML model produces equally good or bad predictions of y_{TTF}, given the 25 events with test data.

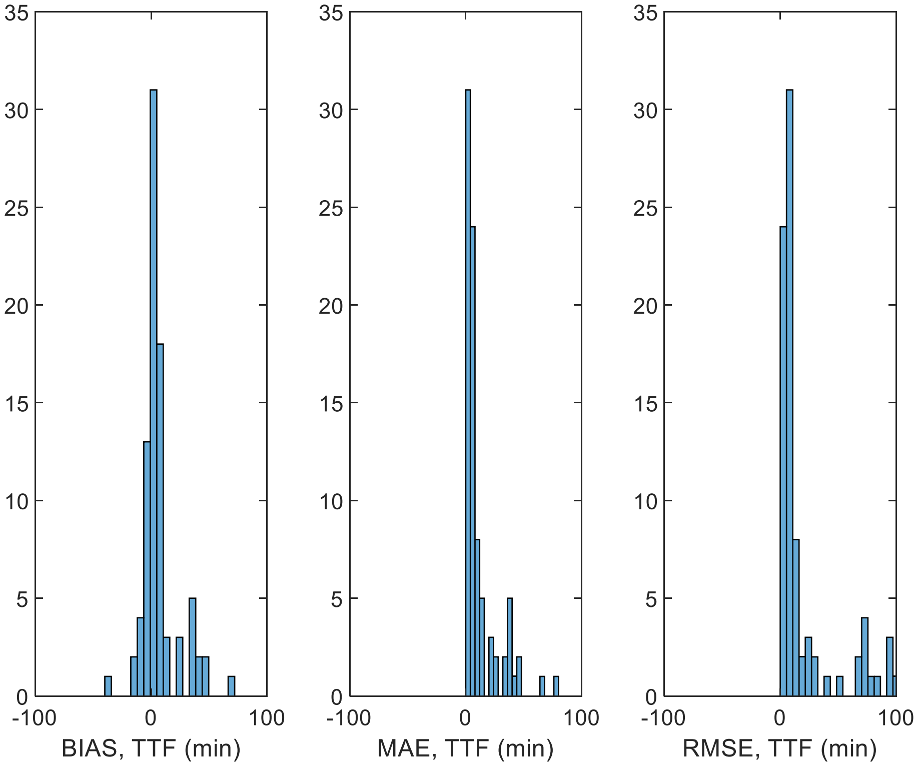

Figure 20: Histograms of the metrics BIAS, MAE and RMSE for the model target yTTF (time to flooding) for each of the 85/230 sections with at least one true positive flooding prediction.

What is the magnitude of the expected flood?

In Figure 21, histograms of the metrics BIAS, MAE and RMSE for each of the 230 sections and the model target y_{VF} (maximum accumulated Volume of Flooded water in the streets) is shown. Almost all sections have a positive bias indicating that the ML model in general predicts lower flooding volumes compared to the hydrological model.

The observed flooded volumes range between 0-100 m^3 and the errors are thus small (MAE=0-2 m^3).

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

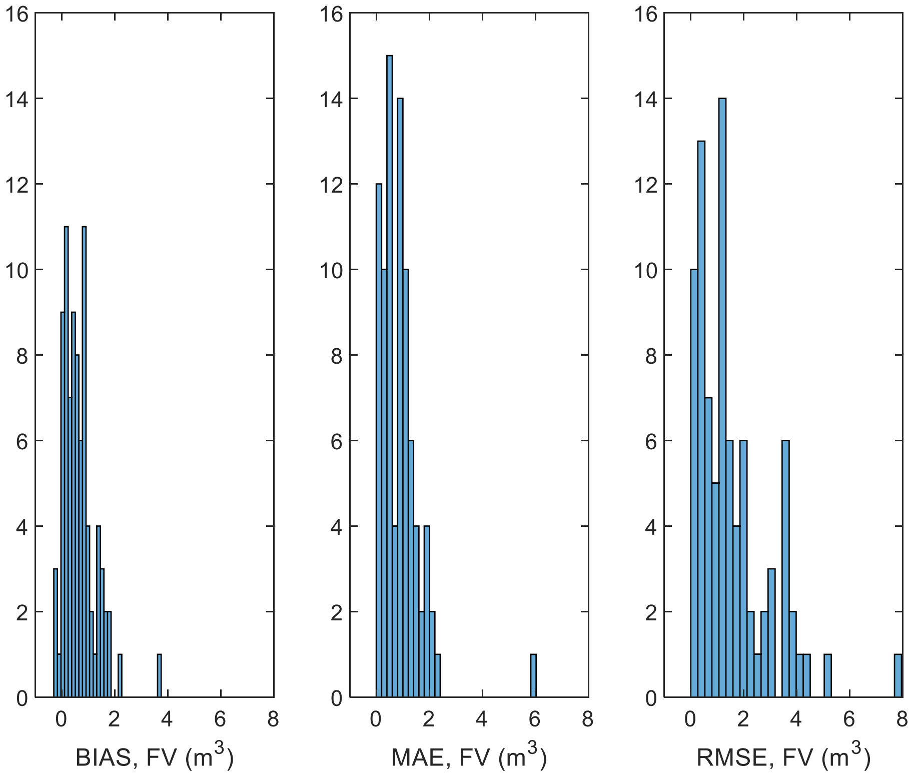

Figure 21: Histograms of the metrics BIAS, MAE and RMSE for the model target y_{VF} (maximum accumulated flooded volume) for each of the 85/230 sections with at least one true positive flooding prediction.

6.2.5. DISCUSSION

The ML model for prediction of flooding risk is very good if compared to the output of the hydrological model, which in this case represent the observations. The model generates very few false negatives, i.e. if a flooding is about to occur the ML model will predict this with high certainty. Also, the dynamic response, measured as the time it takes from the start of the rain event to flooding, is well predicted. The magnitudes of the flooding events are predicted with low error.

I must be highlighted that in this case, the goodness of the ML model is completely conditional on how well the hydrological model is performing. Compared to reality, nothing can be said from the evaluation in this deliverable.

6.3. GÖTEBORG PH REGRESSION MODEL

A water treatment plant is placed at a building site in Göteborg. The building site includes a lot of digging which results in a huge open shaft. In the shaft, a lot of water is collected, both rainwater and groundwater. The water is pumped to the water treatment station which cleans it before letting it out into the stormwater system. To verify that the water is properly cleaned, many parameters of the water is measured at the water treatment station. One of the parameters is the pH on both the influent and effluent water. The effluent water pH should be kept at a steady neutral level.

The model is a linear regression model, which means that it results in a regression equation where each input is assigned a constant.

6.3.1. MODEL OBJECTIVES

The pH in the water treatment plant is measured both on the influent and effluent water. If the effluent water has a very high or low pH, it could be necessary with some extra measures to adjust it. So, for the operator to have time to act, this model predicts when the effluent pH will be high, enabling the operation of an early warning system.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

6.3.2. TARGET VARIABLE

The target variable for the model is the effluent pH in the water treatment station, 40 time-steps ahead (a time step is 5 minutes so roughly 3 hours). The choice of predicting 40 time-steps ahead is part of the model design since it was shown that the effluent pH 40 time-steps ahead correlated well with the influent pH.

6.3.3. INPUT DATA

To predict the outgoing pH, the model takes advantage of previous measurements of the outgoing pH along with the pH in an earlier stage of the cleaning process. Specifically, the model has 6 inputs, the last three time-steps of effluent and influent pH.

6.3.4. MODEL EVALUATION

The regression equation for the linear regression model is

$$O_{t+40} = O_{t-2} * 0.364 - O_{t-1} * 0.013 + O_t * 0.573 + I_{t-2} * 0.017 + I_{t-1} * 0.004 - I_t * 0.003 \tag{13}$$

where *O* is the effluent pH, and *I* is the influent. From the coefficients of the model, we can note that the incoming pH is practically not used, the prediction is mostly based on effluent pH.

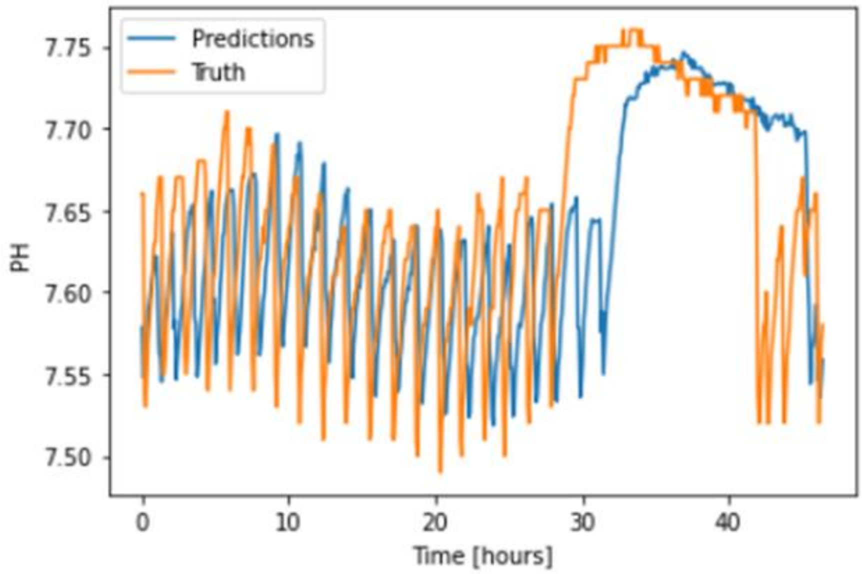

Figure 22: The predicted effluent pH 40 time steps ahead, along with the actual pH 40 times ahead. A time step is 5 minutes.

In Figure 22, the prediction seem to follow the truth quite well up until the big signal jump. The model is not able to follow the sudden change. As mentioned, this is also confirmed by the regression equation.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

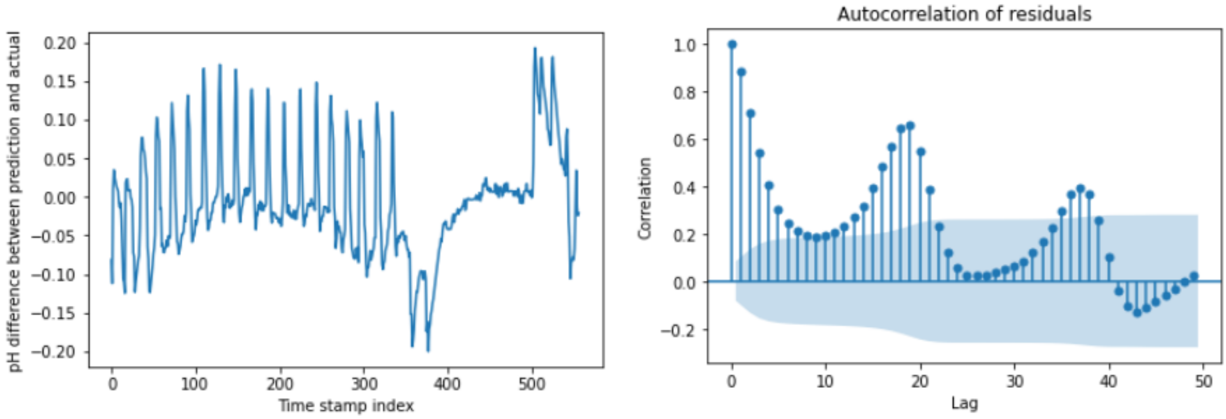

Figure 23: Left: Modelled behaviour left in the residuals. Right: The autocorrelation of residuals, again showing the remaining information in the residuals. The two peaks at lags about 19 and 38 indicate periodicity in the residuals.

If we look at Figure 23, we see that there is a periodicity in the residuals. In the autocorrelation of the residuals, there are peaks at lags about 19 and 38. This means that when the outgoing pH is predicted 40 time steps ahead, this happens to coincide with this periodicity which can give the idea that the model is performing well.

In summary, the model seem to perform well thanks to the periodicity in the outgoing pH. But when it comes to sudden changes, the model is not able to follow. Since the purpose of the model is to be able to give an early warning, this model need to be revised.

This can also be seen using the persistence index. That is, the model prediction (Equation 13) is compared with a simpler model, simply using the current outgoing pH instead of the prediction. This gives a PI of 0.08. A value of 0 would mean that it is equally good to use the current outgoing pH instead of the prediction, so the model is only slightly better than just using the current outgoing pH to predict the outgoing pH 40 time steps ahead.

6.3.5. DISCUSSION

The evaluation of the model shows that it needs to be revised. The persistence index is only 0.08 indicating that an equally good prediction is provided if the current measurement is used as estimation.

7. CONCLUSIONS

In this deliverable, a set of tools and methods for evaluating ML models have been reported. Since all models have unique goals and challenges, the focus of this deliverable is to present a streamlined model evaluation process.

The chosen metrics/key reference indicators for regression; MAE, RMSE, BIAS, R^2 and MARE are generally applicable while PI is a bit more dependent on the case and can be altered to fit specific needs. PI is a good tool for dummy testing models, to see if the prediction is better than something that is already present, and other things than the previous measurement can be used for the dummy check.

All classification measures presented are generally applicable, even though they will take different forms depending on the number of classes etc.

Shapley values are useful, especially when a lot of features are used as input to models. Using the python package SHAP also provides good figures for visualizing the results. Knowing what features are important and how they affect the prediction of the model is a very powerful tool in evaluating the performance of the model.

Monte Carlo dropout is harder to implement than Shapley values and require a neural network with dropout implemented which is very specific. On the other hand, black box models are often neural networks which also quite often have dropout implemented. It can both be useful for evaluating models and to provide uncertainty measures of models.

The conclusion for the whole streamlined model evaluation process is that the purpose of the model should be considered when interpreting the results of the metrics, knowing what the model should be used for is crucial when interpreting if the results are good or not. Multiple metrics should always be used, single metrics can imply extraordinary performance while a combination give the full picture. A combination of multiple metrics is always recommended.

The conclusions for the three models that was evaluated in the deliverable for the sake of the evaluation methodology should be interpreted as examples of what conclusions that can be drawn when doing model evaluation. The following can be said regarding the models:

- The evaluation of the sediment level prediction model (Barcelona) shows good results. The R^2 of 0.8 is high, there seems to be no bias in the model and several expected features are ranked high in the analysis using SHAP. Many input features, however, seems to have no importance at all, which suggests that the model can be simplified. Also, few observations involve high sediment levels and therefore, for these regions, it is difficult to evaluate the model.
- The ML model for prediction of flooding risk (Amersfoort) is very good if compared to the output of the hydrological model, which in this case represent the observations. The model generates very few false negatives, i.e., if a flooding is about to occur the ML model will predict this with high certainty. Also, the dynamic response, measured as the time it takes from the start of the rain event to flooding is well predicted, are the magnitudes of the flooding events. It must be highlighted that in this case, the goodness of the ML model is completely conditional on how well the hydrological model is performing. Compared to reality, nothing can be said from the evaluation in this deliverable.
- The evaluation of the pH prediction model shows that it needs to be revised. The persistence index is only 0.08 indicating that an equally good prediction is provided if the current effluent pH measurement is used as model.

For all applied evaluation tools, it is presupposed that the test datasets are representative considering the objectives of the models. If this is not the case, the applicability of the evaluation results (and models), becomes limited. Two methods for sensitivity and uncertainty analysis (SHAP and Monte Carlo dropout are proposed as tools for dealing with this limitation.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

The above-mentioned models are currently being refined and additional models are being developed in SCOREwater. The final evaluations will be reported in D2.5 (Final version of data-driven models report for a water-smart society) using the herein presented methods, metrics, and key reference indicators.

7.1. LIST OF PYTHON PACKAGES

For the evaluation environment, the following not standard Python libraries were used:

Sklearn – machine learning library

Shap – explaining the importance of inputs on the outputs of machine learning models

Tensorflow – neural networks including Monte Carlo dropout

Scipy – scientific computing

Statsmodels – statistical computations

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

8. REFERENCES

Baldi, P., and Sadowski, P. J. (2013). Understanding dropout. In: *Advances in neural information processing systems* (pp. 2814–2822).

Bennett, N.D., Croke, B.F.W., Guariso, G., Guillaume, J.H.A., Hamilton, S.H., Jakeman, A.J., Marsili-Libelli, S., Newham, L.T.H., Norton, J.P., Perrin, C., Pierce, S.A., Robson, B., Seppelt, R., Voinov, A.A., Fath, B.D., Andreassian, V., (2013). Characterising performance of environmental models. *Environ. Model. Softw*. 40, 1e20. http://dx.doi.org/10.1016/j.envsoft.2012.09.011.

Gal, Y. and Ghahramani, Z., (2015). Dropout as a bayesian approximation: representing model uncertainty in deep learning, *Proceedings of the 33rd International Conference on Machine Learning*.

Ribalta, M., Mateu, C., Bejar, R., Rubión, E., Echeverria, L., Alegre, F.J.V and Corominas, L., (2021). Sediment Level Prediction of a Combined Sewer System Using Spatial Features, *Sustainability*, 13(4013).

Lundberg, S.M. and Lee, S.-I., 2017. A unified approach to interpreting model predictions. *In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., pp. 4765–4774.

Loquercio, A., Segu, M. and Scaramuzza, D., (2020). A General Framework for Uncertainty Estimation in Deep Learning. *IEEE Robotics and Automation Letters*, 5(2), pp.3153-3160.

Sundararajan, M. and Najmi, A. (2020). The many Shapley values for model explanation. arXiv:1908.08474

Srivastava, N, Hinton, G, Krizhevsky, A, Sutskever, I, and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 2014.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

ANNEX 1 – STOCKTAKING

A final Annex of stocktaking was included in all Deliverables of SCOREwater produced after the first half-year of the project. It provides an easy follow-up of how the work leading up to the Deliverable has addressed and contributed to four important project aspects:

1. Strategic Objectives
2. Project KPI
3. Ethical aspects
4. Risk management

STRATEGIC OBJECTIVES

Table 2 lists those strategic objectives of SCOREwater that are relevant for this Deliverable and gives a brief explanation on the specific contribution of this Deliverable.

Table 2: Stocktaking on Deliverable's contribution to reaching the SCOREwater strategic objectives.

Strategic objectives	Contribution by this Deliverable
SO4: Demonstrate benefits of smart water management for increased water-system resilience against climate change and urbanisation	By evaluating the model performance, it is validated that the smart water management lives up to the goal of being smart.
SO5: Identify and mitigate key barriers to implementation of smart, resilient water management	The evaluation will point out weaknesses in the models and thereby identify key barriers.

PROJECT KPI

Table 2: lists the project KPI that are relevant for this Deliverable and gives a brief explanation on the specific contribution of this Deliverable.

Table 3: Stocktaking on Deliverable's contribution to SCOREwater project KPI's.

Project KPI	Contribution by this deliverable
KPI 4: Reduce the pollutant load from construction work in Göteborg	Ensures that the models used to optimize treatment operation works in a desired way.
KPI 5: Reduce the flooding risk through integrated water management in Amersfoort	Ensures that the models used to model the effects of mitigating actions against flooding works as desired.
KPI 6: Reduce the release of wet pipes and discharge of oil, grease and antibiotics to the sewer in Barcelona.	Ensures that the models used for prediction of sewer maintenance works as desired.

ETHICAL ASPECTS

Table 4 lists the project's Ethical aspects and gives a brief explanation on the specific treatment in the work leading up to this Deliverable. Ethical aspects are not relevant for all Deliverables. Table 4 indicates "N/A" for aspects that are irrelevant for this Deliverable.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

Table 4. Stocktaking on Deliverable’s treatment of Ethical aspects.

Ethical aspect	Treatment in the work on this Deliverable
Justification of ethics data used in project	N/A
Procedures and criteria for identifying research participants	N/A
Informed consent procedures	N/A
Informed consent procedure in case of legal guardians	N/A
Filing of ethics committee’s opinions/approval	N/A
Technical and organizational measures taken to safeguard data subjects’ rights and freedoms	N/A
Implemented security measures to prevent unauthorized access to ethics data	N/A
Describe anonymization techniques	N/A
Interaction with the SCOREwater Ethics Advisor	N/A

RISK MANAGEMENT

Table 5 lists the risks, from the project’s risk log, that have been identified as relevant for the work on this Deliverable and gives a brief explanation on the specific treatment in the work leading up to this Deliverable.

Table 5. Stocktaking on Deliverable’s treatment of Risks.

Associated risk	Treatment in the work on this Deliverable
Lack of consensus on scientific or technological approach (M/H)	The evaluation metrics and method have been discussed with Eurecat since they are the future users of the environment, thereby guaranteeing that there is consensus.
Data from Cases are sparse and are not enough to apply all methods and tools (L/M)	The methods and tools for evaluation are tailored to fit with the available data.
Project execution failure, technical problems and delays (key milestones or deliverables delayed) (M/H)	No technical problems have appeared.

This project hase recieved funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement no 820751

scoreWATER