Soft sensing of water depth in combined sewers using LSTM neural networks with missing observations

Palmitessa, Rocco; Mikkelsen, Peter Steen; Borup, Morten; Law, Adrian W.K.

Published in:
Journal of Hydro-Environment Research

Link to article, DOI:
10.1016/j.jher.2021.01.006

Publication date:
2021

Document Version
Peer reviewed version

Soft sensing of water depth in combined sewers using LSTM neural networks with missing observations

Rocco Palmitessa[1,2,*], Peter Steen Mikkelsen[1,a], Morten Borup[1,b], Adrian W.K. Law[2,c]

[1]*Technical University of Denmark, Department of Environmental Engineering, Bygningstorvet, Building 115, 2800 Kgs. Lyngby, Denmark*

[2]*Nanyang Technological University, School of Civil and Environmental Engineering, 50 Nanyang Avenue, Singapore 639798, Singapore*

**Corresponding author: rocp@env.dtu.dk*

a: psmi@env.dtu.dk; b: morb@env.dtu.dk; c: cwklaw@ntu.edu.sg

Abstract

Information and communication technologies combined with in-situ sensors are increasingly being used in the management of urban drainage systems. The large amount of data collected in these systems can be used to train a data-driven soft sensor, which can supplement the physical sensor. Artificial Neural Networks have long been used for time series forecasting given their ability to recognize patterns in the data. Long Short-Term Memory (LSTM) neural networks are equipped with memory gates to help them learn time dependencies in a data series and have been proven to outperform other type of networks in predicting water levels in urban drainage systems. When used for soft sensing, neural networks typically receive antecedent observations as input, as these are good predictors of the current value. However, the antecedent observation may be missing due to transmission errors or deemed anomalous due to errors that are not easily explained. This study quantifies and compares the predictive accuracy of LSTM networks in scenarios of limited or missing antecedent observations. We applied these scenarios to an 11-month observation series from a combined sewer overflow chamber in Copenhagen, Denmark. We observed that i) LSTM predictions generally displayed large variability across training runs, which may be reduced by improving the selection of hyperparameters (non-trainable parameters); ii) when the most recent observations were known, adding information on the past did not improve the prediction accuracy; iii) when gaps were introduced in the antecedent water depth observations, LSTM networks were capable of compensating for the missing information with the other available input features (time of the day and rainfall intensity); iv) LSTM networks trained without antecedent water depth observations yielded larger prediction errors, but still comparable with other scenarios and captured both dry and wet weather behaviors. Therefore, we concluded that LSTM neural network may be trained to act as soft sensors in urban drainage systems even when observations from the physical sensors are missing.

Keywords

Combined sewer overflow, urban drainage systems, machine learning, data-driven models, soft sensing, LSTM, Recurrent Neural Networks

1. Introduction

Urban drainage systems are a critical component of cities' infrastructure. In recent times, advancements in analytics, sensing, transmission, computing and data management have opened new possibilities for information technology to aid the management of these urban drainage systems (Eggimann et al., 2017). One potential application is to train a model to predict the expected behavior of the system. If sufficiently accurate and reliable, the model can act as a soft sensor capable of validating or replacing the hard sensor, e.g. if the sensor is faulty or under maintenance.

The water depth in combined sewers commonly follows a diurnal water pattern in dry weather and peaks in response to precipitation events. The response of the water depth to the observed rainfall is often non-linear, as it is affected by the state of the catchment and the drainage network. Control actuators (gates, pumps, weirs, etc…) can further contribute to the non-linearity. Deterministic hydrodynamic models have evolved to replicate in detail the physical processes governing these drainage systems. They are also termed white-box models, as they are solely formulated on the physical knowledge of the system and disregard any stochasticity (Breinholt et al., 2011). At the other end of the spectrum are black-box or purely data-driven models, which derive the relationships among system states exclusively from the available data without any domain knowledge. These data-driven models can potentially outperform deterministic models especially for online uses, given their ability to better capture the non-stationarity of urban drainage systems (Jonsdottir et al., 2007).

Among the data-driven models for urban drainage systems, Artificial Neural Networks (ANN) were early on identified as promising for their ability to learn the complex, non-linear behavior (Loke et al., 1997). Applications of ANN in urban drainage have explored a wide range of topics, including solid transport modelling (Gong et al., 1996), estimation of sanitary flows (Djebbar and Kadota, 1998) and real time control (Lobbrecht and Solomatine, 2002). Recently, focuses have shifted towards extreme events, and a number of studies have explored the use of ANN for flood forecasting (Savić et al., 2013; Bruen and Yang, 2006; Rjeily et al., 2017; Duncan et al., 2013) and overflow prevention (Sumer et al., 2007; Darsono and Labadie, 2007). For example, Mounce et al. (2014) trained an artificial neural network (ANN) capable of predicting the combined sewer overflow (CSO) depth 75 min ahead with less than 5% error. ANN predictions have also been proven useful in detecting anomalies such as blockages (Bailey et al., 2016), thus enabling proactive maintenance (Rosin et al., 2019). More recent studies have explored the benefits of forecasting flows in urban drainage using hybrid models, in which the ANN is coupled with a hydrodynamic model (She and You, 2019) or a wavelet transformation (Ayazpour et al., 2019).

Recurrent neural networks (RNN) have also gained popularity in urban hydrology since they display even higher potential for time series forecasting, given their capability to preserve a "memory" of the past in layman definition. Long short-term memory (LSTM) networks, a variant of RNN, have been proven to outperform traditional ANN in predicting both the flows (Sufi Karimi et al., 2019) and water levels in

combined sewers (Zhang et al., 2018). With the memory capability, LSTM is now widely used as either a forecasting or soft sensing tool by adjusting the prediction horizon and feeding as input the relevant rainfall data (forecasts or observations, respectively). In both cases, the latest available water quantity observations are commonly used as input features, as the prediction accuracy of ANNs decreases significantly in general when the antecedent observations are unknown to the network (Fernando et al., 2006). However, in the real operation of urban drainage systems, past observations might be missing for short or long periods of time when they are deemed anomalous or the signal from the sensor has not been received. These missing data gaps can pose a challenge to the system control that relies critically on the observed information, for example the activation of pumps when the water level reaches a threshold. It is unclear at this stage whether LSTM can be used effectively in the presence of the missing data gaps.

In this study, we investigate the use of LSTM neural networks as a soft sensing tool for urban drainage systems in the scenarios of missing or limited antecedent observations. Particularly, we compare the prediction accuracy of different LSTM networks trained: i) without knowing the antecedent water depth, ii) with different gap durations in the antecedent water depth, and iii) with different periods of training data. We present and discuss results obtained from 11 months of real observations from a combined sewer overflow chamber in Copenhagen, Denmark. By optimizing the use of LSTM networks in scenarios with limited input data and assessing the accuracy, we develop a custom LSTM approach that is sufficiently robust to supplement missing observations and, therefore, further bridge the gap towards the field implementation of machine learning in the operation of urban drainage systems.

2. Methodology

2.1. Soft sensing with Artificial Neural Networks

Soft sensors are mathematical models of a system designed to estimate relevant system variables (Graziani et al., 2007). Hardware sensors can have high installation and maintenance costs, especially in harsh environments like urban drainage networks. Soft sensors offer a low-cost alternative, thus enabling more comprehensive monitoring networks. Soft sensors can also work in parallel with hardware sensors and be used for validating the sensor observations. A model used for soft sensing carries the knowledge of the physical or statistical relationship among the system variables. Alternatively, this relationship can be inferred by an artificial intelligence model that is trained on past observations.

Artificial Neural Networks (ANNs) are artificial intelligence methods developed based on how biological neural systems are believed to work (Schmidhuber, 2015). During the training step, the internal configuration of the ANN is shaped according to the recurrent patterns between inputs and outputs. A trained network can then be used to estimate or predict the output given only the input information. In an urban drainage system, a neural network can be trained on historical observations of a given hydraulic quantity (such as water levels or flows) and then used to predict the current value of this quantity based on recent observations. If an

observation becomes available, its validity can be evaluated by comparing with the ANN estimate (for example, for anomaly detection). Otherwise, the prediction itself can be used as a replacement of the observed value (hence termed as soft sensing). The process is repeated each time an observation is expected. After a predefined period, the network can be updated (re-trained) with the most recent data in order to intermittently adapt to the changing conditions of the system (Figure 1).

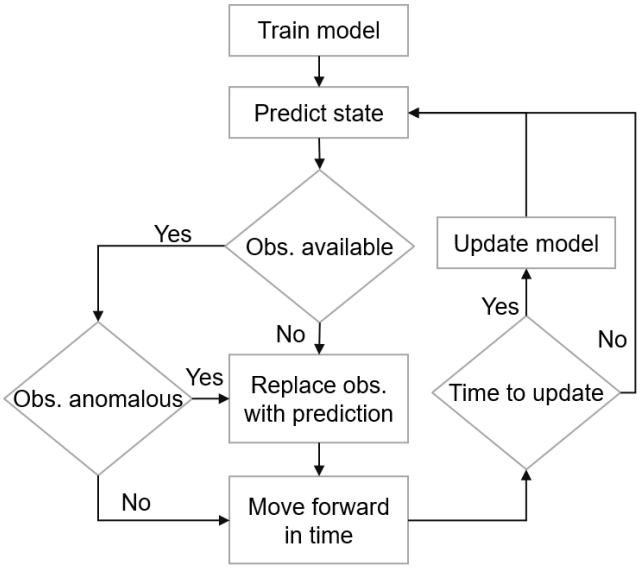

Figure 1. Conceptual overview of soft sensing and anomaly detection, including model updating (re-training).

An ANN is a collection of nodes, artificial neurons, and edges, equivalent to artificial synapses. Neurons and edges have weights, which constitute the trainable parameters of the network. Neurons are typically aggregated into layers, and the first and last layers interface with the user-defined input and output data. The intermediate layers are hidden from the user and hold the learning capability of the network. The signal travels from the input layer to the output layer and is distributed among the neurons proportionally to their weights. The input signal to each node is passed through an activation function to generate the output signal. When the signal finally arrives at the output layer, it is scored against a target value, for example the physical observation of the system corresponding to the predicted state. The calculated score or loss is then used iteratively to optimize the internal weights of the network. This process is the basic learning mechanism of the ANN.

2.2. Long Short-Term Memory neural networks

Different from traditional ANNs, Recurrent Neural Networks (RNNs) take the state of the hidden neuron at the previous time steps as an additional input for the next time step (Elman, 1990). This property makes the RNN particularly suitable to learn sequences of data, i.e. time series. However, RNN have a limited capability of learning long-term dependencies, i.e. when the prediction of the desired output depends on inputs at a much earlier time (Bengio et al., 1994). In Long Short-Term Memory (LSTM) networks, a variant of RNN, this problem is solved by supplementing the neurons with a memory cell. The memory cell can store the long-term information, and is read and written via appropriate gates that open and close according to the current

state of the cell and neuron (Hochreiter and Schmidhuber, 1997). In theory, LSTM networks are capable of capturing the short-term delay between the rainfall and the water depth in an urban drainage network and infer the long-term properties of the catchment.

The learning process of LSTM networks is controlled and optimized with hyperparameters, which are not derived by training, but need to be set before training. The number of layers and neurons fall within this category, together with the number of cycles through the full training dataset, called epochs, and the number of training examples being processed at the same time, called the batch size. The activation function can also be regarded as a hyperparameter, as well as the optimization algorithm and its associated loss function and learning rate. To prevent overfitting on the training data, and thus improve the model performance on unseen data, the LSTM can be equipped with a regularization mechanism. The most frequently used is dropout, which randomly selects some nodes and removes them along with their incoming and outgoing connections (Hinton et al., 2012). The probability that a node is disregarded in the training is called the dropout rate, and can also be regarded as a hyperparameter of the LSTM.

2.3. Prediction accuracy

Neural networks are trained to minimize the error, or loss, between the predicted and targeted values (which are typically observed values). The learning dataset is split in two subsets: a part of the data is used for calibrating the internal parameters (training); the remaining is used for quantifying the error or loss associated to the current sets of parameters (validation). This process is repeated at each learning cycle or epoch. Therefore, the validation loss quantifies the prediction accuracy of the ANN and drives the learning process until it converges to a narrow range or after a predefined number of epochs. Once the training is complete and the optimal set of internal parameters has been identified, the network can be applied to an independent testing dataset.

The Mean Squared Error (MSE) is the default choice of loss function for training ANN on time series. It is calculated as the average of the squared differences between the predicted and observed values. The same metric could be used to evaluate the prediction accuracy on the testing dataset. For better interpretability of the results, we use instead the Root Mean Squared Error (RMSE) in this study, which has the same unit as the observed value. RMSE relates the prediction accuracy to the range of values observed in the system, while relative comparisons can also be made directly among different configurations and scenarios.

3. Experimental setup

3.1. Case study

The key data in this study are water depth observations from a Combined Sewer Overflow (CSO) chamber in the city of Copenhagen, Denmark. The CSO is part of a large combined drainage network that serves the westernmost area of the city and discharges in the local treatment plant (Figure 2, left). A detailed description of the study area is given by Palmitessa et al., 2020. The CSO chamber is designed to discharge excess inflows

to a storage tunnel via a fixed weir set 0.99 m above the chamber invert (Figure 2, right). An ultrasonic sensor records the level of the water surface in the chamber at 1-min resolution, and sends the signal to the central control system of the utility company. The observations are used to detect the occurrence of overflow events and manage the operation of the downstream storage tunnel. The data used in this study was collected in December 2019, and the observations cover the first 11 months of 2019. For simplicity, the water level was converted to water depth by subtracting the bottom level.

The water depth series has hundreds of data gaps likely caused by transmission errors, the vast majority of which has a duration shorter than 15 min. Only seven data gaps in the observation period last longer than 2 hours. These could be due to the maintenance or recalibration of the sensor, but no detailed explanation was obtained from the data provider. The gaps do not hinder the training and testing of the ANN, but rather demonstrate the need for a soft sensor that works in conjunction with the hardware sensor.

The study area is equipped with a dense network of rainfall intensity gauges. These are tipping bucket devices, i.e. small buckets that funnel the precipitation and tip when at max capacity. The volume of the bucket and the time interval between two consecutive tips are used to calculate the rainfall intensity in µm/s, also at 1-min resolution. A total of ten rainfall gauges are located within or nearby the study area (Figure 2, left). Observations from all 10 gauges were obtained for the period covered by the water depth sensor data.

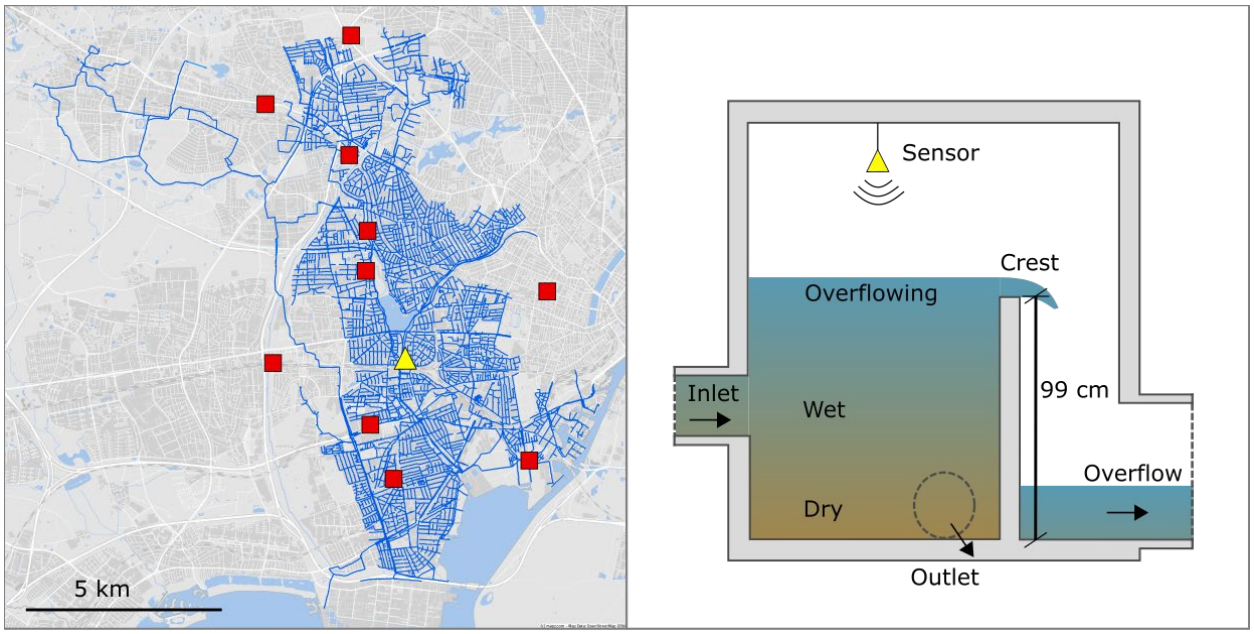

Figure 2. Left: drainage network (blue lines) and location of rainfall gauges (red squares) and overflow chamber (yellow triangle). Right: schematics of overflow chamber with water depth stages (dry, wet and overflowing).

Water depth observations fall within three stages as shown in Figure 2, right and Figure 3: dry weather, when the sole contribution to the flow in the system is due to the sewage, which follows the diurnal water consumption pattern; wet weather, when the catchment runoff mixes with the sewage flow but the water depth in the CSO chamber is below the crest level of the weir; and overflow, when the water depth exceeds the crest level. At each stage, the water depth has a different response to the precipitation. While the

threshold due to the crest level between the wet weather and overflow stages is known, the separation line between the dry and wet weather stages needs to be determined.

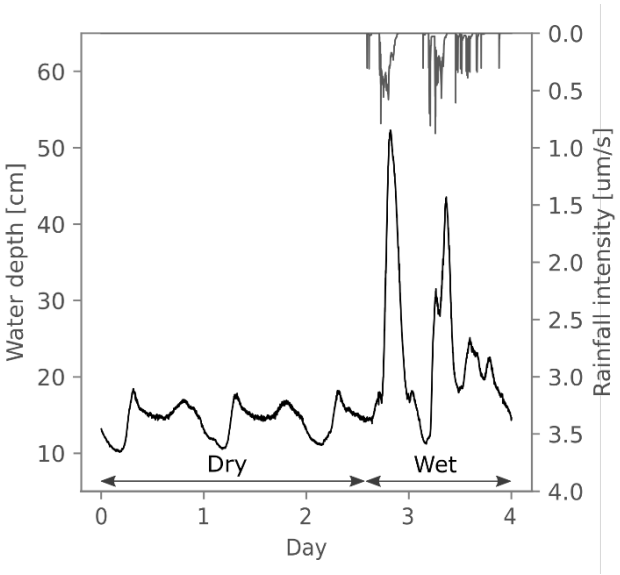

Figure 3. Example of rainfall intensity (grey) and water depth (black) observations in dry and wet weather.

If the water depth observations are plotted against the corresponding rainfall intensity, no clear pattern emerges (Figure 4, left). In fact, due to the delayed response to precipitation, there is no direct correspondence between the water depth stage at a given time and the rainfall measured at the same instance. In theory, the maximum water depth in dry weather could be estimated from prolonged dry periods for matured cities. However, processes such as infiltration and exfiltration, together with the weekly and seasonal changes in water consumption, affect the dry weather values, making it difficult to automatically classify the observed water depth. Therefore, it is beneficial to train a soft sensor capable of working across all stages, eliminating the need to classify the data points beforehand. Moreover, the establishment of the soft sensor can also track the trending changes in the city development. At the same time, it should be noted that the dataset is highly unbalanced, as seen from the cumulative distribution function (Figure 4, right). Within the observation period, 90% of data values fall within 6.5 and 22.5 cm, while only 0.06% of the water depths observations exceed the overflow threshold (corresponding to only 3 hours distributed among 6 events occurring in the course of the 11 months).

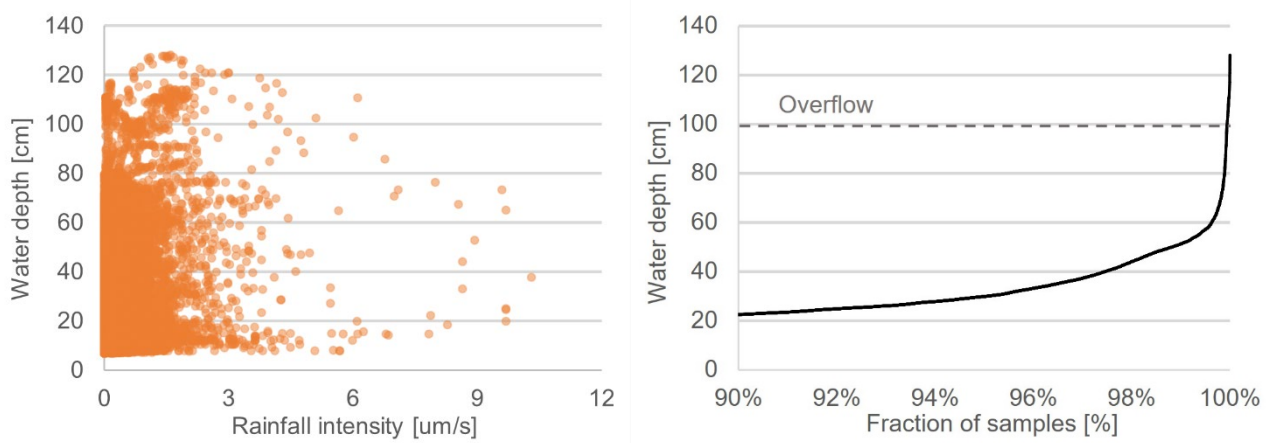

Figure 4. Left: Water depth observations against corresponding rainfall intensity observations (average of available gauges). Right: Cumulative distribution function of water depth observations (only top 10% of sample).

3.2. Input features selection

In time series, antecedent observations carry information about the current value, which is especially true for water depth observations since mass conservation is one of the dominating processes. This is confirmed by the autocorrelation function of the water depth series (Figure 5), which shows the average correlation coefficient of two data points in the series as a function of the time lag between them. The autocorrelation is highest at the time lag of 0 min and decreases to about 0.5 at 180 min. In addition to the antecedent water depth, the associated minute of the day was used as input feature to enable the ANN to estimate the daily wastewater fluctuation in dry weather. For wet weather analysis, the cross-correlation function between the rainfall intensity and water depth was computed for all 10 available gauges and for the average series, with time lags ranging between 0 and 180 min (Figure 5). The function was higher at all time lags for the average rainfall intensity series compared to the individual ones and peaked at about 60 min time lag with a value of 0.55. Therefore, the average rainfall intensity series was chosen as the third input feature. For comparison, the autocorrelation of the water depth was higher than the average rain cross-correlation at all lags.

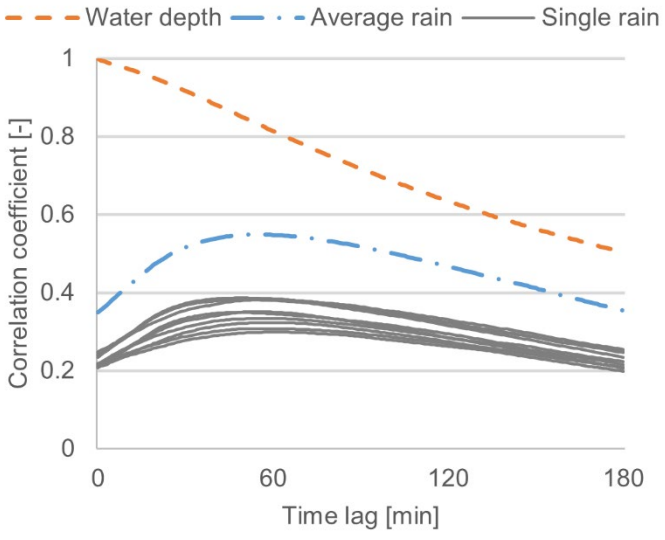

Figure 5. Autocorrelation of water depth (dashed) and cross-correlation of water depth and rainfall intensity at individual gauges (solid) or with average series (dot-dashed). Hyperparameter selection

3.3. Hyperparameter selection

An optimal set of LSTM hyperparameters can significantly improve the accuracy of the prediction but is specific for the problem at hand. Therefore, the choice often relies on a combination of calibration and user judgment. The selection of the main network hyperparameters was based on a simple grid search. The grid included 1, 2 or 4 hidden layers, each having 4, 16 or 64 neurons. Regularization layers with dropout rate of 25, 50 or 75% were attached to each input and hidden layer. Every combination of layers, neurons and dropout rate was tested with a batch size of 1024, 2048 or 4096. To account for the variability in the training outcome, each set of hyperparameters was tested 10 times, for a total of 810 runs. The median RMSE for the validation dataset was used to compare different combinations of hyperparameters and select the best performing one.

All other hyperparameters were chosen among default values or based on the authors' experience. For example, the network was trained with a max of 50 epochs and stopped earlier if the validation loss did not improve significantly for ten epochs. The Rectified Linear (ReL) max(0,x) was used as the activation function and the Adaptive Moment Estimation (Adam) with default learning rate (Kingma and Ba, 2015) was selected as the optimizer to minimize the mean squared error. The model development was carried out using the deep learning library Keras (Chollet and others, 2015).

3.4. Model implementation

Each input feature was individually normalized to the interval [0.1, 0.9] and rearranged in a rolling window of length w. Therefore, the LSTM networks were trained to predict the water depth at any time t (target) given as input the water depth, average rainfall intensity and minute of the day between t-1 and t-w (Figure 6). To quantify the relative contribution of each feature to the prediction accuracy, the following constellations of input features were tested for training the LSTM: A) All three input features; B) Rainfall intensity and time of the day, and C) Water depth and time of the day.

The test was conducted with input windows of length 5, 30, 60, 90, 120, 150 and 180 min. The first 9 months of the dataset were used for learning, of which 80% for training and 20% for validation, and the last 2 months for testing. In the rainfall intensity series, values larger than 0 accounted for 9.1% of the training and validation dataset, and 12.4% of the testing dataset.

In a second test, we fixed w to 120 min and tested learning periods p between 1 and 9 months, with 1 month increments. For direct comparability, the prediction accuracy was always tested on the last 2 months of the dataset and the p observations prior to that were used for learning. The test was conducted for input feature constellation A and B.

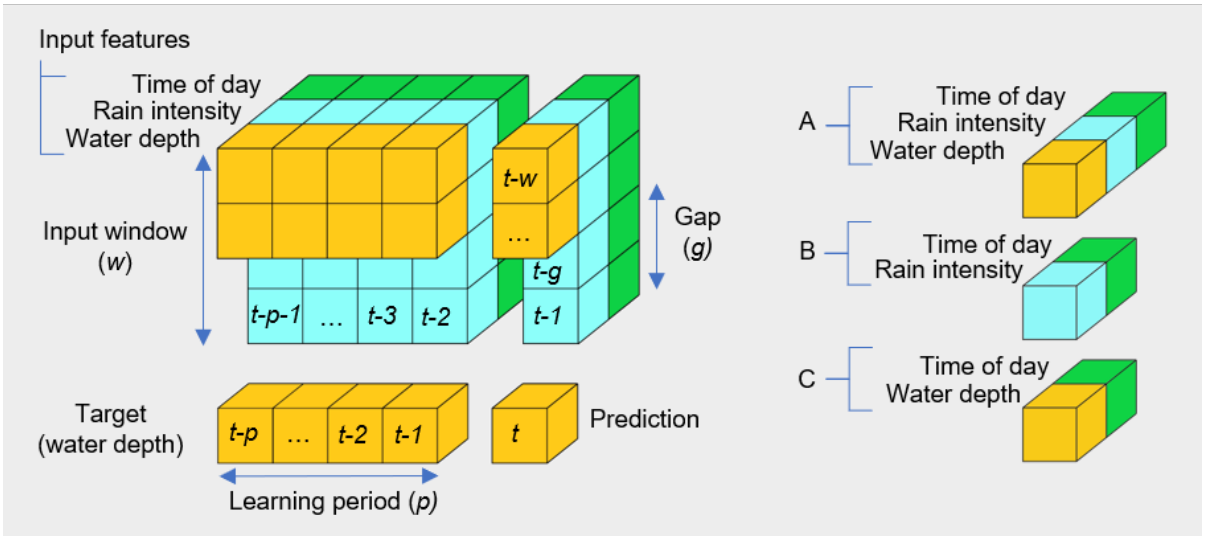

Figure 6. Left: Visualization of LSTM input features and target, with input window length (w), gap length (g) and learning period (p). Right: Definition of input feature constellations (A, B, C).

Finally, we also tested intermediate scenarios in which only some of the antecedent water depth observations are known (A_g). We did so by artificially introducing gaps of length *g* in the input window. In this scenario, the LSTM input still included the rainfall intensity and minute of the day between *t-1* and *t-w*, but only the water depths observations between *t-g-1* and *t-w*. Gap lengths of 5, 30, 60, 90 and 120 min were tested, with the last case (A_{120}) being equivalent to input feature constellation B. For this test, *w* was fixed at 120 min and *p* was set to 9 months. For the ease of comparison across different settings and scenarios, the prediction accuracy was quantified as the RMSE of the LSTM prediction for the testing period, which was the same for all tests. To account for the variability in the LSTM prediction, each combination of *w*, *p* and *g* was tested 20 times and the results are presented as box plots of the combination-specific RMSE scores.

3.5. Model update (re-learning)

As new observations from the system become available, the model can be updated by learning from new information. If the model is not updated, predictions can still be made based on prior trained parameters (equivalent to the You-Only-Look-Once (YOLO) approach for image processing). On the other hand, an updated model can better respond to possible transient changes in the system behavior. In this case, the frequency of the update, or in other terms the interval between updates, should be optimized in relation to the temporal scale of the processes altering the response of the system to the rainfall (e.g. seasonal variabilities in soil saturation and infiltration-inflow).

We also investigated an iterative approach to model update, in which the network was periodically updated with a learning period of fixed length. With this approach, the network is asked to re-learn the input-output relationship after each update interval. Therefore, the model was reset before each update and the most recent observations were used for learning. The test was repeated for learning periods of 3 and 5 months and update intervals of 4, 8, 12 and 24 weeks (Figure 7). Shorter update intervals were not tested with this

approach due to the longer computational time required. To compare the prediction accuracy associated with different update intervals, the testing predictions from all updates were combined in a single 6 months long period and the overall RMSE was computed. Similarly to the previous tests, the results are presented as box plots of 20 runs.

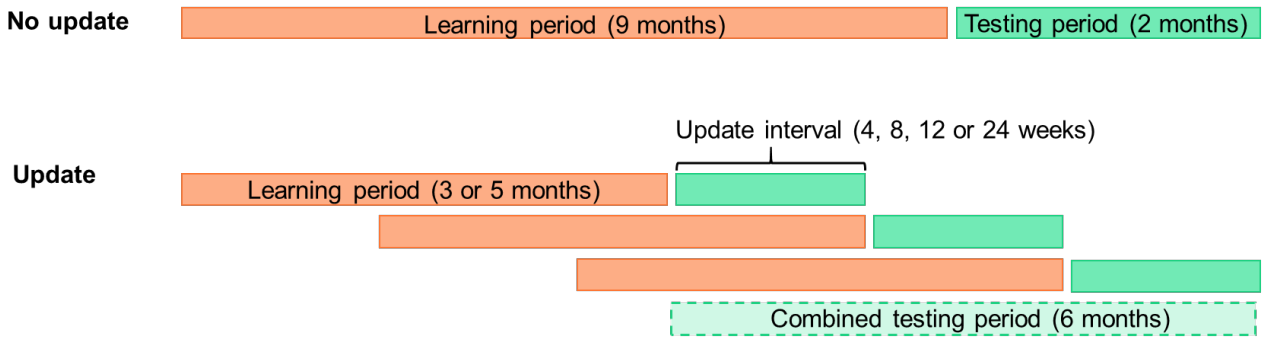

Figure 7. Subdivision of dataset in learning and testing with no model update (top) and with iterative update (bottom).

4. Results

4.1. Hyperparameter calibration

The grid search as described in Section 3 was performed independently for two configurations, with the window length of 120 min, learning period of 9 months, and input B and A_{60}, respectively. Among all 81 combinations of hyperparameters tested, the same combination returned the lowest median test RMSE with both configurations: 2 hidden layers, 64 neurons per layer, batch size of 2048 and 25% dropout rate. This combination of hyperparameters yielded a median validation RMSE of 4.2 and 3.4 cm respectively for the first and second configuration. Generally, combinations with 1 hidden layer, 4 neurons per layer and 75% dropout rate performed worst. Mixed results were, instead, obtained with each of the batch sizes tested. A deeper investigation of the interaction between hyperparameters and a wider/denser grid search would allow to further improve the prediction accuracy, but it is beyond the scope of this study.

4.2. Input window length

Separate LSTM networks were trained with different input window lengths and different constellations of input features. When input A was used, the median RMSE in the testing dataset increased with the input length (Figure 8), from 0.8 cm with *w* of 5 min to 3.2 cm with *w* of 180 min. Therefore, when the most recent water depth was known, adding past information did not improve the prediction accuracy. Conversely, when the antecedent water depth was an unknown to the network (B), the RMSE decreased with the input window length, and the median test RMSE reached a minimum of 5.1 cm for *w* of 120 min.

Moreover, with input C, the prediction accuracy was largely the same as with A. This suggests that when the water depth was predicted using all three features as input (A), the LSTM relied heavily on the antecedent water depth observation and the rainfall data was mostly disregarded. This result is consistent with the

correlation analysis shown in Figure 5, where the correlation coefficient was always higher for the antecedent water depth, and at its highest at a lag of 1 min. Therefore, the LSTM learned to assign a high weight to the most recent water depth and a lower weight to all other inputs. For comparison, the persistence with lag = 1 min returned a RMSE of 0.2 cm in the testing dataset. In other words, the simple prediction based on the water depth observation from a minute before was on average a better predictor of the current state than any of the tested LSTM configurations. In the following, we shall term the approach with such simple prediction as persistence modelling.

It should be noted that the results displayed a large variability among the 20 runs for each configuration. This could be due to the effects of the regularization mechanism, the limited number of epochs used for learning, the randomness of the initial state of the network and the choice of default learning rate in the optimization algorithm. Three instances across all tests conducted in this study even resulted in negative predictions of water depth, and were discarded from the reported results. The issue needs further investigation.

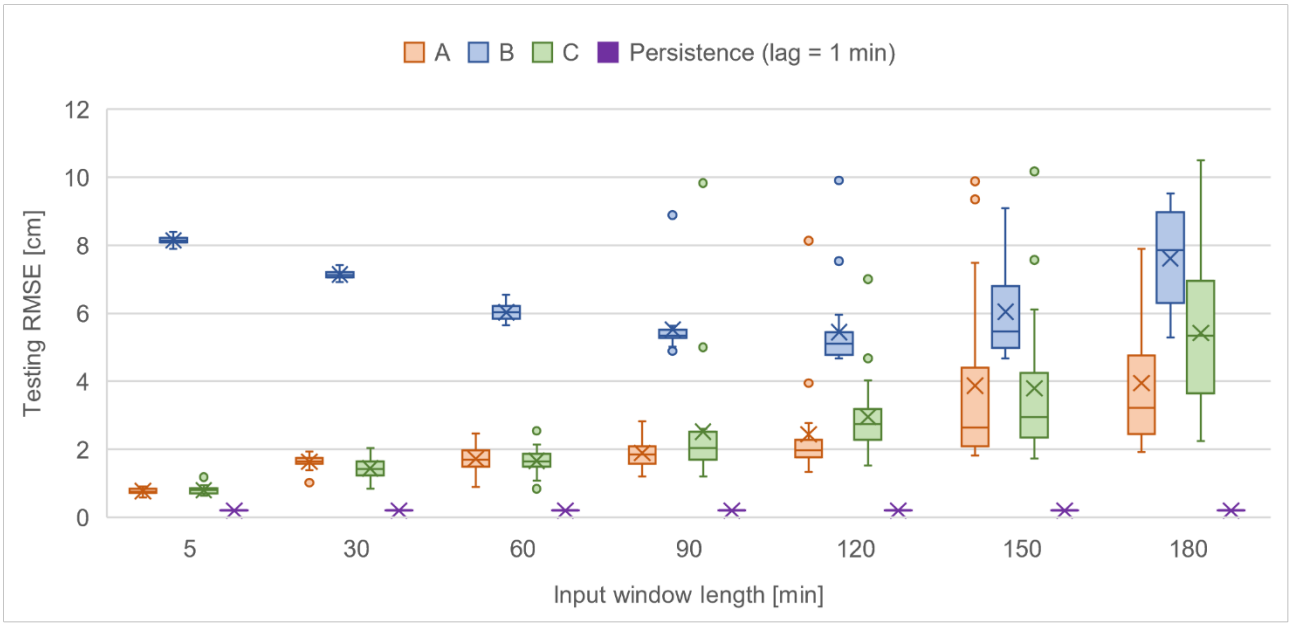

Figure 8. Root Mean Square Error (RMSE) of 20 LSTM predictions in testing dataset with p = 9 months and different input window lengths. Networks trained on all features (A), only rainfall intensity and time of the day (B), or water depth and time of the day (C). For reference, testing RMSE of persistence with lag 1 min is also plotted. Results presented are quartiles (box) and mean (cross). The whiskers extend to the minimum and maximum value or 1.5 times the inter quantile range. All values outside the whiskers range are marked as outliers (circles).

4.3. Learning period

To assess how the learning period affects the prediction accuracy, we trained separate LSTM networks with *p* ranging from 1 to 9 months. The tests were conducted with both inputs A and B and with a fixed window length of 120 min. The prediction accuracy generally decreased with increasing *p* (Figure 9), as the networks were trained on more data. For both configurations, the gain in accuracy was significant when extending the learning period from 1 to 5 months, and marginal for periods longer than 5 months. For example, the median

RMSE in the test dataset decreased from 4.9 cm with a learning period of 1 month to 2 cm with *p* equal to 5 months, which was about the same value calculated with the longest learning period. When the antecedent water depth was unknown (B), the prediction error was consistently higher, but followed the same trend as the LSTM trained on all features (A). This result implied that the memory of the state of the system was kept generally within the last 5-month data.

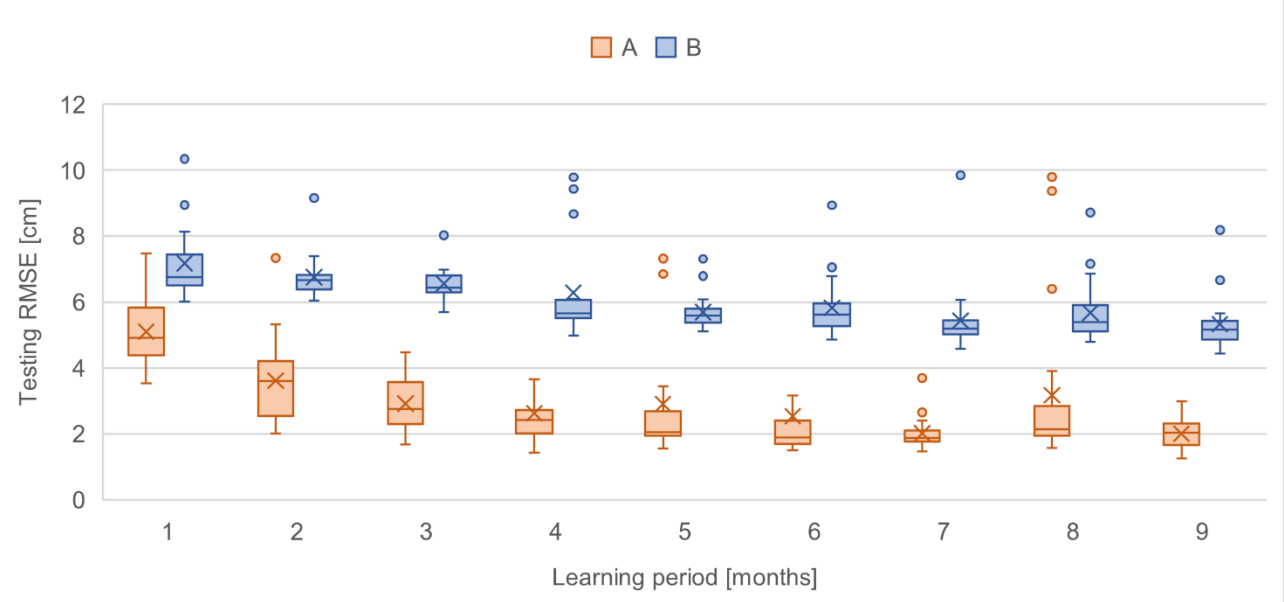

Figure 9. Root Mean Square Error (RMSE) of 20 LSTM predictions in testing dataset with w = 120 min and different learning periods. Networks trained on all features (A), or only rainfall intensity and time of the day (B). Results presented are quartiles (box) and mean (cross). The whiskers extend to the minimum and maximum value or 1.5 times the inter quantile range. All values outside the whiskers range are marked as outliers (circles).

4.4. Gap length

We also tested LSTM networks with w = 120 min and gaps of length g in the antecedent water depth observations (A_g), and we observed that the prediction accuracy decreased as g increased (Figure 10). These represented intermediate scenarios to those investigated above: the case of g=0 min corresponded to including all antecedent water depth observations in the input window (A); the case of g=120 min corresponded to removing the water depth from the input features (B). The median RMSE in the test dataset for gaps of 5, 30, 60 and 90 min fell between the two.

The accuracy of the LSTM predictions was compared to the RMSE of a persistence model with lag equal to g. The neural networks outperformed the persistence model, but only for gaps longer than 30 min. For example, the median RMSE in the test dataset was 3.75 cm for a gap of 60 min when the water depth was predicted with a LSTM trained on all input features, and 5.2 cm when the observation from 1 hour before was used as prediction. This finding suggests that the LSTM was capable of translating the information in the rainfall and the time of the day into water depths to compensate for the limited knowledge on the antecedent water levels. The assumption was verified by repeating the test without the rainfall intensity input (input C). In this case, the prediction accuracy was similar or worse than the persistence model for all gap lengths.

The scenarios of partial knowledge of the past observations were simulated by applying the same gap across both the training and testing datasets, under the assumption that LSTM performs best in a scenario if trained under the same setting. To validate this assumption, we trained LSTM networks on a dataset without gaps, and tested them on a dataset with artificially introduced gaps. Indeed, we observed a significant increase in the prediction error compared to networks trained on a dataset with gaps in the antecedent water depth.

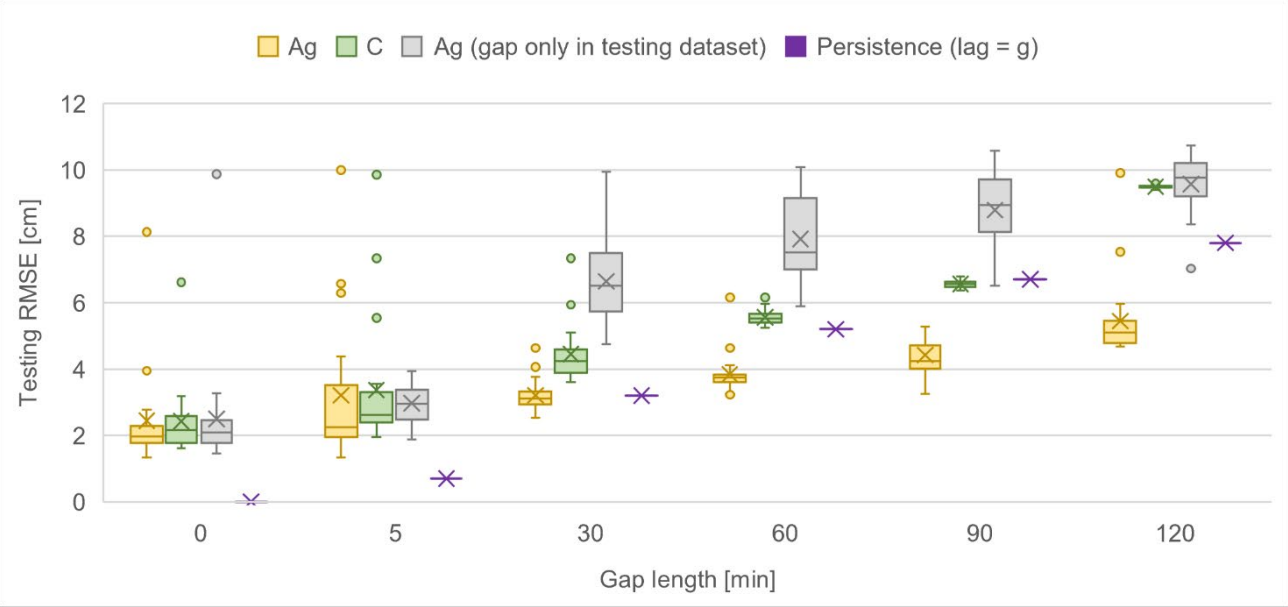

Figure 10. Root Mean Square Error (RMSE) of 20 LSTM predictions in testing dataset with w = 120 min, p = 9 months and gaps of different length g in the antecedent water depth. Networks trained on all features (A_g) or water depth and time of the day only (C). For reference, persistence with lag = g is also plotted. Results presented are quartiles (box) and mean (cross). The whiskers extend to the minimum and maximum value or 1.5 times the inter quantile range. All values outside the whiskers range are marked as outliers (circles).

4.5. Update interval

The iterative update approach was tested in the scenario of input B and *w* = 120. For direct comparability, the prediction accuracy was calculated as RMSE on the same combined testing period (6 months long). For all update intervals tested, the updated LSTMs performed similarly to the non-updated networks (Figure 11). Marginal improvements were observed for update intervals of 8 and 12 weeks compared to 4 and 24 weeks. However, the networks updated with a learning period of 5 months did not perform consistently better than the updates with 3 months of learning data, as observed in Figure 9.

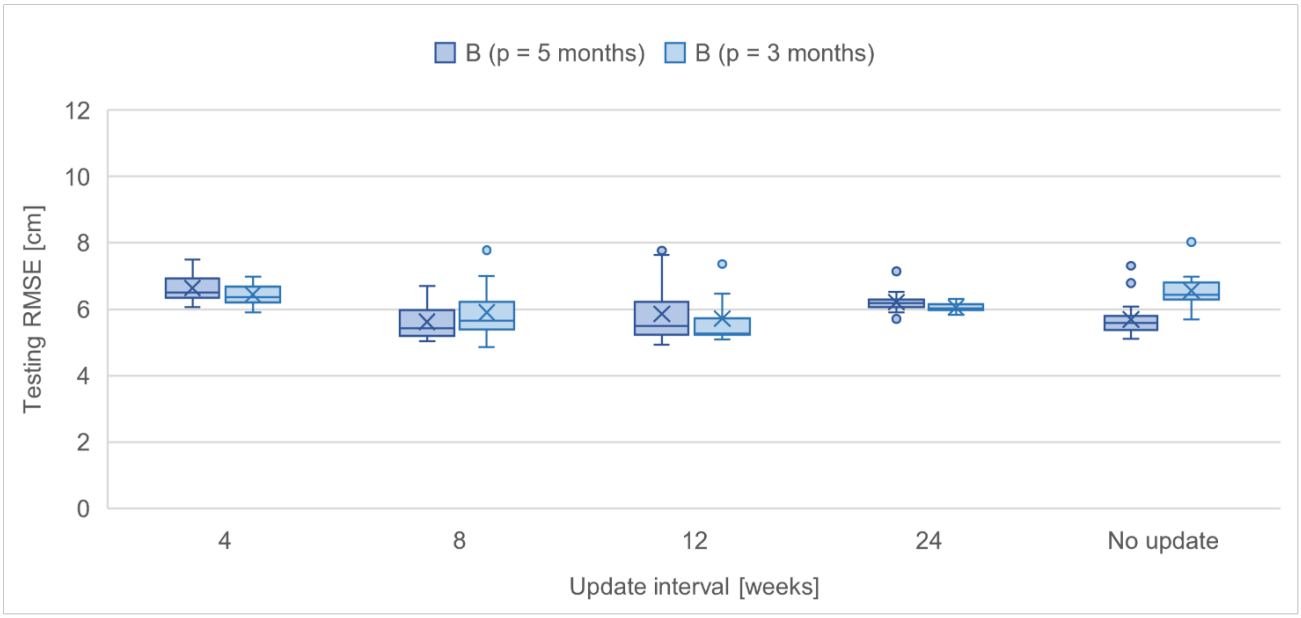

Figure 11. Root Mean Square Error (RMSE) of 20 LSTM predictions in combined testing dataset with w = 120 min, input B and learning periods of 3 or 5 months with different update intervals. Results presented are quartiles (box) and mean (cross). The whiskers extend to the minimum and maximum value or 1.5 times the inter quantile range. All values outside the whiskers range are marked as outliers (circles).

5. Discussion

The prediction accuracy was presented as RMSE for the entire testing dataset, which allowed the direct comparison of different configurations and scenarios with a single score. To analyze the prediction accuracy in relation to the water depth domains (dry, wet and overflowing), we selected three input configurations (A, A_{60} and B with w = 120 min, p = 9 months) and calculated the testing RMSE for ranges of observed water depth (Figure 12). It should be noted that the lower range captured all dry weather observations and the upper range all overflow events in the testing dataset. The analysis showed a clear tendency of the RMSE increasing with the water depth for all three configurations. With input B, the median RMSE increased from 2.14 cm in the 0-20 cm range to 50 cm in the 100-120 cm. For comparison, the median RMSE for all values of water depth was 5.11 cm. These values are higher than the errors calculated for input A and A_{60}, but within the same order of magnitude.

The obtained results show a clear state dependency of the error. This is further exacerbated by scoring the predictions with the RMSE, which squares the errors and therefore penalizes high values. However, as seen in Figure 4, the dry weather observations far outnumber the wet weather values, especially those within the overflowing domain. Consequently, the overall RMSE (0-120 cm) is very close to the score calculated on the dry weather range (0-20 cm). Also, the neural network will tend to learn with higher accuracy the dry weather behavior than the rarer, extreme wet events. In case of even slight delays between the observed and predicted water depths, the RMSE returns high errors even if the magnitude of peak values corresponds well. Therefore, other scoring rules should be investigated in the future to better assess the prediction accuracy

for wet weather. Wet weather predictions will always, however, be more uncertain than dry weather predictions due to the large uncertainties in rainfall data and complex surface runoff processes. Figure 12 shows, however, that the LSTM network performs much better than the persistence model in ordinary wet weather conditions, while the relative difference is rather small during CSO events. The latter can be explained by the fact that the water levels are rather stable once they supersede the CSO crest level.

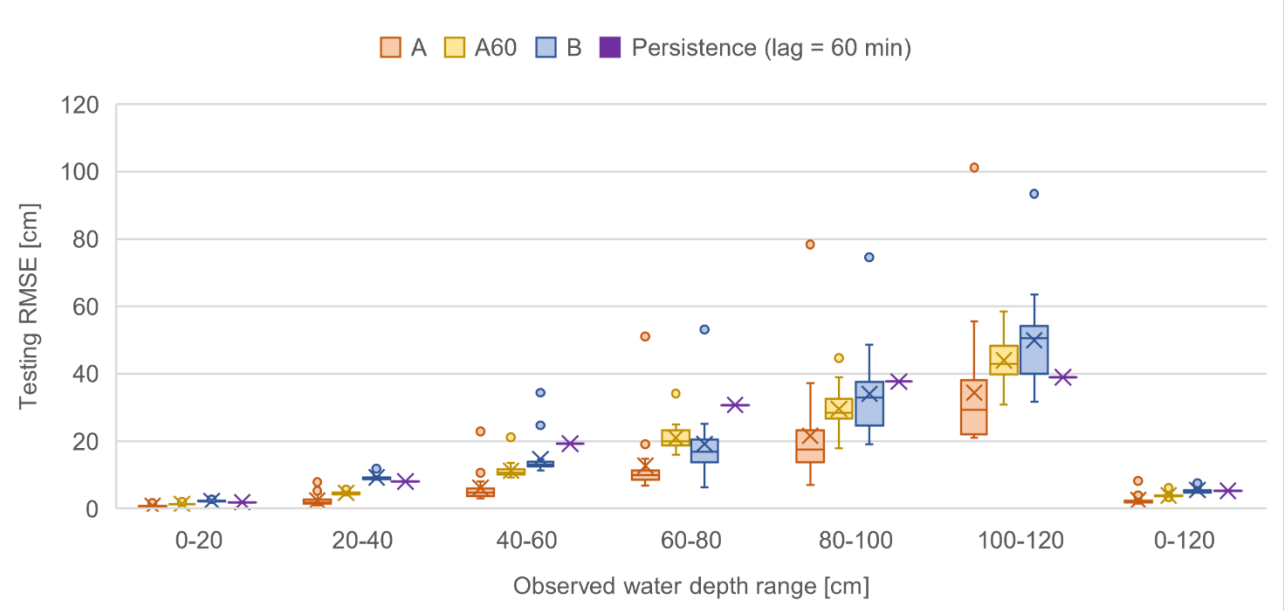

Figure 12. Breakdown of the testing RMSE in water depth ranges, for 20 LSTM networks with w = 120 min, p = 9 months and inputs A, B, and A_{60} (A with g = 60 min). Results presented are quartiles (box) and mean (cross). The whiskers extend to the minimum and maximum value or 1.5 times the inter quantile range. All values outside the whiskers range are marked as outliers (circles).

To better visualize the scores reported above, we plot a selected period of LSTM predictions for the same three input configurations (A, A_{60} and B with w = 120 min, p = 9 months) in Figure 13. The four-day long period roughly included two days of dry weather and two days of wet weather, as seen from the rainfall intensity series (Figure 13, top panels). The LSTM network appeared capable of capturing both the dry and wet weather behaviors for all three configurations. A similar variability of the prediction accuracy is seen for all cases, and in one instance with input configuration B the prediction is a horizontal line, i.e. invariant. This signals a possible failure in the training of the network.

The variability of the results across all 20 runs of the same LSTM configuration is better seen from the prediction residuals, which are given by the difference between the observed and predicted values. In theory, a threshold could be set on the residuals for the automatic detection of anomalous observations. However, such threshold needs to consider both the state dependency of the error and the sudden increase of the residual in correspondence to a delayed prediction. For comparison with the results shown above, the error is also presented as RMSE over the selected four-day long period (median RMSE = 3.28 cm for input B) or respectively for the two dry or wet days. As a matter of fact, the selected period can be considered as

representative of the average conditions of the CSO chamber, both in terms of dry weather baseline and wet weather intensity.

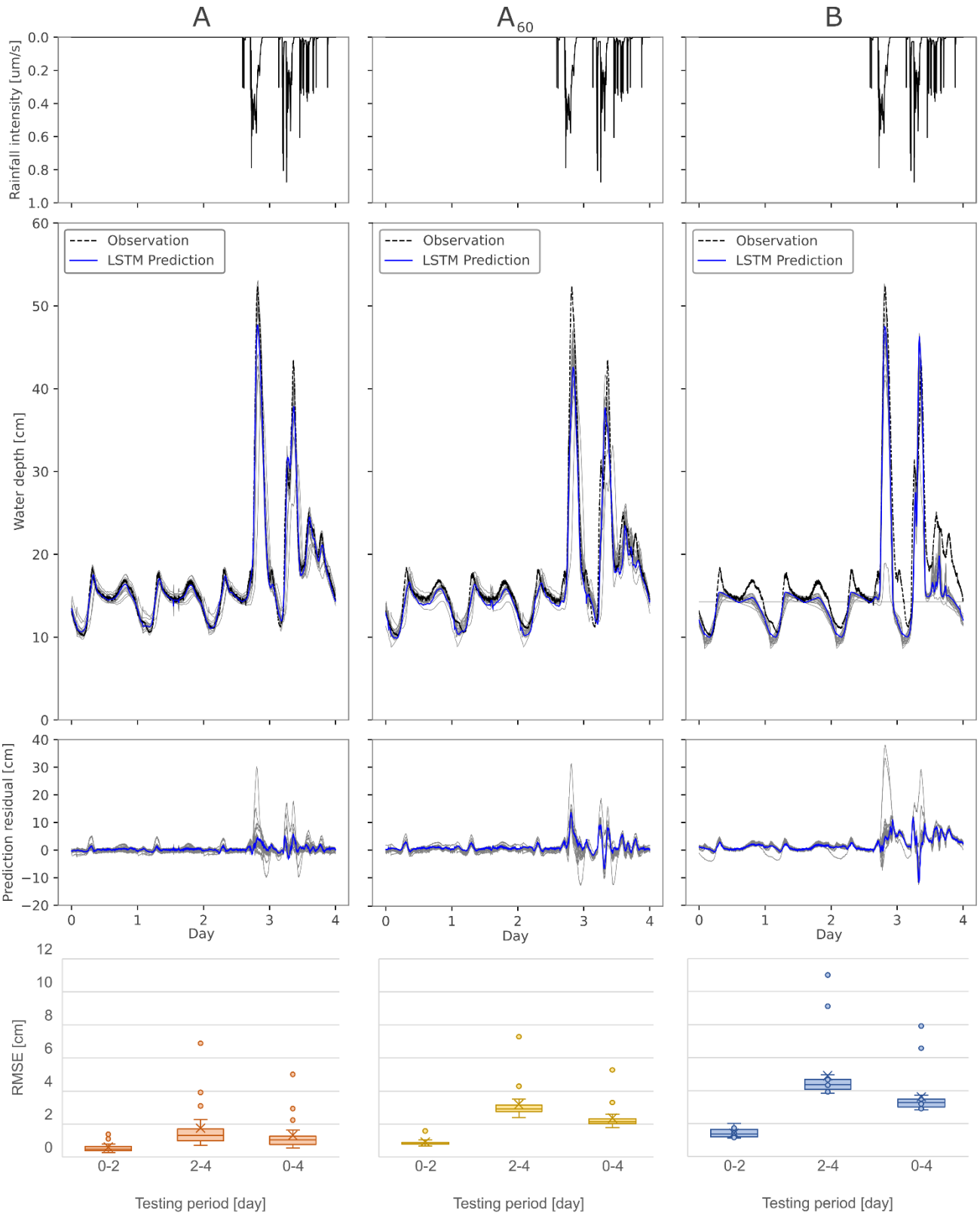

Figure 13. 20 LSTM predictions for selected 4 days of testing dataset with w = 120 min, p = 9 months and inputs A, B, and A_{60} (A with g = 60 min). From top to bottom: rainfall intensity observations; LSTM predictions (median shown in thick blue line) and corresponding observations (dashed); difference between observed and predicted values (median shown in thick red line); Root Mean Square Error calculated for dry, wet or all days in the selected period.

6. Conclusions

We tested the accuracy of LSTM predictions of water depths in a combined sewer overflow chamber in scenarios of limited information on the past observations of the system. Among a range of tested hyperparameter configurations, one with 2 hidden layers, 64 neurons per layer, a batch size of 2048 and a 25% dropout rate yielded the lowest error in terms of RMSE. Multiple training runs on the same data showed variability in terms of testing RMSE, with some runs failing, which illustrates that LSTM training is not a straightforward task.

When antecedent observations of water depth were known to the neural networks, we observed a decrease in prediction accuracy as less recent information was added as input. This shows that the LSTM network generally recognized the high correlation between the current observation and the most recent ones and assigned low weights to all other inputs. For comparison, using the most recent observation as prediction when available (persistence) yielded a lower error than any of the LSTM networks tested.

If gaps of different durations were introduced in the antecedent water depth observations, the LSTM network learned to compensate for the missing information with the other available input features. We observed a crossover point at a gap of 60 min for our system when the LSTM network outperformed the persistence model with a lag of same length. In other words, using the prediction from the LSTM network with 1h gap in the antecedent water depth was on average better than assuming the value to be the same as the observations from 1h before. With this approach, however, a different LSTM network is needed for different gap lengths and the corresponding computational overhead may or may not be justified depending on the application.

When the antecedent water depth was removed from the input features, we observed a further decrease in prediction accuracy. However, the prediction accuracy in this last scenario was qualitatively comparable to the other two, and hence might be sufficient in many soft sensing applications. The results also suggest a state dependency of the error, implying that LSTM is worse at predicting wet weather values compared to dry weather. This may be a consequence of the dataset unbalance, as the number of observations in dry weather far outnumber the wet weather, but can also be due to the fact that the processes governing wet weather discharges are related to uncertain input data, such as rainfall data, and complex runoff processes which are not identified by the network. The LSTM network generally outperformed the 60 min persistence model for non-overflowing wet weather conditions.

Future research could investigate other scores for quantifying the prediction accuracy, as RMSE tends to penalize wet weather errors and gives disproportionate large penalties on delayed predictions. Also, a better calibration of the hyperparameters and adding one that includes the learning rate of the optimizer could help reducing the variability of the results, thus attributing them more generality. Finally, the proposed approach to model update (re-learning) did not show significant improvements compared to a non-update model, so other approaches could be investigated.

Acknowledgements

We thank Lone Bo Jørgensen from the Greater Copenhagen Utility (HOFOR) and the Danish Meteorological Institute (DMI) for providing data for the case study. We also thank Alvin W. Z. Chew and Laura Rieger for their valuable guidance on the network setup. This work is part of a joint PhD programme between the Technical University of Denmark and the Nanyang Technological University, Singapore.

References

Ayazpour, Z., Bakhshipour, A.E., Dittmer, U., 2019. Combined Sewer Flow Prediction Using Hybrid Wavelet Artificial Neural Network Model, in: New Trends in Urban Drainage Modelling. Springer International Publishing, pp. 693–698. https://doi.org/10.1007/978-3-319-99867-1_120

Bailey, J., Harris, E., Keedwell, E., Djordjevic, S., Kapelan, Z., 2016. The Use of Telemetry Data for the Identification of Issues at Combined Sewer Overflows. Procedia Engineering 154, 1201–1208. https://doi.org/10.1016/j.proeng.2016.07.524

Bengio, Y., Simard, P., Frasconi, P., 1994. Learning Long-Term Dependencies with Gradient Descent is Difficult. IEEE Transactions on Neural Networks 5, 157–166. https://doi.org/10.1109/72.279181

Breinholt, A., Thordarson, F.Ö., Møller, J.K., Grum, M., Mikkelsen, P.S., Madsen, H., 2011. Grey-box modelling of flow in sewer systems with state-dependent diffusion. Environmetrics 22, 946–961. https://doi.org/10.1002/env.1135

Bruen, M., Yang, J., 2006. Combined hydraulic and black-box models for flood forecasting in urban drainage systems. Journal of Hydrologic Engineering 11, 589–596. https://doi.org/10.1061/(ASCE)1084-0699(2006)11:6(589)

Chollet, F., 2015. Keras. https://keras.io

Darsono, S., Labadie, J.W., 2007. Neural-optimal control algorithm for real-time regulation of in-line storage in combined sewer systems. Environmental Modelling and Software 22, 1349–1361. https://doi.org/10.1016/j.envsoft.2006.09.005

Djebbar, Kadota, 1998. Estimating sanitary flows using neural networks. Water Science and Technology 38. https://doi.org/10.1016/S0273-1223(98)00752-5

Duncan, A.P., Keedwell, E.C., Djordjević, S., Savić, D.A., 2013. MACHINE LEARNING-BASED EARLY WARNING SYSTEM FOR URBAN FLOOD MANAGEMENT 22, 3697–3711.

Eggimann, S., Mutzner, L., Wani, O., Schneider, M.Y., Spuhler, D., Moy De Vitry, M., Beutler, P., Maurer, M., 2017. The Potential of Knowing More: A Review of Data-Driven Urban Water Management. Environmental Science and Technology 51, 2538–2553. https://doi.org/10.1021/acs.est.6b04267

Elman, J., 1990. Finding Structure in Time, COGNITIVE SCIENCE.

Fernando, A.K., Zhang, X., Kinley, P.F., 2006. Combined Sewer Overflow forecasting with Feed-forward Back-propagation Artificial Neural Network. Transactions on Engineering, Computing and Technology 58–64.

Gong, N., Denoeux, T., Bertrand-Krajewski, J.-L., 1996. Neural networks for solid transport modelling in sewer systems during storm events. Water Science and Technology 33, 85–92. https://doi.org/10.2166/wst.1996.0183

Graziani, S., Xibilia, M.G., Rizzo, A., Fortuna, L., 2007. Soft sensors for monitoring and control of industrial processes, Springer. ed.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors.

Hochreiter, S., Schmidhuber, J., 1997. Long Short-Term Memory. Neural Computation 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jonsdottir, H., Nielsen, H.A., Madsen, H., Eliasson, J., Palsson, O.P., Nielsen, M.K., 2007. Conditional parametric models for storm sewer runoff. Water Resources Research 43. https://doi.org/10.1029/2005WR004500

Kingma, D.P., Ba, J., 2015. Adam: {A} Method for Stochastic Optimization, in: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Lobbrecht, A.H., Solomatine, D.P., 2002. Machine learning in real-time control of water systems. Urban Water 4, 283–289. https://doi.org/10.1016/S1462-0758(02)00023-7

Loke, E., Warnaars, E.A., Jacobsen, P., Nelen, F., Almeida, M. do C., 1997. Artificial neural networks as a tool in urban storm drainage. Water Science and Technology 36, 101–109. https://doi.org/10.2166/wst.1997.0651

Mounce, S.R., Shepherd, W., Sailor, G., Shucksmith, J., Saul, A.J., 2014. Predicting combined sewer overflows chamber depth using artificial neural networks with rainfall radar data. Water Science and Technology 69, 1326–1333. https://doi.org/10.2166/wst.2014.024

Palmitessa, R., Mikkelsen, P.S., Law, A.W.K., Borup, M., 2020. Data assimilation in hydrodynamic models for system-wide soft sensing and sensor validation for urban drainage tunnels. Journal of Hydroinformatics. Advance online publication. https://doi.org/10.2166/hydro.2020.074

Rjeily, Y.A., Abbas, O., Sadek, M., Shahrour, I., Chehade, F.H., 2017. Flood forecasting within urban drainage systems using NARX neural network. Water Science and Technology 76, 2401–2412. https://doi.org/10.2166/wst.2017.409

Rosin, T., Romano, M., Keedwell, E., Kapelan, Z., 2019. Data Analytics for Automated Detection of Blockages in Sewers, in: 17th International Computing & Control for the Water Industry Conference, Exeter, United Kingdom. pp. 3–4.

Savić, D.A., Bicik, J., Morley, M.S., Duncan, A., Kapelan, Z., Djordjević, S., Keedwell, E.C., 2013. Intelligent urban water infrastructure management. Journal of the Indian Institute of Science 93, 319–336.

Schmidhuber, J., 2015. Deep Learning in neural networks: An overview. Neural Networks. https://doi.org/10.1016/j.neunet.2014.09.003

She, L., You, X. yi, 2019. A Dynamic Flow Forecast Model for Urban Drainage Using the Coupled Artificial Neural Network. Water Resources Management 33, 3143–3153. https://doi.org/10.1007/s11269-019-02294-9

Sufi Karimi, H., Natarajan, B., Ramsey, C.L., Henson, J., Tedder, J.L., Kemper, E., 2019. Comparison of learning-based wastewater flow prediction methodologies for smart sewer management. Journal of Hydrology 577. https://doi.org/10.1016/j.jhydrol.2019.123977

Sumer, D., Gonzalez, J., Lansey, K., 2007. Real-time detection of sanitary sewer overflows using neural networks and time series analysis. Journal of Environmental Engineering 133, 353–363. https://doi.org/10.1061/(ASCE)0733-9372(2007)133:4(353)

Zhang, D., Lindholm, G., Ratnaweera, H., 2018. Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. Journal of Hydrology 556, 409–418. https://doi.org/10.1016/j.jhydrol.2017.11.018