



Accelerating hydrodynamic simulations of urban drainage systems with physics-guided machine learning

Palmitessa, Rocco; Grum, Morten; Engsig-Karup, Allan Peter; Löwe, Roland

Published in:
Water Research

Link to article, DOI:
[10.1016/j.watres.2022.118972](https://doi.org/10.1016/j.watres.2022.118972)

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Palmitessa, R., Grum, M., Engsig-Karup, A. P., & Löwe, R. (2022). Accelerating hydrodynamic simulations of urban drainage systems with physics-guided machine learning. *Water Research*, 223, Article 118972. <https://doi.org/10.1016/j.watres.2022.118972>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Accelerating hydrodynamic simulations of urban drainage systems with physics-guided machine learning

Rocco Palmitessa^a, Morten Grum^b, Allan Peter Engsig-Karup^c, Roland Löwe^{a,*}

^a Technical University of Denmark, Department of Environmental and Resources Engineering, Section of Climate and Monitoring, Miljøvej B115, 2800 Kgs. Lyngby, Denmark

^b WaterZerv, Fjenneslevvej 23, st., Brønshøj 2700, Denmark

^c Technical University of Denmark, Department of Applied Mathematics and Computer Science, Section of Scientific Computing, Asmussens Allé, 303B, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:

Hydrodynamic simulation
Scientific machine learning
Surrogate model
Urban drainage

ABSTRACT

We propose and demonstrate a new approach for fast and accurate surrogate modelling of urban drainage system hydraulics based on physics-guided machine learning. The surrogates are trained against a limited set of simulation results from a hydrodynamic (HiFi) model. Our approach reduces simulation times by one to two orders of magnitude compared to a HiFi model. It is thus slower than e.g. conceptual hydrological models, but it enables simulations of water levels, flows and surcharges in all nodes and links of a drainage network and thus largely preserves the level of detail provided by HiFi models. Comparing time series simulated by the surrogate and the HiFi model, R^2 values in the order of 0.9 are achieved. Surrogate training times are currently in the order of one hour. However, they can likely be reduced through the application of transfer learning and graph neural networks. Our surrogate approach will be useful for interactive workshops in initial design phases of urban drainage systems, as well as for real time applications. In addition, our model formulation is generic and future research should investigate its application for simulating other water systems.

1. Introduction

Computational modelling is essential in all phases of managing urban drainage systems (UDS), from design to monitoring and control. Physics-based deterministic models are widely used around the world. These models solve Saint Venant's system of equations in small time steps for each pipe element (Abbott and Ionescu, 1967). We will refer to this approach as "high-fidelity models". High-fidelity (HiFi) models are directly linked to physical system characteristics such as pipe diameter and length, and both water levels and flows are simulated in a realistic manner which makes them attractive to practitioners. However, high simulation times limit stakeholder involvement in the design phase (Leskens et al., 2014), as well as the exploration of impacts of deep uncertainties about future climates and urban developments (Löwe et al., 2020). In addition, real-time applications such as warning and control usually require shorter simulation times than what is feasible with HiFi models.

To circumvent these problems, hydrologists have developed a number of so-called lower-fidelity surrogates (Razavi et al., 2012). In

particular, a variety of conceptual (or lumped) hydrological models were developed for UDS (Burger et al., 2016; Kroll et al., 2017; Moreno-Rodenas et al., 2018; Thrysoe et al., 2019). All these approaches reduce computation times by orders of magnitude. However, they achieve speedup at the cost of simplifying the simulated processes or lowering the spatiotemporal resolution. Typically, only flows are simulated (not water levels), and only few selected locations in the pipe network are considered. The practical consequences are that the existing surrogates are valid only for a limited range of purposes (e.g. sewer overflow but not flood risk), are not straightforward to set up automatically from pipe databases, and extrapolate poorly to a changing drainage system structure. This limitation is not present for cellular automata approaches that only simplify the mathematical description of the processes but preserve the level of detail of the model (Austin et al., 2014). However, similarly small simulation time steps as with a numerical simulator must be selected. Limited speed-ups in the order of factor 5 where therefore achieved.

Machine learning approaches have gained traction in hydrology. They are frequently applied in an input-output setting, including

* Corresponding author.

E-mail address: rolo@dtu.dk (R. Löwe).

settings where physical system characteristics are used as inputs. The resulting models can be transferable between catchments as demonstrated, for example, for rainfall runoff modelling (Kratzert et al., 2019) and flood predictions (Bentivoglio et al., 2022; Löwe et al., 2021). Efforts have also been made to ensure conservation of mass (Frame et al., 2022). Both in rainfall-runoff modelling and in hydraulics, data-driven approaches have also been integrated into numerical solvers, with the aim of either simplifying numerical process equations, uncovering unknown relationships or achieving better model fits. Techniques such as genetic programming (Danandeh Mehr et al., 2018), neural networks (Höge et al., 2022) and curve fitting (Jamali et al., 2021) were used. While this approach can yield better fit with data and preserves physical interpretability, it is subject to similar limitations as the cellular automata approach in terms of computational performance. For example, (Jamali et al., 2021) achieved very limited reductions in simulation time for a problem involving solutions of the 2D shallow water equations.

In summary, there is a gap for surrogate approaches that simulate the full hydraulics for the often many thousand states included in a HiFi model, while still achieving substantial reductions in simulation time. In this paper, we argue that scientific (or physics-informed) machine learning (Karniadakis et al., 2021; Willard et al., 2020) is an avenue worth investigating and we develop a proof of concept for this purpose. Other than purely data-driven methods, scientific machine learning seeks to integrate machine learning and established methods from scientific computing and statistics, in order to utilize domain knowledge and known physical information in the learning process (Baker et al., 2019). There is a rapid development of scientific machine learning methods that model differential and partial differential equation systems using both time-discrete (Chen and Xiu, 2021; Geneva and Zabararas, 2020) and time-continuous approaches (Raissi et al., 2019; Wang et al., 2021). These are partly based on earlier ideas (Hunt et al., 1992; Lagaris et al., 1998), but embrace new contexts with technologies that were not available before.

The computational efficiency of neural networks enables models with large numbers of state variables. This ability is relevant for UDS, where we often want to simulate water levels and flows in hundreds or thousands of links and nodes. Further, scientific machine learning models learn interrelations between multiple states (e.g. water levels in neighboring nodes), and approaches that enable the exploitation of time series trends to increase simulation accuracy exist (Geneva and Zabararas, 2020; Ren et al., 2022). For hydraulics, these properties may enable the consideration of larger simulation time steps than presented in the literature so far. Finally, scientific machine learning enables multiple ways to incorporate physical system understanding into a data-driven model, e.g. in the form of physics-guided model architectures, mathematical constraints that enforce physical behavior, or physics-informed loss functions (Lu et al., 2021; Wang et al., 2021; Willard et al., 2020).

For this paper, our ambition is to:

- 1 Develop a proof of concept for simulating the hydraulics of UDS using scientific machine learning. To our knowledge, this has not been attempted before. It enables fast surrogates that distinguish themselves from existing lumped approaches through high spatial detail and by considering flows, levels and surcharges.
- 2 Specifically, we build on generalized residue networks (Chen and Xiu, 2021). As suggested by (Garzón et al., 2022), we incorporate inductive bias by designing a model architecture that is aligned with physical intuition about the hydraulic behavior of the pipe system.
- 3 Illustrate potentials as well as current bottlenecks and pitfalls when applying these techniques in a realistic setting with time-varying rainfall inputs and complex pipe flow dynamics
- 4 Draw up further research directions based on current developments in the literature.

2. Material and methods

2.1. Principal concepts

Our aim is to construct a fast surrogate model for 1D hydrodynamics in pipes that enables the simulation of both water levels and flows in all nodes and links of an UDS, much like existing HiFi models. HiFi models for UDS's typically simulate catchment runoff independently. Subsequently, it is routed through the nodes into the drainage network where hydraulics are simulated by numerically solving the Saint Venant equations. The runoff simulation typically has marginal computational cost, while the network simulation is computationally expensive. Thus, we propose a surrogate model of the drainage network hydraulics only. The hydraulic states of interest for each node in the drainage network are the water level in the node (h), the flow in upstream and downstream links (Q), as well as excess flows (Q_w), for which we later will distinguish between frequently activated overflows ($Q_w, \text{Overflow}$), and infrequently activated surcharges that cause pluvial flooding ($Q_w, \text{Surcharge}$) (Fig. 1a).

Because we focus only on the network component, the surrogate model is given the runoff R as input and only predicts h , Q and Q_w (Fig. 1b). Moreover, the model is structured autoregressively: The model predicts how the hydraulic states change from one time step to another, given a vector of runoff inputs during this time step. The predicted states are then used as the starting point for the next time step. Thus, the model learns how the hydraulic states change over a time interval Δt , considering surface runoff as a boundary condition. Once initialized, the surrogate can independently simulate time periods of any duration, given a series of runoff inputs. Similar autoregressive approaches were proposed earlier (Hunt et al., 1992), but have not been adapted for hydrodynamics.

We will consider a surrogate time step Δt of 1 min, which is well above the stability criteria for the HiFi model. The surrogate is trained against a limited set of simulation results from a HiFi model (so-called labeled data).

2.2. Model formulation

We start by introducing the concept of generalized residue networks, then we provide a model formulation for urban drainage systems, and finally we introduce physical system understanding into this model.

2.2.1. Generalized residue networks (gResNet)

Consider a dynamical system posed as an initial value problem with state vector \mathbf{x} that changes over time t :

$$\frac{d\mathbf{x}}{dt} = f(\mathbf{x}), \quad t > t_0, \quad \mathbf{x}(t_0) = \mathbf{x}_0 \quad (1)$$

We define the flow map as a real-valued function that maps how the state vector of the system changes from time t to $t + \Delta t$. (Qin et al., 2019) suggested that this flow map can be approximated by the function

$$\mathbf{x}_{t+\Delta t} = \mathbf{x}_t + N(\mathbf{x}_t, \Theta), \quad (2)$$

where N is a deep feed forward neural network with parameters Θ . The principal idea of this approach is that the solution of the differential equation system in Eq. (1) is approximated by an Euler scheme in discrete time steps Δt , using an identity mapping of the model states and a deep neural network that learns the “residue” from data, thereby capturing the system dynamics without knowing the true $f(\mathbf{x})$ a priori. Initializing from some starting states \mathbf{x}_0 , Eq. (2) can be used to simulate the system forward in time. (Chen and Xiu, 2021) extended the approach by replacing the identity mapping in Eq. (2) with a flexible operator mapping $\mathbf{x}_{t+\Delta t} \approx L(\mathbf{x}_t)$ that facilitates training of the model. L can in principle be any model approximation for the system. When L is unknown, they suggest that dynamic mode decomposition (Schmid, 2010) is a good guess, which in turn resembles a feed forward neural network L

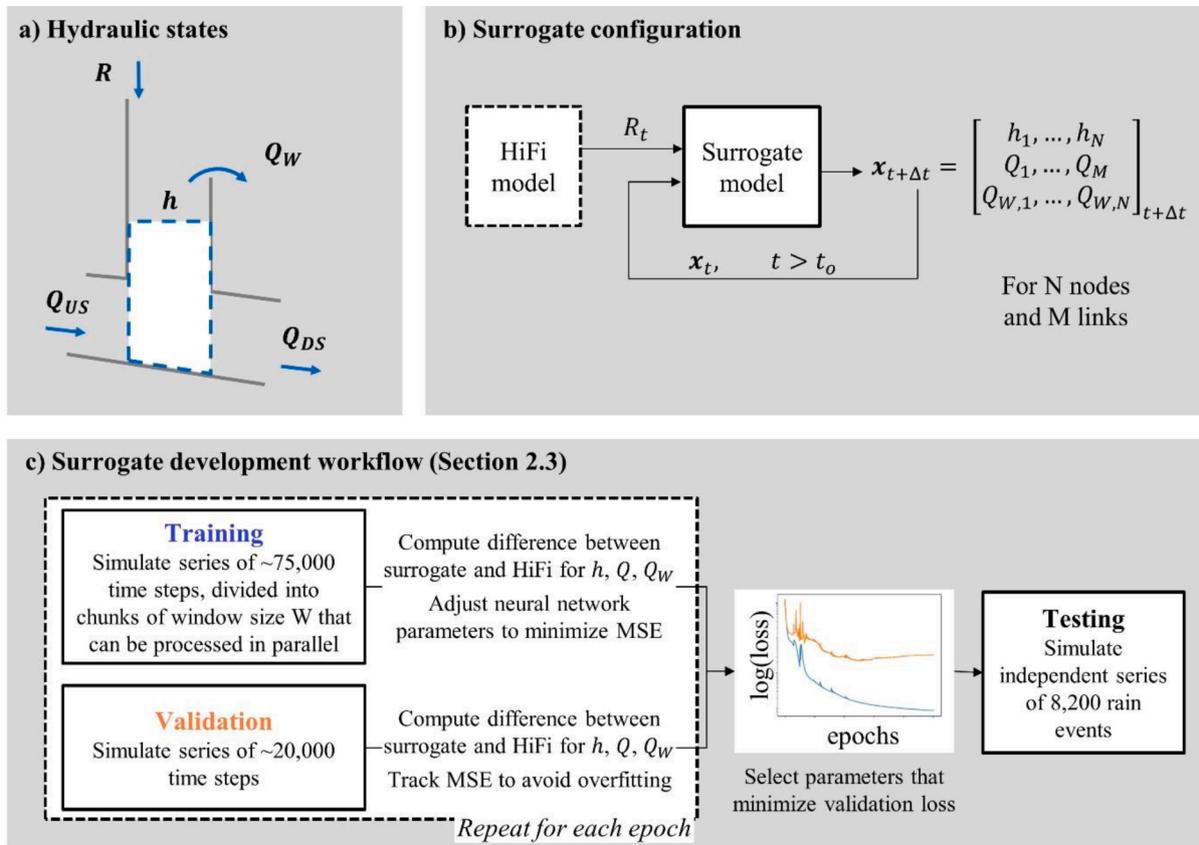


Fig. 1. a): hydraulic states of interest in the urban drainage model are the catchment runoff (R), the water level in nodes (h), the flow (Q) in the links upstream and downstream to the node, and the excess flow (Q_w). b): in the proposed configuration, the runoff and the initial states simulated with the HiFi model are given as input to the surrogate model; after initialization, the surrogate model predictions of water levels and flows at the previous time step are fed back as input for the length of the runoff series. c) Surrogates are validated on an independent series during training, and the final model is tested on a third series.

with a single hidden layer. This leads to the formulation of generalized residue networks (gResNet), where L is a neural network with a single hidden layer:

$$x_{t+\Delta t} = L(x_t, \Theta_L) + N(x_t, \Theta_N). \quad (3)$$

2.2.2. Generalized residue networks for UDS

The model formulation in Eq. (3) is attractive for the simulation of urban drainage networks because it allows for the consideration of a large number of states in the vector x , because we can consider various formulations of the neural network N that reflect different system complexities, and because we can impose physical constraints on model predictions that will be introduced in the next section. Assuming that we want to use Eq. (3) to simulate an UDS with N nodes and M links, we consider a state vector that includes the water levels h in all nodes, the flows Q through all links, and the excess flows Q_w from all nodes (Fig. 1a):

$$x_t = [h_1, \dots, h_N, Q_1, \dots, Q_M, Q_{w,1}, \dots, Q_{w,N}]. \quad (4)$$

The state changes from one time step to another, depending on the boundary conditions imposed on the system, i.e. the vector of runoff volumes $R_t = [R_1, \dots, R_N]$ that enter the nodes in the considered time interval. We therefore consider R_t as an additional input to the deep neural network N , while we preserve the purely state dependent formulation of the prior model L :

$$x_{t+\Delta t} = L(x_t, \Theta_L) + N(x_t, R_t, \Theta_N). \quad (5)$$

Starting from some initial state values x_0 , this model simulates water levels, pipe flows and surcharge flows in time intervals Δt for each node and each link for which the model is trained.

2.2.3. Physics-constrained generalized residue networks for UDS

Predictions from the model in Eq. (5) are in no way constrained to a range that is physically reasonable. (Beucler et al., 2021) suggested in a similar setting that one way of ensuring physically appropriate behavior of the model is to introduce so-called constraint layers C . These can be interpreted as a post-processing step after each prediction, and calculate certain model states based on a physical understanding of the system:

$$x_{t+\Delta t} = C(L(x_t, \Theta_L) + N(x_t, R_t, \Theta_N)). \quad (6)$$

An import learning from initial experiments with Eq. (5) was that surcharge flows $Q_{w, \text{Surcharge}}$ were difficult to learn for a purely data-driven model. However, surcharge flows occur only when the water level in a node reaches the ground level, and the connected pipes exceed their capacity. In this situation, the water volume stored in the pipe network remains constant, and $Q_{w, \text{Surcharge}}$ can be computed from a mass balance calculation that is performed locally for each node. Therefore, we suggest a revised model formulation where the state vector Eq. (4) is reduced to only include water levels and pipe flows:

$$x_t = [h_1, \dots, h_N, Q_1, \dots, Q_M] \quad (7)$$

Surcharges are subsequently calculated in a “constraint layer” as the difference between all the inflows to a node i from upstream pipe elements (considering the link flow values predicted by the gResNet) and runoff, and the outflow to all downstream pipe elements. In addition, we enforce a minimum of 0 for surcharge flows, because we perform HiFi simulations in “spilling configuration”, i.e. surcharging water does not reenter the drainage system. This configuration was shown to yield more realistic representations of pressure levels in intense rain storms than the widely used “ponding configuration” (Jamali et al., 2018):

$$Q_{w,i} = \max\left(\sum_j Q_{US,i,j} - \sum_k Q_{DS,i,k} + R_i, 0\right) \quad (8)$$

Note that while we have removed Q_w from the state vector, we still consider all states h , Q and Q_w when computing the deviation between HiFi model and surrogate during training (Section 0). Thus, the mass balance computation affects the parameter estimates of the gResNet.

2.3. Surrogate training

The proposed model includes parameter sets Θ_L and Θ_N that need to be estimated. For this purpose, we simulate rainfall series of limited duration in a HiFi model and extract time series of the simulated hydraulic states for all nodes and links in 1 min resolution. The surrogate parameters are estimated by minimizing the mean squared error (MSE) between the “labels” generated by the HiFi model and the corresponding hydraulic states simulated by the surrogate. When computing MSE, all three state variables h , Q and Q_w are considered for all nodes and links, irrespective of the model version.

All states are scaled to the range [0,1] before computing MSE, to ensure that all state variables have approximately equal impact on the parameter estimates. Scaling is performed using a min-max approach (Pedregosa et al., 2011), and the scaling range is defined individually for each state variable, based on the minimum and maximum values observed in the HiFi simulations.

As illustrated in Fig. 1c, surrogate development is divided into training, validation and testing steps. To enable faster processing, the training series is divided into chunks (windows) that can be simulated in parallel. The surrogate is initialized from the HiFi results at the beginning of a window and then independently simulates the hydraulic states for a number of time steps, corresponding to the so-called window size. A window size of one time step maximizes the potential for parallel processing and will thus yield the fastest training. However, it also entails a high risk of overfitting, because the surrogate is initialized from the true values at each time step, and thus never forced to generate stable simulations in the training phase. The window approach implies that the surrogate is trained considering a variety of starting values for the states. In the validation and testing steps, surrogate simulations are initialized from an empty pipe system.

2.4. Technical implementation

Surrogate models were implemented in Python 3.8 using Google’s Tensorflow library version 2.7.0 (Abadi et al., 2016). Parameter estimation was performed over 2,000 epochs (or optimizer iterations) using

the Adam algorithm (Kingma and Ba, 2014). The learning rate is a key parameter that affects how aggressively the optimizer changes parameters. It was (by trial and error) configured with a starting value of 1e-3 which exponentially decayed to 1e-4 to avoid divergence in the end of the training (You et al., 2019). We stopped training preliminarily if the validation MSE did not decrease for 500 epochs in a row. Relu activation functions were used in the hidden layers of the neural networks based on initial experiments and because they are the current standard for deep neural networks (Goodfellow et al., 2016).

2.5. Design of experiments

2.5.1. Catchment and HiFi model

We considered two medium-sized test systems extracted from a real UDS in Bochum, Germany (Fig. 2). System 1 includes 60 pipes and a catchment area of 14 ha, while System 2 includes 87 links and a catchment area of 26 ha. Except for the outlets, all nodes are equipped with a spilling weir placed near ground level to simulate water discharged to the surface. System 1 includes an overflow at node 27793 with a weir crest just above the top of the pipe level. This overflow occurs more frequent than spills, which may affect surrogate training. System 2 includes a basin with a volume of 140 m³ close to the outlet. The basin outflow is throttled to a maximum of 0.6 m³/s, and the connected overflow weir crest is placed 3 m above invert level. The basin thus creates backwater in the surrounding pipe stretches. We used System 1 to develop the proposed surrogate configuration, while System 2 was used to validate the approach in a more challenging system with looped network connections and fast backwater dynamics. Detailed system properties are provided in the supporting information.

2.5.2. Training, validation and testing data

We generated training, validation and testing data for the development of surrogates by performing hydrodynamic simulations with the MIKE 1D engine (DHI, 2021). For surrogate training and validation, we performed hydrodynamic simulations using two different sets of rainfall series as input:

A We considered 40 years of rainfall observations in 1 min resolution from a gauge in Odense that is part of the Danish SVK network. We identified rain events (see Supporting Information) and manually selected 154 rain events for training and 43 rain events for validation. We aimed to include events with different temporal variations and intensities. The selected events were concatenated into one series, resulting in a training series consisting of 73,990 one-minute data points, and a validation series of 19,095 data points.

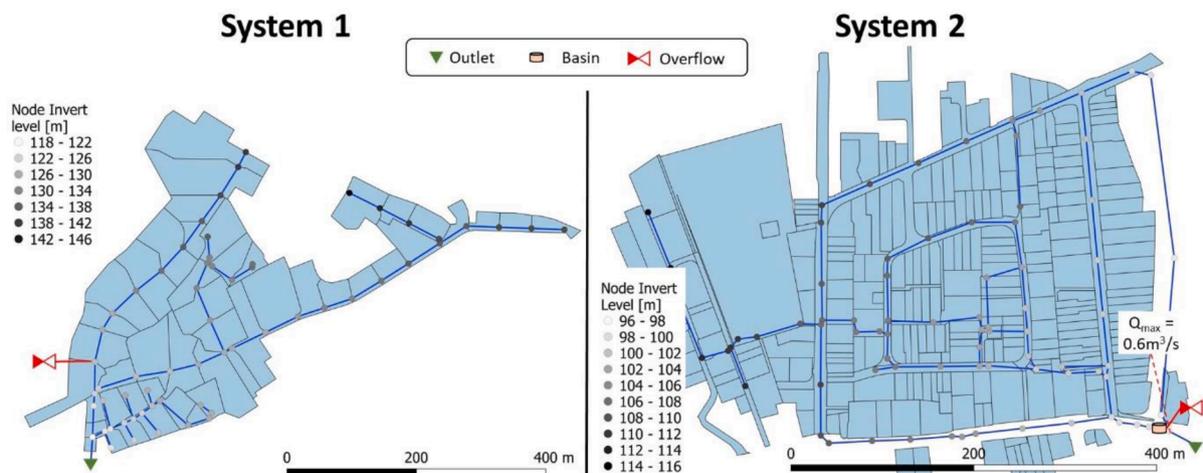


Fig. 2. Map view of test system, with nodes (grey dots), pipes (dark blue lines) and catchments (blue polygons). Nodes with higher invert elevation are marked with darker color. Note the combined sewer overflow in the bottom left of system 1, and the outflow-limited basin in the bottom-right of system 2.

B We aimed to create a rainfall series where extreme events were overrepresented to facilitate surrogate training for these situations. We repeated the above process, considering 10 rain gauges from various locations in Denmark, each with 40 years of rainfall observations. We manually selected 203 events for training and 58 events for validation from the pool of events. We considered all rainstorms where flooding would be likely to occur and generally focused our selection on intense rainstorms. The events were again concatenated into one series, resulting in a training series of 74,769 time steps and a validation series of 21,700 time steps.

All surrogates were tested by performing simulations for a rainfall series consisting of 40 years of continuous observations in Næstved, Denmark (8200 events). This series was not considered in the creation of datasets A or B. The gauge is located approximately 30 km away from the nearest gauge considered in the creation of dataset B and was intended as a fully independent test series.

2.5.3. Summary of experiments

The technical implementation of our surrogates has two main hyperparameters. These are the window size used for subdividing the training data series (see Section 2.3), as well as the complexity of the neural network N that is used to model the residue of the linear predictor L (see Eq. (5)). To identify a reasonable configuration, we compared MSE obtained on the validation dataset (so-called validation loss) and training times for the hyperparameter configurations shown in Table 1. This process was repeated five times for each combination of hyperparameters, because different (random) initializations of the parameters in the neural networks lead to slightly different validation losses.

Subsequently, we selected the configuration with the best compromise between validation loss and computational efficiency, and compared surrogate configurations with and without physical constraints that were trained on datasets A or B. Finally, the preferred configuration was tested on System 2.

2.6. Scoring criteria

To compare the surrogate simulations against HiFi simulations, we considered widely used scoring criteria that were computed for the testing series. We considered RMSE and R^2 to measure both average and event-based performance of the surrogates. All scores were computed in the untransformed space, i.e. the units of RMSE are meter for water levels and m^3/s for flows.

We evaluated the total excess flow volume $\sum_t Q_{w,i,t}$ for individual nodes during individual rain events, to assess whether overflows and surcharges were predicted in the correct locations and with accurate magnitude

Finally, to evaluate whether the model results are biased, we compared total runoff, surcharge, overflow and outflow volumes.

Table 1

Design of experiments. A reasonable complexity of the neural network and an appropriate training window size were selected in the first stage. Subsequently, performance of different surrogate versions was assessed, before the final surrogate was tested in system 2 (S2).

| | Sewer system (Fig. 1) | Physical constraints | Training series | Window sizes [min] | Neural network N specifications |
|--|-----------------------|----------------------|-----------------|---------------------|---|
| Stage 1 - Selection of hyperparameters | S1 | Without, With | A, B | 1, 10, 60, 120, 360 | S1 = 2 hidden layers of 10 neurons each S2 = 4 hidden layers of 20 neurons each S3 = 6 hidden layers of 50 neurons each S4 = 6 hidden layers of 100 neurons each |
| Stage 2 - Assess performance and effect of physical constraints and training series | S1 | Without, With | A, B | 60 | S4 |
| Stage 3 - Validate performance in a separate system | S2 | With | B | 60 | S4 |

3. Results

3.1. Hyperparameter selection

Fig. 3 illustrates the validation loss (smallest validation MSE obtained during training) for different combinations of neural network configuration and window size, considering surrogates that included physical constraints for surcharges and that were trained on series A. The results for this surrogate configuration illustrate the considerations arising during training. Results for other surrogate configurations are included in the supporting information.

Smaller window sizes allow faster training of the surrogates when parallel processing is exploited but increase the risk of overfitting. This issue is very clear from the figure, where models trained with window sizes of 1, and in some cases 10 min yield substantially larger validation losses. Surrogates with too few parameters lead to unstable validation results because they have difficulties learning the dynamics of the system. Considering also the results obtained for data series B (supporting information), network sizes S3 and S4 are preferable. Moving from neural network size S1 to S4, computation times per epoch on a single CPU increased moderately (in the order of 30%). Considering the more stable convergence properties of the more complex models, we moved forward with network size S4 and a window size of 60 min for the remaining study. Experiments with more complex neural networks did not yield better convergence in the tested sewer systems (Supporting Information).

3.2. Effects of training series and physical constraints on event-based performance

Fig. 4 shows RMSE computed for h , Q , $Q_{w,overflow}$ and $Q_{w,surcharge}$ for individual rain events in System 1. The presented results were generated from surrogates trained on data series B. Amongst the 5 training iterations for each model (Fig. 3), we show results for the best performing model without physical constraints (lowest validation loss) and the worst performing model with physical constraints (highest validation loss). We used the peak flow simulated by the HiFi model at the outlet to distinguish low and high flow conditions in the figure. A general increase of RMSE can be observed for high flow conditions. This may be linked to a change of dynamics in extreme flow situations and the relatively fewer occurrences of such situations even in data series B. Nevertheless, RMSE values for levels and flows are in the range of few cm and few l/s, respectively. This suggests that the surrogates generally achieve high accuracy. RMSE values for water levels and flows are slightly higher for the model considering physical constraints. In this model configuration, the simulated water levels are tied to the activation of mass balance computations for excess flows. Thus, the surrogate has less freedom to fit the water level data than the model without constraints. In contrast, the model including physical constraints achieves much lower RMSE values for $Q_{w,overflow}$ and $Q_{w,surcharge}$ (panels C and

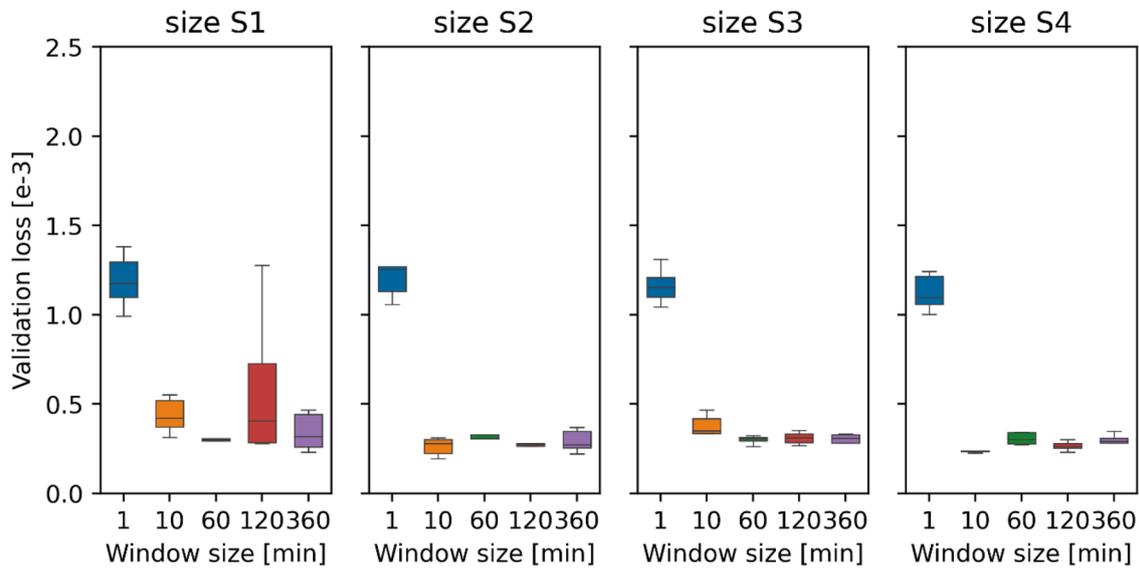


Fig. 3. Box plots of validation loss as a function of network size and input window size. The model included physical constraints on surcharges and was trained with rain events A (each configuration was tested 5 times).

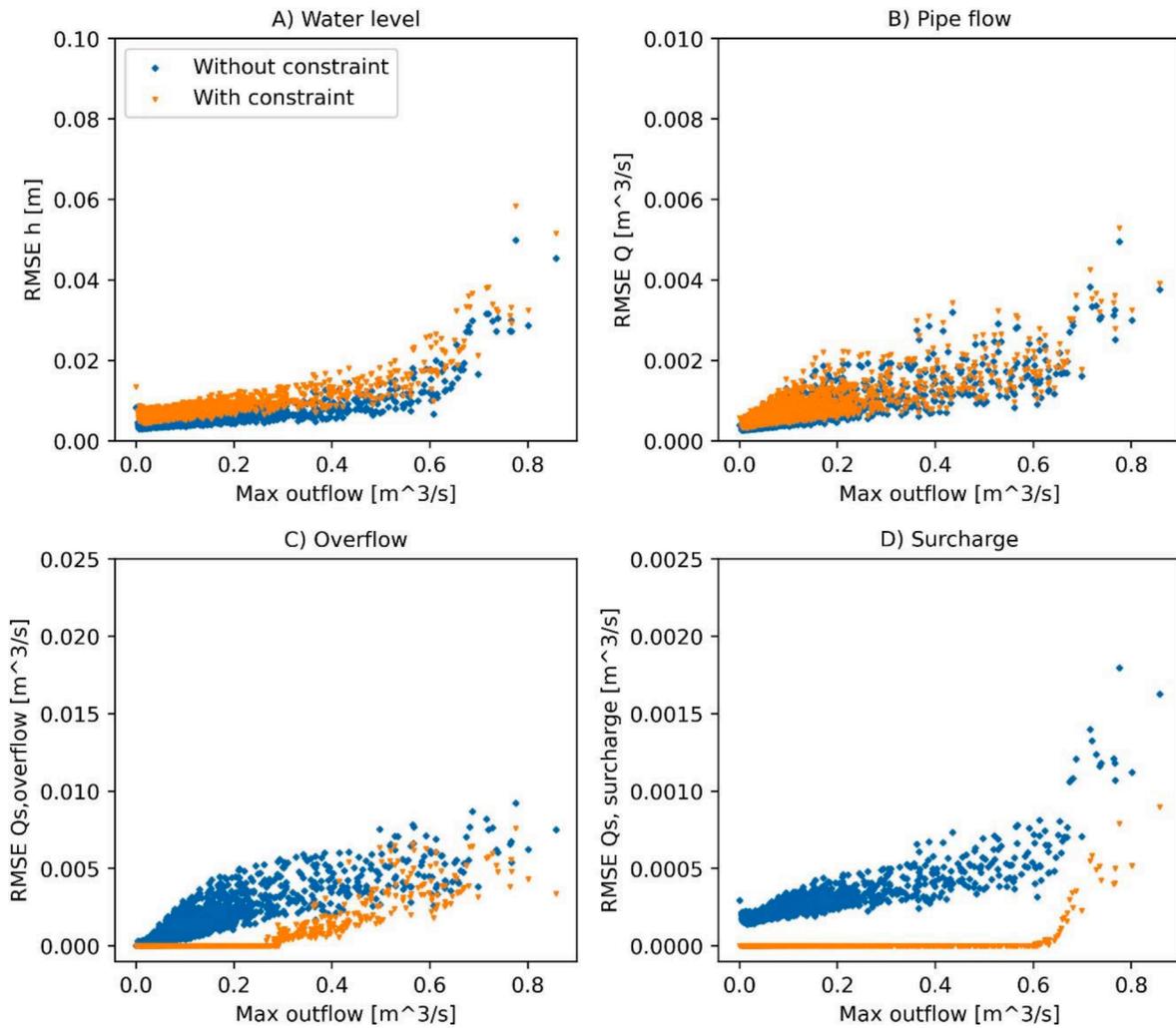


Fig. 4. RMSE of water level in nodes (A), discharge in pipes (B), discharge over overflow weir (C) and surcharge (D) in System 1 as a function of event intensity, expressed as maximum discharge at the outlet. Each dot represents a different rainfall event in the testing dataset. The results are shown for the emulator trained on data series B, with and without physical constraints.

D). Deviations in low flow conditions (where the HiFi model does not simulate excess flow) are entirely removed. In addition, smaller RMSE values are also achieved in periods where excess flows occur.

Fig. 5 analyses the behavior of predicted overflows and surcharges in System 1 more in depth. The figure compares total overflow and surcharge volumes for individual rain events in the testing dataset, considering each node individually. Panels A and C consider surrogates trained on data series A, while panels B and C consider surrogates trained on data series B.

Comparing the two surrogate configurations, we can conclude that the introduction of physical constraints leads to a much better representation of excess flows. However, comparing panels A and C against panels B and D, a clear effect of the training series is also noticeable. When trained on series A, predictions of overflow and surcharge volumes are slightly more uncertain, because these events are too sparse in the training data. The predicted water levels and link flows are then not properly tuned to enable an accurate mass balance calculation. Considering training series B, especially the errors for surcharges are much reduced, because these situations now occur sufficiently often during training to enable the surrogate to learn the corresponding dynamics.

3.3. Time-averaged test scores

Fig. 6 shows score values for the different hydraulic states, as well as total water volumes that enter and leave the models. We compare the score values obtained for different surrogate configurations in System 1 (Stage 2). In addition, we compare the performance of the surrogate configuration that was trained on series B and includes physical constraints between Systems 1 and 2. As already indicated by the previous results, imposing physical constraints led to improved simulations of surcharges and overflows, which is indicated by improved RMSE values (panel A), much improved R^2 values (panel B) and reduced volume errors (panel C). Comparing the two model configurations with physical constraints, the performance of the surrogates trained on different data series was very similar. However, surrogates trained on series B yielded slightly lower RMSE values and slightly higher R^2 values. Similar average score values were obtained for systems 1 and 2, with the exception of higher average RMSE scores that can be associated with larger flows in this system.

3.4. Surrogate performance in specific elements

Fig. 7 illustrates for the surrogate with physical constraints how the accuracy of simulated water levels and link flows varies in space. Figs. 8 and 9 show time series plots for the elements highlighted in Fig. 7. In the

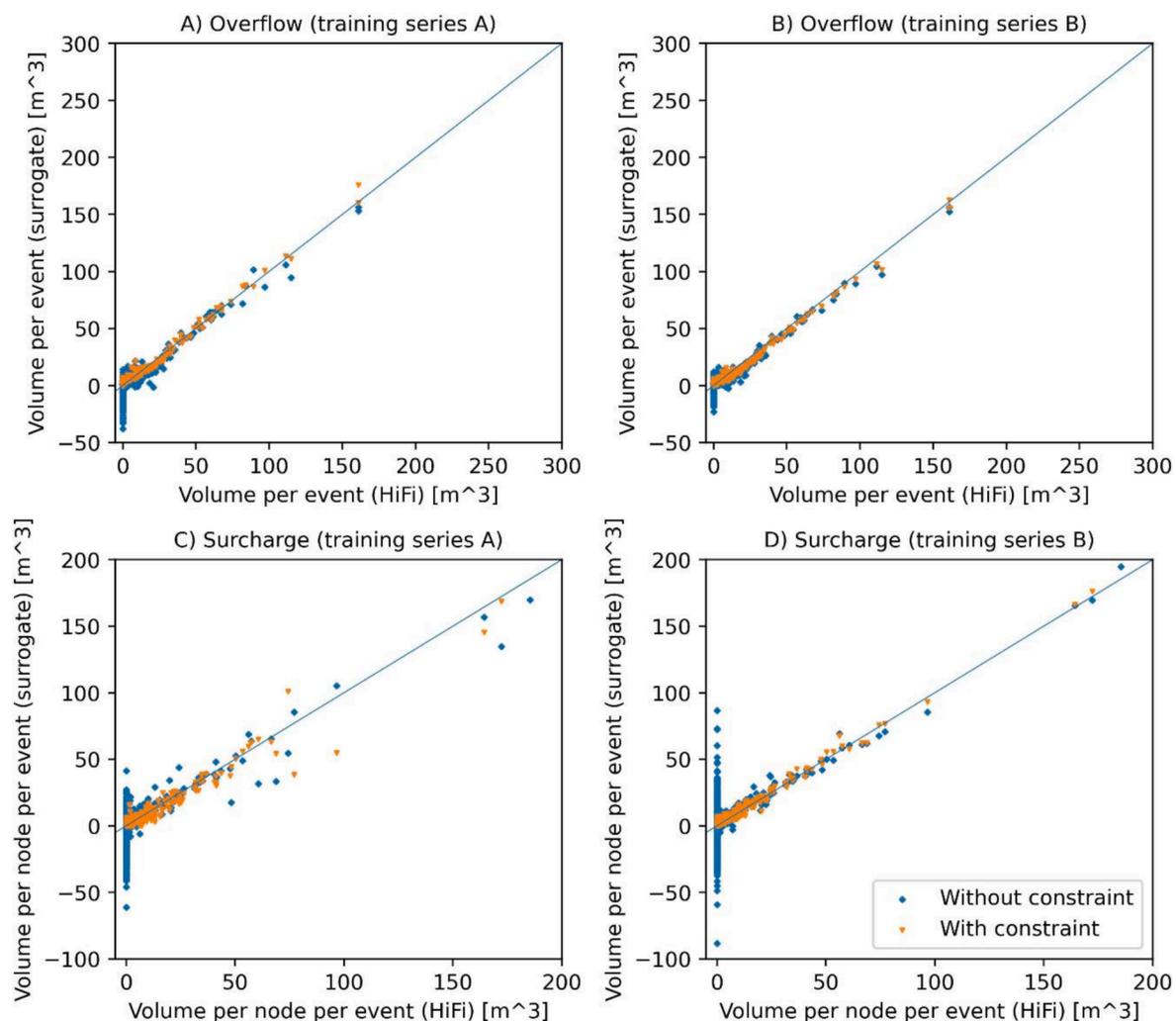


Fig. 5. Panels A and B: Total overflow volume simulated for weir 27793 in individual rain events by the HiFi model as well as by the emulator trained on datasets A and B. Panels C and D: Total surcharge volumes simulated by the HiFi model, and the emulator trained on datasets A and B. Each dot represents the surcharge volume simulated for an individual node during an individual rain event in the testing dataset.

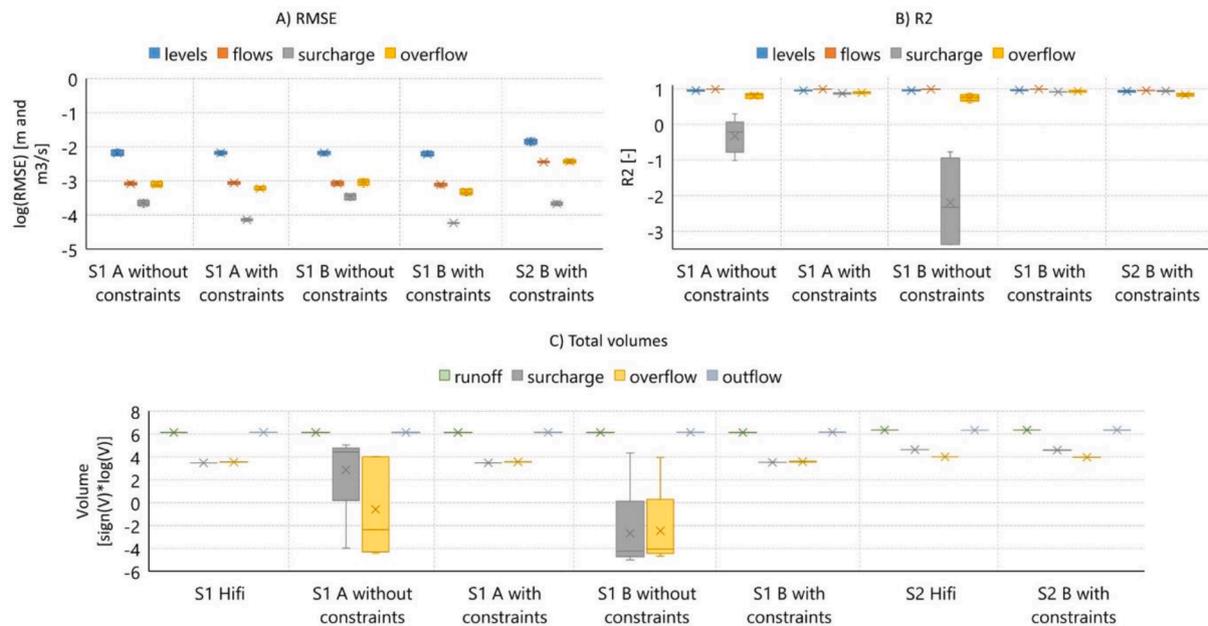


Fig. 6. Evaluation scores of emulator predictions for the test dataset, divided by metric (A: Root Mean Square Error, B: R^2) and hydraulic state (water level in nodes, flows in pipes, surcharge and overflow) for all four tested configurations in system 1 (S1), as well as for models with physical constraints that were trained on series B in system 2 (S2), C: total runoff, surcharge, overflow and outflow volumes simulated for the testing series. Panel C shows log-transformed absolute values that were multiplied by the sign of the original value. In all subfigures scores are arranged in the same order as shown in the corresponding legend.

time series plots, we again consider the worst performing surrogate across 5 training iterations.

Figs. 7 and 8 generally suggest strong performance in System 1. The hydrographs clarify that surrogates without physical constraints create false predictions of excess flows Q_w with both positive and negative signs. These issues are more pronounced for surcharges (panel D) than for overflows (panel C). The latter occur more frequently in the dataset and are thus easier to learn for the surrogate. The issue is avoided when introducing physical constraints for surcharge predictions into the surrogate. A limitation becomes clear from lower R^2 in some upstream links and nodes in the center of the catchment. In this area, the connected sub-catchments have very low imperviousness ratios, resulting in small, infrequent pipe flows. The surrogate can then not learn the relationship between inflow and hydraulics. This issue can be avoided by ensuring that training data are generated in such a way that high and low inflow situations occur in all parts of the system.

For System 2, Figs. 7 and 9 similar illustrate strong surrogate performance, except for locations where fast flow reversals occur. In these locations, the surrogate is only partly able to simulate the flow reversal and, quite similar to the surcharge results without physical constraints, creates false predictions of small flows with both positive and negative signs in low flow conditions that lead to very low R^2 values. Note that these issues do not affect surrogate performance in the remaining system.

3.5. Computation time

Table 2 compares computational effort for training and running the surrogates against the time required to simulate the testing series in the HiFi model. The surrogates are currently trained against results from a HiFi simulations. Therefore, we also included the time required to perform these simulations for the training and validation dataset (“label generation”). Tests were performed on an HP Zbook G9 with Intel Core i9-10885H processor and 32GB RAM. All tests were performed using a single CPU only.

Training times ranged between 0.5 and 2 h. Surrogates trained on data series A generally converged faster, which may be related to the less complicated dynamics in this series. Surrogates including physical

constraints were in the order of 10% faster than surrogates without physical constraints, because the surrogates included $\sim 33\%$ fewer state variables. Excess flows Q_w were instead computed efficiently in a postprocessing step.

Prediction times for the testing series (8,200 rain events) were in the order of 100 s and thus in a range that enables real time applications as well as interactive simulations in workshop settings. For HiFi simulations, we considered a fixed simulation time step of 5 s. Note that the simulation time step of a HiFi model can both be lower or higher, depending on the hydraulic conditions in the specific system, and if adaptive time stepping schemes (DHL, 2021) are employed. Surrogate simulation times will remain constant as long as the same simulation time step (here one minute) is considered. However, prediction accuracy may be reduced in some situations.

While we tested computation times in a single CPU setting, the surrogates can exploit parallel computing, which is important for implementation. We did not achieve reduced simulation times by employing GPUs. This is most likely due to the autoregressive character of the models, which implies that the technical implementation includes a loop over a time series. The higher processing speed of CPUs then outperforms the parallelization capacities of GPUs.

4. Discussion

4.1. Results and limitations

The results shown in the previous section illustrate that our surrogate modelling approach can simulate water levels, flows, overflows and surcharges much alike a state-of-the-art hydrodynamic model. The surrogates enable long time series simulations (in our case 8,200 rain events extracted from 40 years of rainfall observations) of water levels, flows and surcharges in all nodes and pipes of a drainage system within few minutes. While these results are promising, our work has several clear limitations that will need to be addressed in further research to make the approach applicable in practice:

- 1 Our test cases are examples of typical drainage networks, including overflows and surcharges, loops, and backwater situations, but all

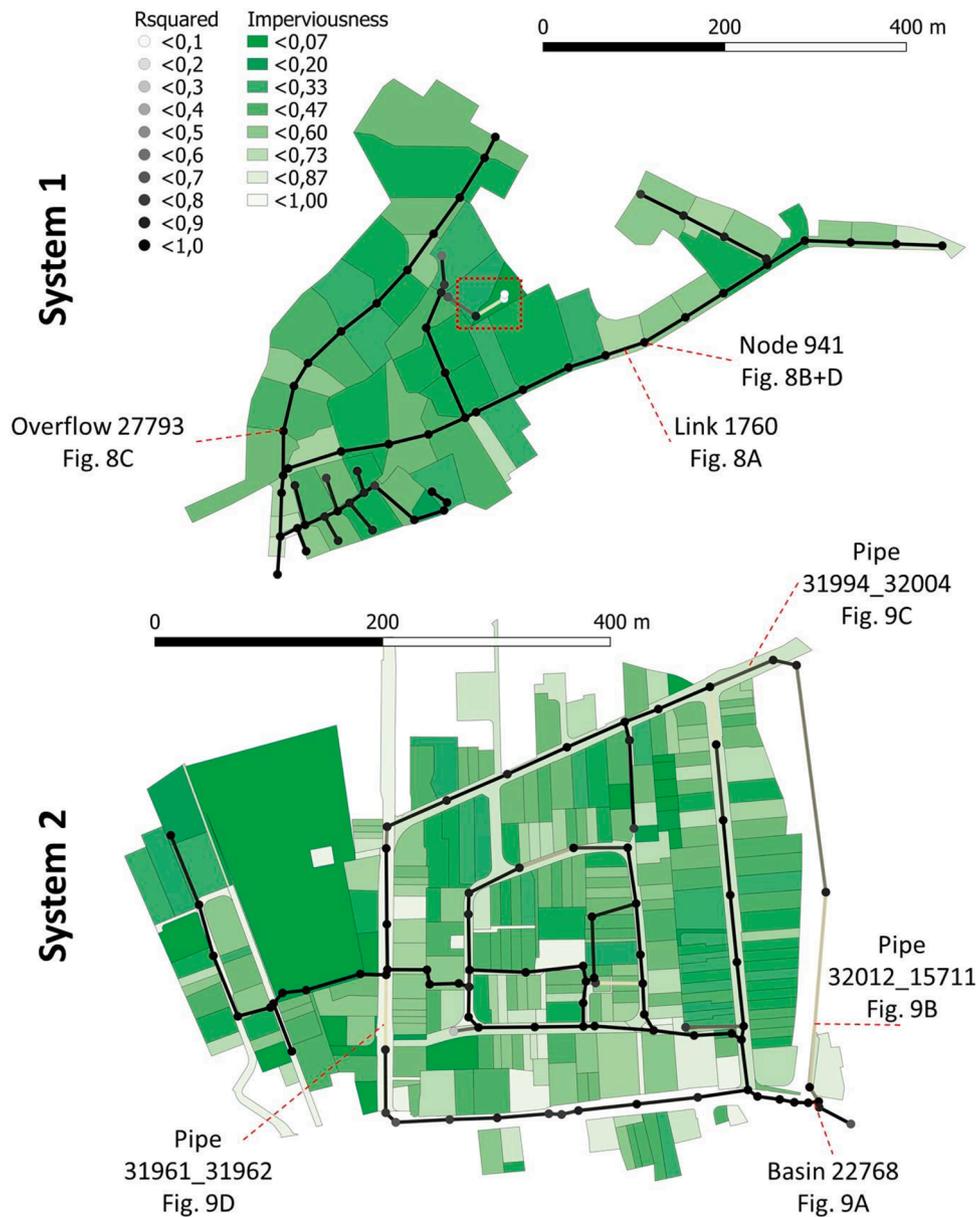


Fig. 7. R^2 values averaged over 5 training iterations for node water levels and link flows, considering the surrogate with physical constraints trained on series B. Catchments are colored according to which portion of the catchment area is impervious. The square in the center of System 1 marks areas with very infrequent flows due to low imperviousness.

pipes are circular and no controlled structures are included. The results suggest that the current surrogate configuration can only to a limited extent simulate fast flow reversal dynamics. This issue does, however, not propagate to other parts of the system because the surrogate, simplistically speaking, simulates hydraulics based on correlations between inputs and states, rather than iteratively.

Our current surrogate formulation may have difficulties to learn the dynamics in systems with very irregular cross sections (e.g. when simulating open water bodies), or trunk sewers that can be dominated by inflows in few selected points. Potential improvements are outlined in Section 4.3, but there will be hydraulic situations that a surrogate cannot fully simulate. These limitations need not impair its applicability for assessing, for example, surcharge frequencies. In practice, we can support the user by highlighting the accuracy of simulations for the validation series in all links and nodes.

2 One avenue for upscaling our approach to large drainage systems is to subdivide the drainage network into many small subsections, each of which is simulated by an independent sub-model (see Section 4.2).

Implementing such an approach requires that sub-models can be trained independently from each other, and subsequently generate accurate simulations for the combined systems. We have not documented such approaches.

- 3 We have not considered the effect of controlled actuators such as pumps or moveable weirs. In many cases, the water level in a controlled actuator can be derived using simple mass balance calculations that consider storage volume and the controlled outflow (Balla et al., 2022). Similar to the previous point, the main challenge is then to develop dedicated training approaches such that the surrogate learns to simulate the system dynamics given one or more water level boundaries.
- 4 We created artificial rainfall series for training the surrogates. These were defined based on hydrological intuition with the aim of representing the range of relevant dynamics. We have not investigated how many data points should be included in the training series. Structured approaches for selecting the rain events included in the

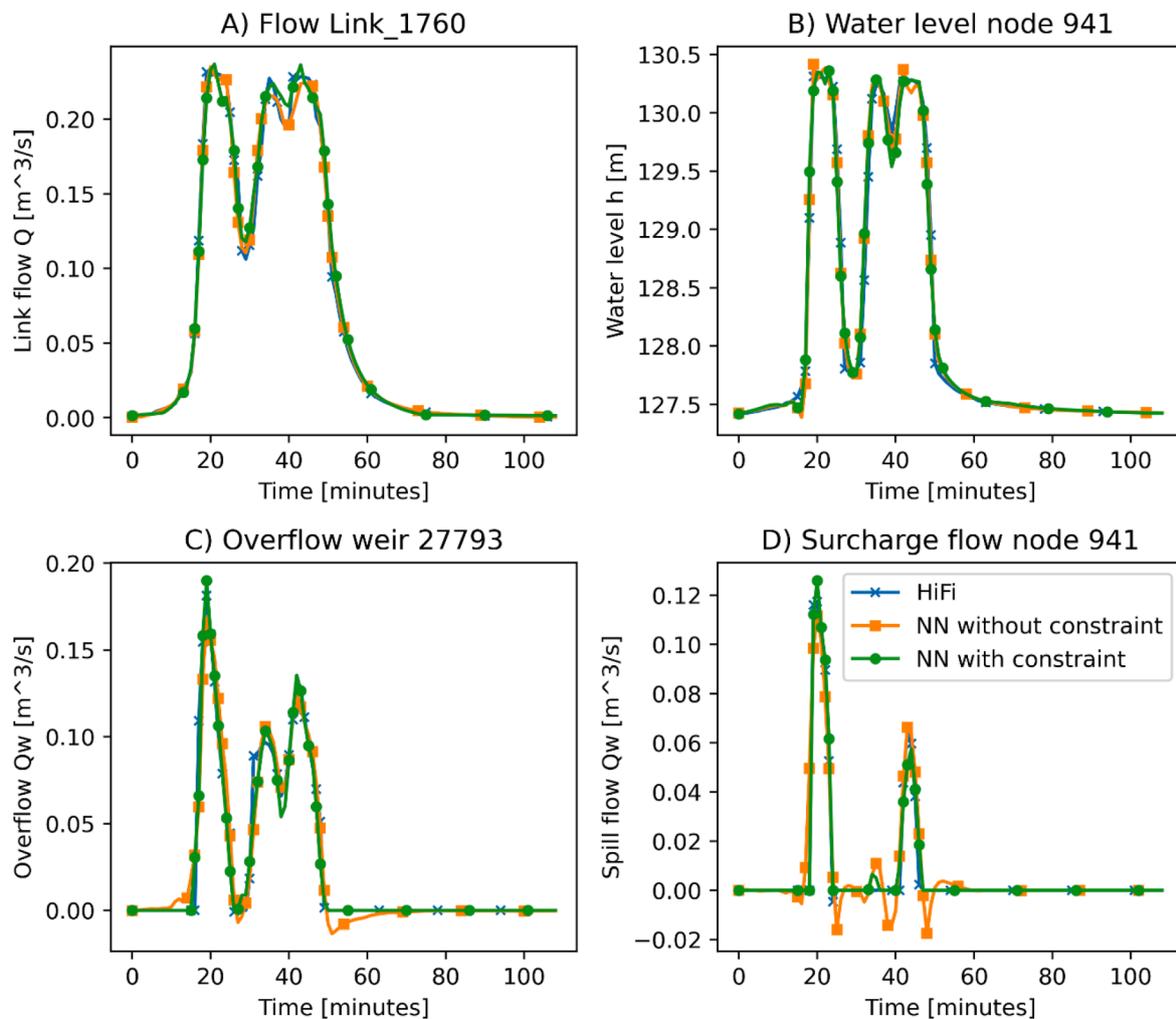


Fig. 8. Time series plots for selected elements in system 1 (see Fig. 7) during a rain event in the test dataset. HiFi model simulations in system 1 (blue crosses) are compared against neural network (NN) emulators without (orange squares) and with (green dots) physical constraints. The surrogates were trained using rain series B and network size S4.

training data (e.g. (Allen et al., 2022)) may enable surrogate training with shorter series and reduced training times.

5 Fig. 8 suggests a clear dependency between the complexity of the neural network included in the surrogate model and the accuracy of the simulations. In general, we would expect that an increase of the number of state variables in the model requires that more complex neural networks with more parameters are needed in the surrogate to achieve sufficient accuracy. The form of this relationship is currently unknown.

Further, better convergence properties may be obtained with alternative network architectures that, for example, employ separate neural networks for water levels and flows, or replicate more advanced numerical solution schemes than the Euler scheme used in our work (Wang and Lin, 1998).

6 All simulations in our study considered a uniform distribution of rainfall in space. For our test cases, this assumption is quite realistic. However, surrogate developments for larger drainage systems will need to consider spatial rainfall variations.

7 While pipe flow simulations are performed with high accuracy, we are not currently enforcing mass balance in the model architecture.

8 Surrogate accuracy was compared solely against HiFi simulations, while the HiFi model does not necessarily reflect reality (Pedersen et al., 2022).

4.2. Upscaling to large drainage networks

In its current configuration, our surrogate approach enables assessments of how changes in urban runoff affect sewer overflows, surcharge frequencies and wastewater treatment plant inflow within a few minutes and thus sufficiently fast. However, any modifications of the hydraulic behavior of the pipe system will require retraining of the surrogate models. In workshop settings, it is infeasible to perform retraining of surrogates for large drainage networks with many thousand links. However, if the model is divided into subsystems then only smaller submodels where changes are made will need to be retrained. This approach raises issues in ensuring that multiple submodels interact in a stable and hydraulically appropriate manner. However, it is common practice to identify “hydraulically simple” locations for subdividing urban drainage networks when setting up conceptual models (Kroll et al., 2017) and the process can be automated (Johansen and Skindhøj, 2022). Another option would be to use conceptual models to simulate the main flows in the sewer network, and then to use these simulations as inflow boundary to the hydraulic surrogates presented in our study. This approach would yield guaranteed mass-conservative simulations, while extending conceptual models with hydraulic detail that is not available today. It would also offer a straight-forward setting for implementing controlled actuators as part of the conceptual model.

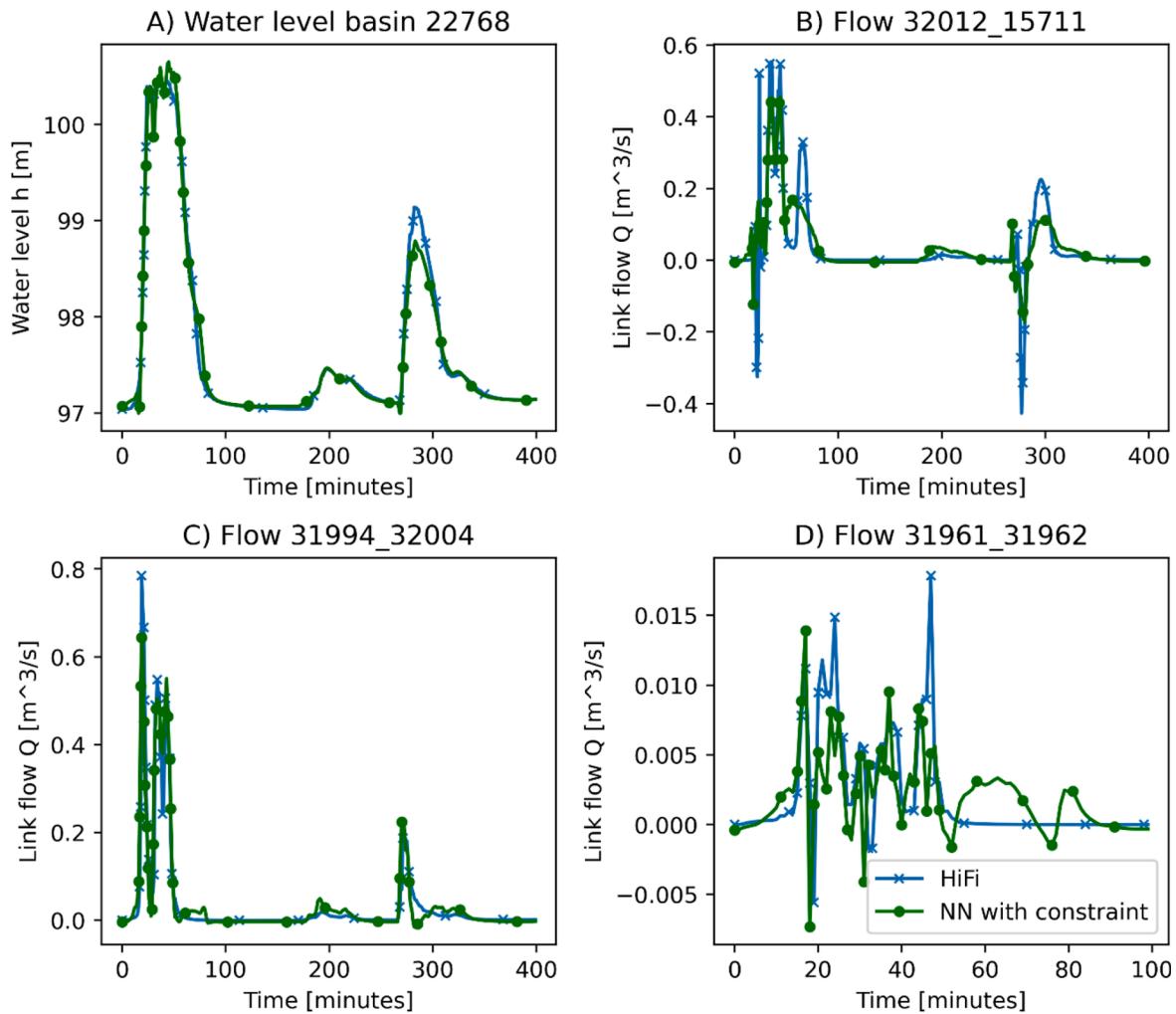


Fig. 9. Time series plots for selected elements in system 2 (see Fig. 7) during a rain event in the test dataset. HiFi model simulations in system 1 (blue crosses) are compared against neural network (NN) emulators with physical constraints (green dots). The surrogates were trained using rain series B and network size S4.

Table 2

Computation time for training surrogate models (includes label generation and actual training process) and simulating the testing series (8,200 rain events; 1.5E6 time steps). Surrogate training and simulation times are averaged across 5 model runs. HiFi simulations considered a fixed time step of 5 s. All time estimates were determined using a single CPU.

| Rain | Model | Label generation [s] | Training [s] | Simulation time [s] |
|-----------------|--------------------|----------------------|--------------|---------------------|
| System 1 | | | | |
| A | Hi-fi | | | 6,200 |
| | Emulator (wo phy.) | 450 | 2,700 | 80 |
| B | Emulator (w phy.) | 450 | 2,020 | 73 |
| | Emulator (wo phy.) | 450 | 4,434 | 80 |
| B | Emulator (w phy.) | 450 | 3,940 | 74 |
| | System 2 | | | |
| B | Hi-fi | | | 11,405 |
| | Emulator (w phy.) | 673 | 5,992 | 114 |

4.3. Research perspectives

We anticipate that future research can much improve our surrogate approach by reducing training times and incorporating additional physical constraints into the model architecture. In terms of reduced training efforts, graph neural networks (Zhou et al., 2020) enable the creation of surrogates where new states for a node or pipe are predicted based on the current state of its neighbors. While the approach presented in our paper considers all states in the drainage network as input to a single neural network, graph approaches would process each node and link individually. When combined with physical system properties such as pipe lengths and slopes, the resulting surrogates may become transferable across catchments. The approach may also improve simulations of flow reversal, because the hydraulics in each node and link would be predicted depending on the current state of the neighboring nodes and links.

Physics-informed loss functions (Raissi et al., 2019; Wang et al., 2021) could enable surrogate training without generating training data in a hydrodynamic model first. While training times would increase, it may be attractive from a practical viewpoint to avoid data management and stability issues associated with automated handling of a numerical engine. In addition, physics-informed loss functions may enable bypassing numerical simplifications that are commonly implemented in commercial software packages for drainage system simulation, because the surrogate can be trained directly against the PDE system. Finally, transfer learning (Jin et al., 2021), i.e., initializing the surrogate

parameters based on experience from previous training iterations is likely to enable surrogate training in much fewer epochs than documented in our paper.

We have far from exploited all opportunities for implementing physical knowledge into the surrogates. For example, flow formulas were used in the development of cellular automata for drainage networks (Austin et al., 2014) and could be incorporated into the L term in Eq. (3) to generate robust flow predictions. Another relevant consideration would be to compute volume changes in each node and link during each simulation time step, and to penalize the surrogates for mass balance deviations during training.

Finally, the model configuration presented in Eq. (3) is generic and can be applied to other water systems. (Wandel et al., 2021) used a similar approach to simulate 3D fluid flows for graphics applications, suggesting that we can derive similar surrogates for, for example, flows in secondary clarifiers in wastewater treatment or 2D surface flows in flood situations.

5. Conclusions

We presented a new surrogate approach for simulating pipe hydraulics, based on physics-guided machine learning using generalized residue networks. We anticipate that this approach will complement existing numerical models in initial design phases for drainage systems, in automated calibration approaches and data assimilation, as well as real-time monitoring and control applications, where fast simulations are required. Based on our results, we draw the following conclusions:

- 1 We can create surrogate models that simulate water levels, flows and surcharges in all nodes and links of a drainage network in a manner that resembles a state-of-the-art numerical engine with sufficient accuracy. A current limitation is to appropriately capture fast flow reversal processes.
- 2 Compared to a numerical engine, simulation times for long rainfall time series of several years are reduced between one and two orders of magnitude. This enables fast and detailed drainage simulations in interactive workshop settings and in real time applications.
- 3 Surrogate training times for a drainage system with ~ 60 links are currently in the order of one hour, considering a single CPU. Transfer learning approaches and graph neural networks will likely enable significant reductions of training efforts in practice.
- 4 Implementing physical constraints in the surrogate in the form of mass balance calculations for surcharges improves performance compared to a purely data-driven approach.
- 5 Surrogate training needs to cover the entire range of hydraulic situations for which the surrogate is intended to be used. Extreme events need to be oversampled in the training data to ensure that surrogates can accurately learn the dynamics of these situations, and a sufficient variation between high and low inflows needs to be applied to all parts of the system.

Generalized residue networks are a generic technique for time-dependent modelling. Their application for dynamic simulations of other types of water systems is therefore an interesting research avenue that warrants investigation.

CRediT authorship contribution statement

Rocco Palmitessa: Conceptualization, Data curation, Investigation, Methodology, Software, Writing – original draft. **Morten Grum:** Conceptualization, Methodology, Writing – review & editing. **Allan Peter Engsig-Karup:** Conceptualization, Methodology, Writing – review & editing. **Roland Löwe:** Conceptualization, Data curation, Investigation, Methodology, Software, Writing – original draft.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was funded by the Danish Environmental Agency (Miljøstyrelsen) under the MUDP programme through the Clacos project (grant number 2020-15748). We thank Ralf Engels, Michaela Ringelkamp and Marko Siekmann from the City of Bochum for providing system data as well as sparring during model development. We thank Peter Steen Mikkelsen for proofreading the manuscript and for his support in initiating the project.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.watres.2022.118972.

References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2016. Tensorflow: a system for large-scale machine learning. In: 12th Symposium on Operating Systems Design and Implementation 2016, pp. 265–283.
- Abbott, M.B., Ionescu, F., 1967. On the numerical computation of nearly horizontal flows. *J. Hydraul. Res.* 5, 97–117. <https://doi.org/10.1080/00221686709500195>.
- Allen, K.M., Mollerup, A.L., Rasmussen, S.F., Danielsen Sorup, H.J., 2022. Efficient job list creation for long-term statistical modelling of combined sewer overflows. *Water Sci. Technol.* 85, 1424–1433. <https://doi.org/10.2166/wst.2022.065>.
- Austin, R.J., Chen, A.S., Savić, D.A., Djordjević, S., 2014. Quick and accurate cellular automata sewer simulator. *J. Hydroinform.* 16, 1359–1374. <https://doi.org/10.2166/hydro.2014.070>.
- Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., Najm, H., Parashar, M., Patra, A., Sethian, J., Wild, S., Willcox, K., Lee, S., 2019. Workshop report on basic research needs for scientific machine learning: core technologies for artificial intelligence. United States. <https://doi.org/10.2172/1478744>.
- Balla, K.M., Bendtsen, J.D., Schou, C., Kallese, C.S., Ocampo-Martinez, C., 2022. A learning-based approach towards the data-driven predictive control of combined wastewater networks – an experimental study. *Water Res.* 221, 118782 <https://doi.org/10.1016/j.watres.2022.118782>.
- Bentivoglio, R., Isufi, E., Jonkman, S.N., Taormina, R., 2022. Deep learning methods for flood mapping: a review of existing applications and future research directions. *Hydrol. Earth Syst. Sci. Discuss.* <https://doi.org/10.5194/hess-2022-83>.
- Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., Gentine, P., 2021. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.* 126, 98302. <https://doi.org/10.1103/PhysRevLett.126.098302>.
- Burger, G., Bach, P.M., Urlich, C., Leonhardt, G., Kleidorfer, M., Rauch, W., 2016. Designing and implementing a multi-core capable integrated urban drainage modelling Toolkit: lessons from CityDrain3. *Adv. Eng. Softw.* 100, 277–289. <https://doi.org/10.1016/j.advengsoft.2016.08.004>.
- Chen, Z., Xiu, D., 2021. On generalized residue network for deep learning of unknown dynamical systems. *J. Comput. Phys.* 438, 110362 <https://doi.org/10.1016/j.jcp.2021.110362>.
- Danandeh Mehr, A., Nourani, V., Kahya, E., Hrnjica, B., Sattar, A.M.A., Yaseen, Z.M., 2018. Genetic programming in water resources engineering: a state-of-the-art review. *J. Hydrol.* 566, 643–667. <https://doi.org/10.1016/j.jhydrol.2018.09.043>.
- DHI, 2021. MIKE 1D - DHI Simulation Engine for 1D river and urban Modelling - Reference Manual. Horsholm, Denmark.
- Frame, J., Kratzert, F., Klotz, D., Gauch, M., Shelev, G., Gilon, O., Qualls, L.M., Gupta, H. V., Nearing, G.S., 2022. Deep learning rainfall-runoff predictions of extreme events. *Hydrol. Earth Syst. Sci.* 26, 3377–3392. <https://doi.org/10.5194/hess-26-3377-2022>.
- Garzón, A., Kapelan, Z., Langeveld, J., Taormina, R., 2022. Machine learning-based surrogate modelling for Urban Water Networks: review and future research

- directions. *Water Resour. Res.*, e2021WR031808 <https://doi.org/10.1029/2021WR031808>.
- Geneva, N., Zabarás, N., 2020. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys.* 403, 109056 <https://doi.org/10.1016/j.jcp.2019.109056>.
- Goodfellow, I., Bengio, Y., Courville, A., 2016. *Deep Learning*. MIT Press.
- Höge, M., Scheidegger, A., Baitty-Jesi, M., Albert, C., Fenicia, F., 2022. Improving hydrologic models for predictions and process understanding using Neural ODEs. *Hydrol. Earth Syst. Sci. Discuss.* <https://doi.org/10.5194/hess-2022-56>.
- Hunt, K.J., Sbarbaro, D., Zbikowski, R., Gawthrop, P.J., 1992. Neural networks for control systems. *Automatica* 28, 1083–1112.
- Jamali, B., Haghghat, E., Ignjatovic, A., Leitão, J.P., Deletic, A., 2021. Machine learning for accelerating 2D flood models: Potential and challenges. *Hydrol. Process.* 35, 1–14. <https://doi.org/10.1002/hyp.14064>.
- Jamali, B., Löwe, R., Bach, P.M., Ulrich, C., Arnbjerg-Nielsen, K., Deletic, A., 2018. A rapid urban flood inundation and damage assessment model. *J. Hydrol.* 564, 1085–1098. <https://doi.org/10.1016/j.jhydrol.2018.07.064>.
- Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 426, 109951 <https://doi.org/10.1016/j.jcp.2020.109951>.
- Johansen, M., Skindhøj, N., 2022. MSc Thesis. Technical University of Denmark. <https://findit.dtu.dk/en/catalog/62f59a382272a203a46b89e7>.
- Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. *Nat. Rev. Phys.* 3, 422–440. <https://doi.org/10.1038/s42254-021-00314-5>.
- Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. <https://doi.org/10.48550/ARXIV.1412.6980>.
- Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A.K., Hochreiter, S., Nearing, G.S., 2019. Toward improved predictions in ungauged basins: exploiting the power of machine learning. *Water Resour. Res.* 55, 11344–11354. <https://doi.org/10.1029/2019WR026065>.
- Kroll, S., Wambeccq, T., Weemaes, M., Van Impe, J., Willems, P., 2017. Semi-automated buildup and calibration of conceptual sewer models. *Environ. Model. Softw.* 93, 344–355. <https://doi.org/10.1016/j.envsoft.2017.02.030>.
- Lagaris, I.E., Likas, A., Fotiadis, D.I., 1998. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.* 9, 987–1000. <https://doi.org/10.1109/72.712178>.
- Leskens, J.G., Brugnach, M., Hoekstra, A.Y., Schuurmans, W., 2014. Why are decisions in flood disaster management so poorly supported by information from flood models? *Environ. Model. Softw.* 53, 53–61. <https://doi.org/10.1016/j.envsoft.2013.11.003>.
- Löwe, R., Böhm, J., Jensen, D.G., Leandro, J., Rasmussen, S.H., 2021. U-FLOOD – topographic deep learning for predicting urban pluvial flood water depth. *J. Hydrol.* 603, 126898 <https://doi.org/10.1016/j.jhydrol.2021.126898>.
- Löwe, R., Mair, M., Pedersen, A.N., Kleidorfer, M., Rauch, W., Arnbjerg-Nielsen, K., 2020. Impacts of urban development on urban water management – limits of predictability. *Comput. Environ. Urban Syst.* 84, 101546 <https://doi.org/10.1016/j.compenurbysys.2020.101546>.
- Lu, L., Jin, P., Pang, G., Zhang, Z., Karniadakis, G.E., 2021. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* 3 <https://doi.org/10.1038/s42256-021-00302-5>.
- Moreno-Rodenas, A.M., Bellos, V., Langeveld, J.G., Clemens, F.H.L.R., 2018. A dynamic emulator for physically based flow simulators under varying rainfall and parametric conditions. *Water Res.* 142, 512–527. <https://doi.org/10.1016/j.watres.2018.06.011>.
- Pedersen, A.N., Brink-Kjær, A., Mikkelsen, P.S., 2022. All models are wrong, but are they useful? Assessing reliability across multiple sites to build trust in urban drainage modelling. *EGUsphere* 2022, 1–32. <https://doi.org/10.5194/egusphere-2022-615>.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: machine learning in python - user guide [WWW Document]. *J. Mach. Learn. Res. URL*. <https://scikit-learn.org/stable/modules/preprocessing.html>. accessed 4.20.22.
- Qin, T., Wu, K., Xiu, D., 2019. Data driven governing equations approximation using deep neural networks. *J. Comput. Phys.* 395, 620–635. <https://doi.org/10.1016/j.jcp.2019.06.042>.
- Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707. <https://doi.org/10.1016/j.jcp.2018.10.045>.
- Razavi, S., Tolson, B.A., Burn, D.H., 2012. Review of surrogate modeling in water resources. *Water Resour. Res.* 48 <https://doi.org/10.1029/2011WR011527>.
- Ren, P., Rao, C., Liu, Y., Wang, J., Sun, H., 2022. PhyCRNet: physics-informed convolutional-recurrent network for solving spatiotemporal PDEs. *Comput. Methods Appl. Mech. Eng.* 389, 114399 <https://doi.org/10.1016/j.cma.2021.114399>.
- Schmid, P.J., 2010. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* 656, 5–28. <https://doi.org/10.1017/S0022112010001217>.
- Thrysoe, C., Arnbjerg-Nielsen, K., Borup, M., 2019. Identifying fit-for-purpose lumped surrogate models for large urban drainage systems using GLUE. *J. Hydrol.* 568, 517–533. <https://doi.org/10.1016/j.jhydrol.2018.11.005>.
- Wandel, N., Weinmann, M., Klein, R., 2021. Teaching the incompressible Navier-Stokes equations to fast neural surrogate models in three dimensions. *Phys. Fluids* 33. <https://doi.org/10.1063/5.0047428>.
- Wang, S., Wang, H., Perdikaris, P., 2021. Learning the solution operator of parametric partial differential equations with physics-informed DeepOnets. *Sci. Adv.* 7, eabi8605. <https://doi.org/10.1126/sciadv.abi8605>.
- Wang, Y.J., Lin, C.T., 1998. Runge Kutta Neural Network for identification of continuous systems. *IEEE Trans. NEURAL NETWORKS* 9, 294–307. <https://doi.org/10.1109/icnnc.1998.726509>.
- Willard, J., Jia, X., Xu, S., Steinbach, M., Kumar, V., 2020. Integrating physics-based modeling with machine learning: a survey. *arXiv* 1, 1–34.
- You, K., Long, M., Wang, J., Jordan, M.I., 2019. How does learning rate decay help modern neural networks?
- Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: a review of methods and applications. *AI Open* 1, 57–81. <https://doi.org/10.1016/j.aiopen.2021.01.001>.