Machine Learning-Based Surrogate Modeling for Urban Water Networks
Review and Future Research Directions

Garzón, A.; Kapelan, Z.; Langeveld, J.; Taormina, R.

DOI
10.1029/2021WR031808

Publication date
2022

Document Version
Final published version

Published in
Water Resources Research

AGU ADVANCING EARTH AND SPACE SCIENCE

Water Resources Research

REVIEW ARTICLE

10.1029/2021WR031808

Key Points:

- Machine learning surrogate models have been widely employed for a variety of applications concerning urban water networks
- New research should investigate machine learning metamodels that account for inductive biases, robustness, and transferability
- Further research should focus on complex problems involving uncertainty and multi-objective optimization, as well as improved benchmarking

Correspondence to:

A. Garzón,
J.A.GarzonDiaz@tudelft.nl

Citation:

Garzón, A., Kapelan, Z., Langeveld, J., & Taormina, R. (2022). Machine learning-based surrogate modeling for Urban Water Networks: Review and future research directions. *Water Resources Research*, *58*, e2021WR031808. https://doi.org/10.1029/2021WR031808

Received 21 DEC 2021
Accepted 5 MAY 2022

Author Contributions:

Conceptualization: A. Garzón, Z. Kapelan, J. Langeveld, R. Taormina
Formal analysis: A. Garzón, Z. Kapelan, J. Langeveld, R. Taormina
Funding acquisition: R. Taormina
Investigation: A. Garzón
Methodology: A. Garzón
Supervision: Z. Kapelan, J. Langeveld, R. Taormina
Visualization: A. Garzón
Writing – original draft: A. Garzón, R. Taormina
Writing – review & editing: A. Garzón, Z. Kapelan, J. Langeveld, R. Taormina

Machine Learning-Based Surrogate Modeling for Urban Water Networks: Review and Future Research Directions

A. Garzón[1], Z. Kapelan[1], J. Langeveld[1,2], and R. Taormina[1]

[1]Faculty of Civil Engineering and Geosciences, Department of Water Management, Delft University of Technology, Delft, The Netherlands, [2]Partners4UrbanWater, Nijmegen, The Netherlands

Abstract Surrogate models replace computationally expensive simulations of physically-based models to obtain accurate results at a fraction of the time. These surrogate models, also known as metamodels, have been employed for analysis, control, and optimization of water distribution and urban drainage systems. With the advent of machine learning (ML), water engineers have increasingly resorted to these data-driven techniques to develop metamodels of urban water networks (UWNs). In this article, we review 31 recent articles on ML-based metamodeling of UWNs to outline the state-of-the-art of the field, identify outstanding gaps, and propose future research directions. For each article, we critically examined the purpose of the metamodel, the metamodel characteristics, and the applied case study. The review shows that current metamodels suffer several drawbacks, including (a) the curse of dimensionality, hindering implementation for large case studies; (b) black-box deterministic nature, limiting explainability and applicability; and (c) rigid architecture, preventing generalization across multiple case studies. We argue that researchers should tackle these issues by resorting to recent advancements in ML concerning inductive biases, robustness, and transferability. Recently developed neural network architectures, which extend deep learning methods to graph data structures, are preferred candidates for advancing surrogate modeling in UWNs. Furthermore, we foresee increasing efforts for complex applications where metamodels may play a fundamental role, such as uncertainty analysis and multi-objective optimization. Lastly, the development and comparison of ML-based metamodels can benefit from the availability of new benchmark datasets for urban drainage systems and realistic complex networks.

Plain Language Summary Analysis and improvement of urban water networks requires hydrodynamic models. Since these models are computationally expensive, researchers and engineers often resort to fast alternatives known as surrogate models. With the rise of artificial intelligence, machine learning methods have been increasingly used for surrogate modeling of urban water networks. In this study, we thoroughly reviewed recent articles in the field to outline the current state-of-the-art and propose future research directions. While many successful applications already exist, we found that these models have three main limiting factors: (a) they need large amounts of data, (b) they are not explainable, and (c) they are too specific to each case. We argue that researchers can overcome these limitations by considering recent advancements in artificial intelligence and implement modeling techniques that better leverage the structure of the underlying data. Other promising directions include developing comprehensive benchmark databases and leveraging surrogate models for more complex applications.

1. Introduction

Urban water networks (UWNs) comprise drinking water distribution and urban drainage systems (WDS and UDS). The former are responsible for supplying drinking water to cities and the latter for evacuating wastewater and stormwater runoff. These infrastructures are a fundamental part of the city and are directly linked to its development (Brown et al., 2009). Each of these systems faces challenges to improve and maintain quality service in a dynamic urban environment under a widening range of climatic conditions; especially, in a climate-changing situation. Designing, optimizing, and intervening in these systems requires approximating their hydraulic behavior. Many models have been developed in the past years for simulating UWNs, for example, SOBEK (Deltares, n.d.), WaterCAD (Bentley®, n.d.), EPA-SWMM (Rossman, 2010), among others. Traditional modeling approaches are either based on accurate description of the physical processes (e.g., EPANET; Rossman et al., 2000) or rely on simplified conceptual models (e.g., SIMPOL; Dempsey et al., 1997); nonetheless, the former usually entail computationally expensive calculations while the latter lack fidelity, that is, level of detail. Applications such as optimization, real-time modeling, and uncertainty analysis need efficient models for evaluating the performance of a system multiple times or as fast as possible. Consequently, they require short execution times while maintaining a sufficient level of detail.

Water modelers have resorted to surrogate models (SMs) to replace computationally costly models. Following the classification given by Razavi et al. (2012b), SMs, also known as metamodels or reduced-order models, can be categorized as lower-fidelity physically-based surrogates (LFPB) or response surface (RS) surrogates. On the one hand, LFPB metamodels modify the original model to reduce its computational effort. These metamodels can simplify the original model by lowering the resolution (e.g., larger time-steps) of the inputs, outputs, or internal processes. The latter is achieved by simplifying the full model by replacing its computationally costly components with faster alternatives (e.g., kriging, linear regression, neural networks; Fernandez et al., 2017). On the other hand, RS surrogates avoid using the original model and replace it altogether with a faster-to-run alternative. In what follows, we summarize the advantages and disadvantages of LFPB and RS metamodels according to Razavi et al. (2012b).

LFPB surrogates, also known as multifidelity based surrogates or "coarse" models, include techniques such as network simplification (Dempsey et al., 1997; Paluszczyszyn et al., 2013; Ulanicki et al., 1996), and skeletonization (Shamir et al., 2008). Compared against RS metamodels, LFPB surrogates are expected to better emulate the unexplored regions of the explanatory variable (input) space (i.e., regions far from the previously evaluated points with the high-fidelity model) and, as such, perform more reliably in extrapolation. As for their drawbacks, LFPB models rely on the assumption that high-fidelity and low-fidelity models share the basic features and are correlated in some way. If this assumption is not satisfied, the surrogate modeling framework would not work, or provide minimal gains. Moreover, mapping the outputs from low resolution to the original resolution is not a trivial task, and may add complexity or uncertainty to the estimations.

Response surface surrogates, also known as statistical and black-box models, include techniques such as polynomials (Schultz et al., 2004), kriging (Baú & Mayer, 2006), and neural networks (Behzadian et al., 2009). In this branch of SMs, the original model is perceived as an input-output function and the metamodel is used to imitate the response surface as best as possible. Some of their advantages include the possibility of maintaining the fidelity of the original model, being model-independent (i.e., not requiring access to the components, such as code or equations of the original model), and easier implementation with respect to LFPB surrogates. Nonetheless, they can be hard to train for high-dimensional problems, which may require substantial computational costs to create large enough databases to train the metamodels. Moreover, RS metamodels require meticulous validation to minimize the chance of over-fitting and maximize their ability to extrapolate.

ML is a subset of artificial intelligence (AI) which is a broad term for tools that imitate cognitive human capabilities. The use of AI has rapidly increased in recent years. The number of peer-reviewed publications across all fields between 2000 and 2019 has grown around 12 times (D. Zhang et al., 2021) and with them, multiple algorithms, architectures, and tools have been created. Fields in which ML methods have shown outstanding results include computer vision, speech recognition, and language processing. Most of these applications use supervised learning, which identifies a branch of ML that is similar to RS metamodeling. Supervised ML employs a set of input-output examples, also known as the labeled training dataset, to calibrate a model by minimizing the error between the model predictions and the values assumed as ground truth. The algorithms in this set usually increase their performance at a given task as the amount of labeled examples grows larger. Due to their successes, supervised ML methods, and in particular deep learning (DL) and artificial neural networks (ANNs), are widely employed for surrogate modeling across many fields of science and engineering (Liu et al., 2021; Peng et al., 2020; Wu et al., 2020). Although scientific studies on ML applications for water resources date back to over two decades ago (Maier & Dandy, 2000), Hadjimichael et al. (2016) noted that this trend is not necessarily witnessed in the urban water sector.

Surrogate models have been used in diverse areas including surface water, groundwater, hydrology, hydraulics, and water resources planning and management (Razavi et al., 2012b). Metamodels are a particularly valuable tool for UWNs. These systems present (a) high density of strongly interconnected components (e.g., pipes, pumps, tanks), (b) nontrivial governing equations, and (c) discrete design and operational variables (e.g., commercial diameters, pump and valve settings). As a consequence, these models face challenges like computationally demanding water quality simulations (Torres-Matallana et al., 2018) and nondeterministic polynomial time-hard (NP-hard) problems for design optimization (Yates et al., 1984). Previous reviews have studied the application of metamodels in water resources. Razavi et al. (2012b) outline taxonomies, practical details, and advances of these SMs in water resources along with recommendations for future research. Among the multiple insights of this work, they highlight the nontrivial effort to choose the right metamodel approach to the problem at hand and advocate for further research on these methods, especially in their assessment and validation. Furthermore, in the same year, Razavi et al. (2012a) numerically assessed metamodeling strategies in computationally intensive optimization, showing that metamodeling is not always a reliable approach, especially for complex response surfaces. The authors also warned about the inappropriateness of neural network models when having a limited computational budget since their effective training conventionally requires a large set of training examples. Later, Broad et al. (2015) presented a formalized qualitative process to determine the most suitable scope for a metamodel based on the evaluation of a fitness function to maximize accuracy. Hadjimichael et al. (2016) reviewed the application of AI methods to UWS management and their integration with decision support systems.

While valuable, these published reviews give low emphasis to SMs for UWNs, and do not account for the recent growth in machine learning-based surrogate models (MLSMs) driven by the rapid advancements in AI. This study aims to fill this gap by assessing the current state of MLSMs for UWNs in order to propose future directions based on identified outstanding issues and recent developments in ML. To achieve this purpose, we applied the review methodology described in Section 2 to review 31 published applications of metamodels for water networks. The results of the review are reported and discussed in Section 3, while major current gaps are detailed in Section 4. We propose future research directions in Section 5 and provide conclusions in Section 6.

2. Materials and Methods

We conducted a semi-systematic (Snyder, 2019) review of MLSM applications for UWNs to synthesize the state-of-the-art of the field. The review integrates the multiple applications of ML metamodels across water network applications, and explores them in a transversal manner. First, we searched journal articles in which MLSMs were applied to UWNs. Second, we determined a set of criteria to assess the relevant characteristics when applying these metamodels to UWNs' problems.

2.1. Search Methodology

We reviewed journal articles published in the last two decades (2001–2021) that use MLSMs for WDSs and UDSs. We established two main search criteria: surrogate modeling and water networks. Since both topics have a multiplicity of names, each of them was represented by a set of keywords. For surrogate modeling, the search terms were: "Surrogate model*", "Metamodel*", "Response surface", "model emulation", and "hybrid model". In the case of water networks, the search terms referred to both water distribution and drainage systems along with popular software for their analysis, "Water distribution", "Water supply", "Drinking water", "Urban drainage", "Wastewater", "Sewer", "Sewerage", "EPANET", "WaterCAD", "SWMM", "SOBEK", and "Urban water".

For the search, we employed the SCOPUS database. By intersecting the search terms, we identified an initial set of 64 articles that were further filtered to only include ML applications, yielding a total of 31 articles to review (17 in WDS, 14 in UDS). Next, we searched through the citations of the selected set of articles and other relevant articles in the field (i.e., Maier & Dandy, 2000; Maier et al., 2014; Razavi et al., 2012b) for further references. However, the original set already contained the cited articles. Therefore, the results are equivalent to the keyword search. This validates the thoroughness of the original search and makes the methodology more replicable by avoiding arbitrarily selected articles.

This list of articles may not be totally inclusive since some studies do not use the formal terminology of surrogate modeling, as indicated by Razavi et al. (2012b). Nevertheless, the purpose of this study is to compile the recent state-of-the-art, identify gaps in knowledge, and propose future research directions. We believe that the selected set of articles is sufficient to achieve this goal.

2.2. Analytical Methodology

In addition to the search criteria, it was necessary to establish an analytical framework that allowed to classify, compare, and evaluate the application of the metamodels across the collected literature. To achieve this, we

Table 1
Categories of Network Size Based on Number of Pipes or Area

Size	Number of pipes in the simulation model	Area [km^2]
Small (S)	<100	<5
Medium (M)	101–250	5–10
Intermediate (I)	251–500	10–20
Large (L)	>500	>20

identified the most relevant aspects of each article in three broad categories: (a) purpose, (b) case study, and (c) metamodel.

Purpose includes general information about the application of the metamodel. It includes the type of network (distribution or drainage) and the application category (e.g., optimization, real-time) as major grouping categories. In addition, it includes the specific application (e.g., optimization of operation, real-time for flood prediction) as a more detailed description for each article.

Case *study* contains information on the physical water network used for the testing and validation of a developed metamodel. This includes the name or location of the case study, whether it is a real case or a benchmark, and its size, indicated by the number of pipes or by the area. The size attribute is also reported as a categorical value ranging from small (S) to large (L), as shown in Table 1.

Metamodel reports details on the computational algorithm (e.g., ANNs, Support Vector Machines) used to replace the original simulator along with further details on its architecture (i.e., deviations from a hidden layer ANN). The type and number of input and output variables are also reported to infer the dimensionality of the SM and the complexity of the RS to approximate. As for the performance, we report the computational speed-up provided by the metamodel and the accuracy to the original simulation, usually approximated with a quantitative metric. These criteria have been identified as the most relevant ones by previous related studies (Broad et al., 2015; Razavi et al., 2012b). Nevertheless, it is possible to consider other factors, such as development time, robustness, and explainability. While assessing these criteria may enrich the analysis, they are not employed in most of the surveyed articles, and they are thus not included in this review.

3. Review—Current Status of Machine Learning Surrogate Models in Urban Water Networks

The analysis of the surveyed articles shows an increase in research activity between 2015 and 2020 with approximately two-thirds of the manuscripts published during this period. In terms of application, most of these articles are related to optimization. For the case study, there is a noticeable difference between WDSs and UDSs since the latter networks lack the use of benchmark cases. Regarding the metamodel, the most popular algorithm is the fully connected ANN; because of this, we report the details of the used metamodel as deviations from a single hidden layer, fully connected ANN, also referred to as simple multi-layer perceptron (MLP). Table 2 summarizes the extracted information of the reviewed articles arranged in the previously mentioned categories: purpose, case study, and metamodel.

3.1. Metamodel Purpose

Figure 1 shows that the two main application categories for metamodels are optimization (48%) and real-time applications (32%), with several examples for both WDSs and UDSs. Metamodels have been also used, although to a lesser extent, for conducting uncertainty analyses, system state estimation (i.e., inferring variables at ungauged points), and to complement LFPB surrogates. The last one refers to the use of an RS method (e.g., linear approximations, polynomials, ANNs) to complement an LFPB metamodel by replacing a slow component or fine-tuning the outputs for better accuracy, for example, surrogating water exchange between sub-catchments with ANNs (Wolfs & Willems, 2017), or correcting the predictions of a hydrodynamic model of wastewater flows (Vojinovic et al., 2003). In all cases, metamodels are used to reduce the computational efforts required for the hydraulic simulation of these complex systems, which may severely compromise the feasibility of the applications.

Optimization usually employs population-based algorithms (e.g., genetic algorithms, particle swarm, ant colony optimization, among others) which require multiple runs. These algorithms create an initial population, and they improve the obtained solutions through continuous iteration. Usually, these algorithms employ mechanisms inspired on genetics, such as crossover and mutation for finding (near) optimal solutions. Evolutionary algorithms are the most well-established metaheuristic for solving water resources problems (Maier et al., 2014); nonetheless, they tend to be highly computationally intensive.

Table 2
List of Reviewed Articles and Metamodeling Approaches

Purpose				Case study				Metamodel			Metamodel performance	
Water network	Application category	Reference	Application	Location	Size: Pipes in model/[area km^2]	Classification by size	Type	Deviations from simple MLP	Inputs (number)	Outputs (number)	Computational saving	Accuracy
Water distribution systems	Optimization	Sayers et al., 2019	Design	TLN, GOY, MOD, BIN	8, 30, 317, 454	S, S, I, I	Benchmark	2 hidden layers	Diameters *	Rating of the network (1)	Not reported	Not reported
		Dini & Tabesh, 2019	Renovation planning	TLN and Ahar, Azerbaijan	8 and 192	S, M	Benchmark and real case		Diameters *	Nodal pressure* and chlorine concentration *	Not reported	Not reported
		Dini & Tabesh, 2017	Model calibration	TLN and Ahar, Azerbaijan	8 and 192	S, M	Benchmark and real case		Observed residual chlorine *	Wall decay coefficient (1)	63 × faster (124 min–118 s)	Average error (3.85%)
		Andrade et al., 2016	Design	HAN and Maricopa, Arizona	34 and 1090	S, L	Benchmark and real case	Comparison of ANNs varying number of inputs and outputs	Diameters and chlorine dosing rates	Chlorine concentration. (HAN): 3; (Maricopa): 9	Not reported	NSE (~90%)
		Bi & Dandy, 2014	Design	(I) NYT, (II) modified NYT and (III) Jilin	21, 21, and 34	S, S, S	(I) Benchmark, (II) modified benchmark, and (III) synthetic network		Diameters and chlorine dosing rates (I & II: 22; III: 35)	Pressures at some nodes (I & II: 4; III: 5) and residual chlorine at one node (I & II: 1; III: 7)	(I & II) 91%; (3.79–0.33 hr) (III) 93%, 88%, and 77%; (11.08 hr–0.8, 1.37, and 2.58 hr)	MSE (not reported, 0.001 as one stopping criteria)
		Broad et al., 2010	Operation	Wallan, Australia	2097; (Sk: 1376)	L (L)	Real case		Trigger levels (45) and chlorine rates (5)	Pressure head at critical node (1), chlorine residual (1), energy value (1), or total chlorine dosed (1)	99.99% (1056 days–1.4 hr)	NSE (~0.6 for the full model, ~0.98 for skeletonized model)
		Behzadian et al., 2009	Sensor placement	(I) Anytown; (II) Mahalat, Iran	41, and 1814; (Sk: 217)	S, L (M)	Benchmark and real case		Available sensors	Sampling design accuracy (1)	(I) 87% (32–4 min); (II) 96% (1550–60 min)	Pareto similarity: 93%

Table 2
Continued

Purpose				Case study				Metamodel			Metamodel performance	
Water network	Application category	Reference	Application	Location	Size: Pipes in model/[area km^2]	Classification by size	Type	Deviations from simple MLP	Inputs (number)	Outputs (number)	Computational saving	Accuracy
		Salomons et al., 2007	Operation	Haifa-A, Israel	126	M	Modified real case		Pumping status (13), valve settings (1), DMA demands (6), storage levels (9)	Power consumption (5), pressures (4), future storage levels (9)	25 × faster	RMSE (0.481%) ~5 cm averaged over all tanks
		Martínez et al., 2007	Operation	Valencia, Spain	772	L	Modified real case		Pumping status (6), valve settings (10), DMA demands (6), storage levels (2)	Power consumption (6), flow rates (3), pressures (4), future storage levels (2)	94 × faster	RMSE (1.30%)
		Broad et al., 2005	Design	NYT	21	S	Benchmark		Diameters and chlorine dosing rate (22)	Four pressure nodes (1) or chlorine concentration (1)	700 × faster (21–0.03 hr)	RMSE (0.05–0.250)
	Real-time	Pasha & Lansey, 2014	Warm solutions for pump scheduling	Modified Anytown	37	S	Modified benchmark	SVM	Pump combination, demand multiplier, initial tank levels	Energy and final tank levels	84.25%* (54–8.5 hr)	NSE (0.99)
		Rao & Alvarruiz, 2007; Rao & Salomons, 2007	Real-time pump scheduling	Modified AnyTown	41	S	Modified benchmark		Number of operating pumps (1), aggregated demand (1), and tank levels (3)	Power consumption (1), pressures (3), new tank levels (3)	10 × faster	RMSE (1.65%)
	Uncertainty analysis	Yoon et al., 2020	Seismic risk assessment	A-city, South Korea	85	S	Anonymous real case	15 layers—deep neural network	Components' state (218)	Network performance (1)	99% (600–1 s)	<5%

Table 2
Continued

Purpose				Case study				Metamodel			Metamodel performance	
Water network	Application category	Reference	Application	Location	Size: Pipes in model/[area km^2]	Classification by size	Type	Deviations from simple MLP	Inputs (number)	Outputs (number)	Computational saving	Accuracy
		Beh et al., 2017	Planning under deep uncertainty	Adelaide, Australia	NA	L	Real case	Combination of 4 MLPs	Supply augmentation options (9) and uncertain variables: population and climate change scenarios (2)	(I) PV of cost	99.98% (33.6 years–50 hr)	Relative error (+-5%)
										(II) PV of Greenhouse gases		
										(III) Reliability		NSE (~0.94, 0.95, 0.78, and 0.84)
										(IV) Vulnerability		
	System state estimation	Lima et al., 2018	Nodal pressure estimation	Campos do Conde II and Cambuí, Brazil	153 and 167	M, M	Real case		Pressure in sensors steady state: (3)—extended (24 hr): 96. Cambuí: (4)	Pressure in nodes steady state: (118)—extended (24 hr): 2832. Cambuí: Steady (154 and 4)	Not reported	Relative error (<1%) and (<4%)
		Meirelles et al., 2017	Calibration with estimated pressures	Campos do Conde II, Brazil and C-Town	153 and 429	M, I	Real case and benchmark		Pressure in sensors steady state: (3)—Extended (24 hr): 96. C-Town: 5 MLPs, one per DMA.	Pressure in nodes steady state: (118)—extended (24 hr): 2832	Not reported	Average error (0.15 m) Max. Error (13.83 m)

Table 2
Continued

Purpose				Case study				Metamodel			Metamodel performance	
Water network	Application category	Reference	Application	Location	Size: pipes in model/ [area km^2]	Classification by size	Type	Deviations from simple MLP	Inputs (number)	Outputs (number)	Computational saving	Accuracy
Urban drainage systems	Optimisation	Seyedashraf et al., 2021	Design	Bogotá, Colombia; Windsor, Canada	511 and 122	L, M	Stormwater—real cases	Generalized regression—2 hidden layers	SUDS characteristics: area, type, and location (20)	Boundary condition: inflow (1)	94.5% (163–9 ms)	Mean error (<0.015) CC (0.99)
		W. Zhang et al., 2019	Design	Urban catchment in China	182	M	Stormwater*—real case	Ensemble of 100 MLPs	Tank length and width (2)	Flood depth (1) or peak flow (1)	80%–90% (600–39 days)	NSE (between 0.66 and 0.92 depending on the rainfall scenario)
		Raei et al., 2019	Design	Tehran, Iran	[20 km^2]	I	Stormwater*—real case	2 hidden layers	Area sizes of the LID, imperviousness and rainfall (3), TSS/BOD build-up (+1), TSS/BOD wash-off (+1)	The volume of runoff (1) or BOD (1) or TSS (1)	Not reported	NSE (0.99)
		Latifi et al., 2019	Design	Tehran, Iran	[20 km^2]	I	Stormwater*—real case		Rainfall value, 6 build-up coefficients, 6 wash off coefficients, 6 imperviousness coefficients, and 32 values for area and type of LIDs (51)	Runoff volume, BOD, TSS (3)	Not reported	Not reported
		Huang et al., 2015	Design	Zhong-He district, Taiwan	[20.29 km^2]	L	Stormwater*—real case		Catchment precipitation, full pipe percentage of water flow in 3 points, the quantity and capacity of rain barrels in four regions (12)	Water level/ flooding at t + 1 (1)	Not reported	MAE (<15%), CC (>0.94–0.97)
	Real-time	Kim & Han, 2020	Flood prediction	Seoul, Korea	[3.19 km^2 *]	M	Stormwater*—real case	8 hidden layers	Total rainfall, max. rainfall in 1–3 hr, rainfall intensity, statistics (SD, skewness, kurtosis), inter-event time (9)	Total accumulative overflow (1)	99.6%* (14 min to 2–3 s)	Mean relative errors between 2% and 62%
		Keum et al., 2020	Flood prediction	Seoul, South Korea	[7.4 km^2]	M	Stormwater*—real case	ANFIS	Rainfall(t-1), volume (t-1), building coverage ratio	Volume (t)	99%* (101–1.2 min)	NSE (0.959)*
		Kim et al., 2019	Flood prediction	Gangnam area, Korea	[7.4 km^2]	M	Stormwater*—real case	SVNARX and SOFM	Accumulative rainfall		98.52% (88–1.3 min)	NSE (0.6–0.94)

Table 2
Continued

Purpose				Case study				Metamodel			Metamodel performance	
Water network	Application category	Reference	Application	Location	Size: pipes in model/ [area km^2]	Classification by size	Type	Deviations from simple MLP	Inputs (number)	Outputs (number)	Computational saving	Accuracy
		She & You, 2019	Outflow prediction	Tianjin, China	33/[0.1314 km^2]	S	Real case with synthetic data	Radial Basis function and NARX	Rainfall intensities (6)	Drainage outfall (1)	Not reported	SSE (0.273)
		Berkhahn et al., 2019	Flood prediction	Anonymous	1224 and 299	L, I	Stormwater*—modifications of real cases	1–4 hidden layers	Precipitation intensities every 5 min (24 for a 2 hr rain event)	The maximum water level at different water cells	NA	RMSE (<0.35 cm)
		Chiang et al., 2010	Flood prediction	Yu-Cheng, Taiwan	[16.45 km^2]	I	Stormwater*—real case	RNN with 1 hidden layer, 3 neurons	Registered water level and precipitation at time t (4)	Water level at time t + n (1)	NA	NSE (>0.97), CC (>0.93), NRMSE (<0.26)
	LFPB complement	Bermúdez et al., 2018	Surface flood volume estimation	Ghent, Belgium	6025/[27.50 km^2]	L	85% combined—real case	Ensemble of ANNs	Rainfall-runoff volumes aggregated over 10 and 30 min windows and volume in the underground system of the closest storage cell (3)	Presence of flooding (1) and magnitude (1)	$10^4 \times$ faster* (408 min–1 s)	NSE (~0.9) but variable
		Wolfs & Willems, 2017	Sewer water quantity simulation	Ghent, Belgium	6025/[27.50 km^2]	L	85% combined—real case		Volumes between two sub-catchments (2)	Flow (1)	$10^6 \times$ faster (30 min–2 ms)	NSE (0.95 in average)
		Vojinovic et al., 2003	Wet weather flow prediction	Catchment in Auckland, New Zealand	[1.07 km^2]	S	Combined and separated—real case	Radial basis function	Error, rainfall, model output (1 - 3)	Error estimation of flow (1)	NA	Improvements of 15%–26%

Note. * denotes information not explicitly mentioned in the article; "Sk" denotes a skeletonized network. BOD, biochemical oxygen demand; CC, correlation coefficient; I, intermediate; L, large; LID, low impact development; M, medium; MAE, mean absolute error; MLP, multi-layer perceptron; MSE, mean squared error; NSE, Nash-Sutcliff efficiency; RMSE, root mean squared error; RNN, recurrent neural network; S, small; SSE, squared sum of error; TSS, total suspended solids.

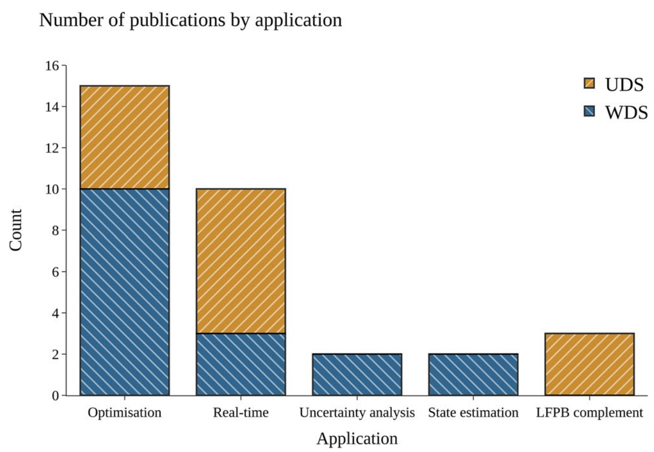

Figure 1. Types of applications that use machine learning metamodels for water distribution systems (WDS) and urban drainage systems (UDS).

Optimization can be used to formulate and solve multiple UWN problems. This explains the high number of metamodeling publications dedicated to this topic. A popular use of MLSMs for optimization in UWNs is for the (re) design of the networks. For example, applications that use MLSMs include changes in pipe diameters and chlorine dosing rates (Andrade et al., 2016; Bi & Dandy, 2014; Broad et al., 2005; Sayers et al., 2019) or operation of storage tanks and pumps (Broad et al., 2010; Martínez et al., 2007; Salomons et al., 2007). The goal for design is to select which new system components to install or identify existing ones to substitute. For operation, the aim is to find an optimal policy on how to operate the existing components. In particular, water quality parameters are considered as outputs for optimization problems in around half of the articles reviewed. In WDS, 6 out of the 10 optimization included chlorine related parameters. For UDS, the relevant parameters were biochemical oxygen demand and total suspended solids which were considered in two (Latifi et al., 2019; Raei et al., 2019) out of five articles. Note that modeling water quality parameters requires higher computational costs compared to hydraulic simulations alone (Broad et al., 2005). This further justifies the use of metamodeling techniques (e.g., Andrade et al., 2016; Bi & Dandy, 2014). Regardless of the task, the goal is to maximize the performance of the system described by the objective function(s) and a number of constraints (e.g., physical, regulatory, economic, among others). In addition, other problems such as water quality model calibration (Dini & Tabesh, 2017), renovation planning (Dini & Tabesh, 2019), and sensor placement (Behzadian et al., 2009) have resorted to metamodels.

Although MLSMs accelerate optimization algorithms, they present a series of drawbacks. First of all, these models need training data (simulation examples) to calibrate their internal parameters (e.g., the weights and biases in a neural network) to replicate the RS. Generating a sufficiently large training dataset can be a time-consuming process, and data sufficiency depends on the complexity of the input-output mapping and it cannot be known a priori. Secondly, the training process is another optimization process in itself, with its own hyperparameters (e.g., learning rate, number of training epochs, parameter initialization, among others depending on the optimizer) and its convergence to a desired performance is not guaranteed. Furthermore, errors of approximation in the RS can mislead the optimization to suboptimal or unfeasible solutions as noted by Broad et al. (2005b), especially in regions near the boundaries or outside the training range.

When comparing water distribution with drainage systems, it is clear that the applications of optimization in UDSs are less diverse. The reviewed articles focus on the optimization of stormwater sewers' design with low impact development management (Latifi et al., 2019; Raei et al., 2019; Seyedashraf et al., 2021) or for flood mitigation (Huang et al., 2015; W. Zhang et al., 2019). Meanwhile, WDS optimization is more varied, with applications to operation, calibration, sensor placement, and long-term planning. This difference partially depends on the stochastic nature of the rainfall events driving the functioning of combined and stormwater sewers, which in turn favor real-time control over the optimization of the operations, typical of WDS. Also, the research conducted on MLSMs for optimization in UDSs is rather recent (2015 or later) compared to WDS (from 2005). Applications in UDSs that typically do not use metamodels can benefit from the experience of tackling similar problems in the context of WDSs. Examples include sensor placement (Sambito et al., 2020), calibration (Tscheikner-Gratl et al., 2016), and optimization of operation (Van Bijnen et al., 2017).

In contrast to off-line optimization, real-time applications require accurate outputs within a short period of time. Real-time operation uses the current state of the system to modify its behavior and improve its functioning in future time steps. In the case of UDSs, they are usually designed to retain stormwater for a certain period, to avoid combined sewer and stormwater outflows (Rosin et al., 2021; She & You, 2019) or to reduce flooding (Berkhahn et al., 2019; Chiang et al., 2010; Keum et al., 2020; Kim et al., 2019; Kim & Han, 2020). Whereas, in WDSs, the objective is to deliver high-quality drinking water while minimizing pumping costs (Pasha & Lansey, 2014; Rao & Alvarruiz, 2007; Rao & Salomons, 2007).

In the case of WDSs, the reviewed real-time applications involve solving optimization problems in a short period of time. Examples of this are the optimal operation of the WDS for the next few hours (Rao & Salomons, 2007) and warming up solutions for pump scheduling for the next day (Pasha & Lansey, 2014). In these cases, the use of the MLSMs is essential to reduce the computational time required to perform the optimization. Consequently, these applications suffer from the already mentioned drawbacks for optimization with MLSMs. For UDS, real-time applications concern Real-Time Control (RTC), where operation and validation relies on real data (Beeneken et al., 2013; Langeveld et al., 2013; Lund et al., 2018). This is an issue since the usual targets are infrequent events, that is, outflows and flooding; therefore, the availability of records may be scarce or nonexistent.

The third application in order of frequency is uncertainty analysis of the UWNs' performance. These analyses are usually carried out via multiple simulations to test the response of the system to multiple possible scenarios or uncertain input conditions, leveraging the computational efficiency of SMs. In WDSs, ANNs have been used to replace computationally expensive models for accelerating Monte Carlo analyses. For example, Yoon et al. (2020) performed a seismic risk assessment of a water distribution network considering earthquakes of different magnitudes and epicentres. In UDSs, Beh et al. (2017) used metamodels to directly estimate reliability and vulnerability metrics. In this case, resorting to MLSMs was crucial for the feasibility of the study. Otherwise, the explicit robustness assessment would have been impossible in practice. Creating a metamodel for uncertainty analysis entails having a model with explicit robustness as output, or generating a training dataset with multiple runs per example. However, the former is rarely the case and the latter consumes a large quantity of computational budget.

Other works tested the ability of ANNs to estimate the system state at ungauged locations with measurements from a limited number of sensors. Lima et al. (2018) and Meirelles et al. (2017) simulated pressure measured at strategically located sensors and an ANN to estimate the pressure of all the remaining nodes in a WDS. SM for state estimation both decreases the degrees of freedom for the addressed calibration problem, and according to the authors, these metamodels could also be used to detect anomalies and predict the current state of the network in real-time. Nevertheless, in these studies, the pressure at all nodes is known since the MLSM is trained on simulations. For applications depending on sensor data, pressure at only few nodes would be known and hence it would not be possible to estimate the error for the ungauged nodes. One alternative to handle this issue is to use data from some sensors for training and other sensors for testing. This way, it is possible to estimate the error at unobserved nodes. However, this process reduces the available training data, and it is not clear how representative the testing sensors are with respect to the remaining ungauged nodes.

Metamodels for UDSs have also been used to complement LFPB surrogates, either to approximate some parts of the model (e.g., the most time-consuming) or to correct the predictions produced by a model. Wolfs and Willems (2017) created a modular approach in which they replaced the hydraulic simulation of drainage flow between subcatchments with an ANN, this was part of a larger framework in which the goal was to simulate outgoing discharges for a given rainfall event. Similarly, Bermúdez et al. (2018) employed an ensemble of ANNs to accelerate a component of an LFPB model, used to estimate the occurrence and magnitude of flooding. In a different manner, Vojinovic et al. (2003) used MOUSE (MOdel for Urban Sewers), a hydrodynamic process model, to estimate flows within wastewater pipes during wet weather periods and trained a neural network to compensate for the output errors (residuals), leading to an overall increase in accuracy. Even though this hybrid approach bridges both metamodeling practices, the LPFB metamodel inherits the RS problems, for example, database creation and training difficulties.

In summary, SMs in water networks have been primarily used for optimization and real-time applications due to their ability to quickly evaluate outputs while remaining sufficiently accurate. Nevertheless, the use of these metamodels is not bound to these two applications. SMs can replace computationally expensive hydrodynamic models for uncertainty analyses and state estimation, or help the original models by correcting outputs or approximating time-consuming components.

3.2. Case Studies

Figure 2 shows the number of case studies analyzed in the reviewed literature. In WDSs, each article usually presents two or more networks. Since the articles introduce new problem formulations or methodologies, the authors apply them to different networks to prove its effectiveness across different WDSs. Studies in optimization usually follow a common pattern where preliminary trials are performed on small benchmark networks before proceeding with implementation in larger real case scenarios. This pattern is repeated in all the cases, whether it is

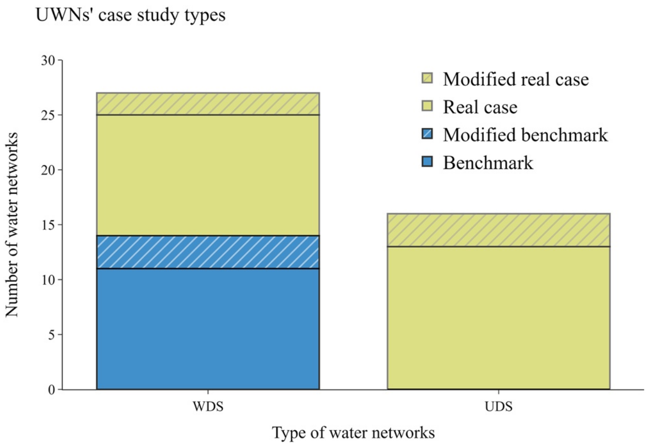

Figure 2. Case study type distribution for water distribution systems (WDS) and urban drainage systems (UDS).

in the same article or in sequential articles, as in the case of the POWADIMA project by Martínez et al. (2007), Rao and Alvarruiz (2007); and Salomons et al. (2007). In the cases of real-time applications, the networks were usually modified benchmarks of medium size. For applications in uncertainty analysis and state estimation, the networks were real cases of large size. The reviewed articles for UDSs, in contrast to WDS, present only applications with real networks, some of them with modifications (e.g., Berkhahn et al., 2019; She & You, 2019).

Regarding UDSs, most of the articles do not report the size of the system in number of pipes. Consequently, the extent of the system was often assessed by the reported area. This suggests that when MLSMs are used, the water network is set aside and only the relation input-output is considered. The extent of the case study (number of pipes or area) is a proxy of the complexity of the case studies which is the relevant dimension. Nevertheless, some applications can involve medium-sized networks but with high complexity (e.g., different control elements, multiple objectives, changing scenarios, among others). Besides the particular characteristics of each network and application, the metamodeling process was the same regardless of the size of the network. However, the required time for creating the database and training the model increases with the complexity of the case study. So far, the procedure does not vary as a function of the complexity of the case study; nonetheless, considering modifications to the training process or the metamodels based on the complexity of the case study could yield better approximations to the RSs.

Since each system has a different area and number of pipes, we proposed the categorization in Table 1. The ratio between the number of small networks and the rest is noticeably bigger in WDSs than in UDSs due to the use of benchmarks to test the methodologies. Even though the use of metamodels is justified in larger networks, its use decreases as the size increases.

3.3. Metamodeling Methods

Regardless of the water network type and metamodel applications, the preferred method for metamodeling is the ANN. ANNs are computational models based on the complex interaction of multiple individual components (i.e., units or neurons). Each unit performs the same procedure: receiving information, executing an operation (usually a linear transformation of the inputs), applying a nonlinear transformation to the result (e.g., hyperbolic tangent, sigmoid, rectified linear unit), and sending the information to the next connected units. Each of the units has trainable parameters that determine the relative weight of each of the inputs. Units are arranged in layers; each ANN has at least one input layer and one output layer, where the inputs are presented to the network and the computed outputs are collected, respectively. Between these layers, there are one or more hidden layers, where most of the information processing takes place. ANNs learn to approximate the input-output relationships in the data by tuning the trainable parameters (i.e., unit's weights and biases) during the backpropagation learning process, which is usually carried via the gradient descent and by computing the partial derivatives of the hidden layers using the chain rule of derivation. For a complete review of ANNs, the reader is redirected to Goodfellow et al. (2016) and Shen (2018) for a specific review related to water resources.

In all cases, training requires a database of examples. The paradigm of ML indicates that performance of the algorithms improves with the size of the database on which they are trained. For supervised ML, this database is generally divided to create a separate dataset for validation. According to the reviewed articles, typical sizes of training datasets vary between 1,000 and 100,000 examples. The size of the training database depends on multiple factors, for example, the complexity of the response surface or the available computational budget to create the examples and train the model. For instance, Andrade et al. (2016) used 10,000 examples for one case and 100,000 for a larger one in the same article. Other approaches implement dynamic training. For example, in some optimization applications, the training set is updated in an online fashion after each iteration (see Behzadian et al., 2009; Bi & Dandy, 2014).

The analysis of the literature shows that the MLP is the most widely used MLSM. The MLP is a specific ANN architecture that consists of a series of layers in which all the units of a layer are connected to all the neurons in the previous and next layer; hence it is also known as the fully connected ANN. Most of the reviewed studies in this article used this architecture with one hidden layer due to its simplicity, high speed, and accuracy. Still, the ANNs can be customized to increase the accuracy of certain applications. This practice of creating deep networks, that is, with more layers and units per layer, is part of modern deep learning (Goodfellow et al., 2016).

In WDSs, there are two cases of variations on the number of layers: Sayers et al. (2019) used two hidden layers for optimization of design while Yoon et al. (2020) used 15 layers in their ANN to estimate the network performance after earthquake events. Deep networks may increase performance but they are more prone to overfitting, and require more training time and examples. Also, it is not possible to know the number of layers and units that yield the best performance. For example, Modesto De Souza et al. (2021) tested multiple architectures of an MLP for pressure estimation in a WDS. Their results suggest that the optimal number of layers is two but this can vary for other applications. On the other hand, UDSs present more variation on the implemented MLPs including varying the number of hidden layers (Berkhahn et al., 2019; Kim & Han, 2020; Raei et al., 2019), changing the activation function to a radial basis function (She & You, 2019; Vojinovic et al., 2003), or adding fuzzy logic (Keum et al., 2020).

As previously stated, MLPs are the most popular MLSMs. This is not surprising because of their ease of implementation and success in multiple applications, as well as publicity from the AI community. However, the MLP, and in general, the ML methods present several drawbacks. As Razavi et al. (2012a) indicated in their numerical assessment of metamodeling strategies in computationally intensive optimization, "the likelihood that a metamodel-enabled optimizer outperforms an optimizer without metamodeling is higher when a very limited computational budget is available; however, this is not the case when the metamodel is a neural network. In other words, neural networks are severely handicapped in limited computational budgets, as their effective training typically requires a relatively large set of design sites, and thus are not recommended for use in these situations" (Razavi et al., 2012a). Therefore, the use of an ANN may even harm the development of an application. In that same work, the authors show that there are cases for which using the original model outperforms using a metamodel. Consequently, they recommend further research on determining where it is worth pursuing a metamodeling approach. In recent years, the widespread availability of parallel computing (e.g., cloud computing and graphics processing unit) and user-friendly DL libraries, such as Pytorch (Paszke et al., 2019), have reduced this problem.

Apart from MLPs, researchers have applied other algorithms from the set of ML tools. For example, Pasha and Lansey (2014) used support vector machines (SVMs) for improving the real-time estimation of water tank levels and thus decreasing pump energy consumption in a WDS. In UDSs, Chiang et al. (2010) implemented an early form of recurrent neural network (RNN) for water level predictions at gauged and ungauged sites. Their decision of using this architecture was motivated by its increase in performance. However, the main disadvantages of this architecture lie in training difficulty (Pascanu et al., 2013) and computational costs (Strubell et al., 2020).

Similarly, Kim et al. (2019) and She and You (2019) leveraged the time structure in rainfall time series for real-time flood prediction with a nonlinear autoregressive network with exogenous inputs (NARX) neural networks. This architecture is a feedforward ANN that calculates the next value of a time series as a function of both past input and output values. In each study, the authors tailored the model to the conditions of their problem. Kim et al. (2019) added a second verification step to account for values that incur serious inundation damage and She and You (2019) implemented a NARX neural network for the monotonic parts of a hydrograph (i.e., ascending and descending stages) and a radial basis function MLP for the non-monotonic interval (i.e., around the peak).

3.3.1. Metamodel Inputs and Outputs

The inputs to the metamodels in UWN applications are usually decision and explanatory variables while the outputs can vary based on the scope of the problem. Based on the inputs used in the reviewed articles, there is not a single consistent variable across the different applications in any of the water networks; they are problem-specific. For example, flood prediction in UDSs relies on rainfall time series, while the design of WDSs relies on inputs such as pipe diameters and chlorine rating doses. On the other hand, the output of the metamodels are usually state variables of the UWN or performance metrics. For example, a metamodel can be developed to estimate a pressure-dependent metric, such as the resilience Network Resilience Index (NRI; Prasad & Park, 2004), or it can output the pressures in a WDS, used to compute the NRI. Other examples of surrogated components are water level in storage units or pump energy consumption. Other examples of overall metrics are sampling accuracy (Behzadian et al., 2009), the economic cost of interventions, greenhouse gases, reliability, and vulnerability (Beh et al., 2017).

Determining the output and scope of the metamodel entails deciding if the metamodel should emulate the model or one of the objectives computed after the hydraulic simulation. The reader is referred to Broad et al. (2015) for a complete methodology about metamodel scope for risk-based optimization and its application to WDS design. In contrast, there are no applications for objective approximation using MLSMs in UDS.

A converging trend is visible when inspecting the dimensions (i.e., number) of the inputs and outputs: the number of inputs is higher than the number of outputs. This is no surprise since most of the studies estimate one or two target values that summarize the desired state of the network (e.g., overall performance, minimum chlorine concentration, total flooding volume) with multiple decision and state variables. Nevertheless, some authors have used fewer variables to produce more outputs. For example, in WDSs, Lima et al. (2018) and Meirelles et al. (2017) estimated 118 pressure nodes with only known pressure at three nodes, while Kim et al. (2019) predicted urban floods in multiple nodes with a single rainfall time series.

More inputs or outputs allow accounting for more complexity in the applications; nonetheless, increasing any of the two presents downsides. For the input dimensions, Razavi et al. (2012b) argue against using a large number of explanatory variables (>20) since the minimum number of training examples can be excessively large. On the other side of the model, the number of output variables also is recommended to be low. In theory, the number of output variables is not restricted; moreover, it is one advantage of ANNs over other RS metamodels as they can act as multi-output emulators. However, an ANN with multiple outputs will seek to find a compromise between the errors of all the outputs, which might be detrimental for the overall accuracy of the MLSM. For this reason, an alternative approach is to train an ANN for each output variable. Since each objective has a metamodel, the accuracy increases but also does the training time. As noted by Andrade et al. (2016), considering one multi-output ANN or multiple ANNs with single output depends on the problem at hand. The size of the water network is the most important factor since, for small systems, the results with one or multiple ANNs are equivalent in performance. In addition, the choice of one model or the other should consider desired accuracy, available metamodeling time, and required speed of execution.

3.3.2. Metamodel Performance

Regarding the performance of a metamodel, multiple criteria can be considered; nevertheless, the most important ones are the computational speed and prediction accuracy. The computational saving is reported as a reduction of the time that the application would have taken by running the original model. This quantity was reported by nearly half of the reviewed studies and it was on average higher than 90%, most of the time over 98%. This is a satisfactory indication since the purpose of these SMs is to reduce the computational burden of intensive applications. Furthermore, we added the computational times of the original model and the metamodel when they were reported in the reviewed articles. Although these depend on multiple variables (e.g., number and quality of the processors, amount of RAM, input/output resolution), reporting computational times in absolute terms facilitates the comparison for individual applications. In most cases, the new running times were orders of magnitude shorter than those of the original models, justifying the use of metamodels. Nonetheless, around half of the studies did not report this saving. Although quantifying the computational saving usually entails additional effort, it is recommended for future researchers who use a metamodel to consider such an estimate. Since the design and training time could be longer than the expected saved time, having an estimate of the potential saving aids in the decision of making a metamodel.

In terms of prediction accuracy, researchers use multiple indicators to assess the degree of exactness of the ML algorithm to approximate the original model. These common metrics include root mean squared error (RMSE), Nash-Sutcliffe efficiency coefficient (NSE), mean absolute error, and Pearson correlation coefficient. This multitude of metrics hinders a straight comparison between models or applications, but overall it is possible to observe highly accurate fittings between the metamodel and the original model. It is worth noticing that the metamodel will reflect reality as much as the original model is capable of doing so. Metamodels are second-level abstractions and therefore may only be as good as the original model in terms of accuracy.

In addition to the previously mentioned criteria, Razavi et al. (2012b) include development time, and Asher et al. (2015) add surrogate-introduced uncertainty as assessment metrics. For these criteria, seven of the reviewed articles calculated or referred to the time it took to train the models and only five performed an analysis on the metamodels' robustness. Given the versatility and multipurpose nature of the SMs, there are other performance indicators, for example, ease of development, explainability, generalization, or re-trainability. Along these lines, the reviewed articles disregard these indicators since the development of the metamodel is specific for each case study and the implementation goes unnoticed. These indicators are secondary in comparison to computational saving and accuracy. Both metrics constitute the most relevant metrics used in the literature, including this review.

4. Current Issues in Metamodeling

Based on the current status presented in the previous section the following issues were identified.

4.1. Basic Applications

MLSMs have been used to tackle various issues, namely, optimization, uncertainty analyses, real-time applications, state forecasting, and aiding LFPB metamodels. Although these generally addressed relevant problems, each of the reviewed articles had a simple framing, that is, the inputs are few design or input variables (e.g., diameters, chlorine dosage, accumulated rainfall) and the outputs are usually summary values (e.g., critical pressure, chlorine residual, flood volume). This approach is comprehensible for several reasons. First, most of the time the simplifications made still retain sufficient problem information to find an adequate solution. Second, this approach avoids problems related to high dimensionality in the inputs and outputs. Lastly, it allows researchers to easily explain and communicate their approach.

Although this approach is effective, it could end up being simplistic for the complexity of water networks. Considering a small set of interventions may discard types and combinations of interventions (e.g., allowing not only for change in diameters but also adding pumps or doing both at the same time). Furthermore, other changes in the network or their components, or even interactions with other city systems could be explored. However, these are rarely considered since they represent a challenge for traditional RS metamodels; current MLSMs are very specific to the cases in which they are trained on. Because of this, new approaches are required, mainly in optimization and uncertainty analysis.

As can be seen in Section 3, the most popular application for MLSMs is optimization. In this application, multiple authors (Beh et al., 2017; Doorn, 2021; Kapelan et al., 2005; Razavi et al., 2021) have remarked on the importance of considering new objective functions. For some of these, it is preferable to surrogate the objective function directly rather than surrogating the system variables used to compute the objective function value. For example, robustness for designing water systems, especially under deep uncertainty, requires considering multiple scenarios for which is not possible to assign a probability or ranking. This analysis is desirable because water networks are systems with long lifespans of service. Nonetheless, objectives like robustness (Beh et al., 2017) and sensitivity (Razavi et al., 2021) tend to be more computationally intensive; therefore, their need for metamodels increases.

A relevant missing layer of complexity is uncertainty analysis, especially for UDSs. The current practice to design the system is to use a single benchmark storm and assume it is representative of the future rain events the system will face. However, two UDSs with similar performance during a design event could behave very differently for other rainfall patterns. According to Ng et al. (2020), the final design considering a single strong storm does not guarantee optimal performance during long mild storms and for a succession of frequent small events. Naturally, the authors recognize that performing a design considering multiple events would increase the computational effort but also suggest the implementation of SMs for dealing with this difficulty.

4.2. Case Studies: Lack of Benchmarking With Complex Networks

Benchmark water networks are effectively open access datasets that contain the information necessary to create a simulation model of a system (network topology, system components with related characteristics, water consumption, rainfall, etc.) and other related information about the analyzed benchmark problem (e.g., information about cyber-attack events analyzed or surveillance data). Here, it is necessary to distinguish between synthetic and real data. Even though the synthetic data allow to implement and compare algorithms, they may not reflect all the processes that real data can account for. Benchmarks are used as reference points to compare the performance of models and algorithms. Without benchmark networks it is more difficult to compare the performances of different metamodeling approaches. Consequently, it is hard to establish what techniques work better than others and how to improve these.

There is a clear difference between types of infrastructure in the number of used networks since benchmark networks in UDSs are not as available as in WDSs. In water distribution, there is a set of water networks called Water Distribution System Research database. The ASCE Task Committee on Research Databases for WDS created this database which is hosted by the University of Kentucky (2013). There are benchmarks for multiple problems in categories such as network expansion, operation, and design. This allows modelers to easily obtain data for the development and comparison of algorithms in networks of different sizes. On the other hand, there is no consolidated set of benchmark networks for UDSs, let alone an entire structured database. This is attributable to factors such as the difficulty of taking measurements in sewer environments and, according to Pedersen et al. (2021), the little interest of utility companies in making the datasets publicly available. Consequently, all the applications on UDSs were entirely developed for real cases, which is positive for the bridging between the theoretical approaches and the practice, but hampers the development of algorithms on the systems, due to the difficulty of comparison and the process of accounting for particularities of each system.

Regarding the size of the case studies, most of the systems in which the MLSMs were used were medium or small. Metamodels are most useful in problems with large computational times, that is, in applications with large water networks. In the case of WDSs, a common practice to test the effectiveness of a method is developing a metamodel for a small benchmark network and then using the same steps for creating a metamodel in a large real case. Even though this practice is reasonable, it assumes the response surface of both networks is comparable or similar. However, this is not necessarily the case as reported by Andrade et al. (2016) who noted contrasting accuracies between large and small case studies when training metamodels. Exploring solution spaces is already an issue when using metamodels, independent of the network, as reported by Broad et al. (2005), but analyzing large networks represents additional challenges that increase in complexity in a nonlinear manner.

4.3. Machine Learning and MLP Limitations

Although the MLP is not the only ML technique, it is the most popular one among MLSMs. Given that its structure allows it to address multiple types of problems, it has become a one-size-fits-all model. Nevertheless, it presents multiple issues, namely, the curse of dimensionality, black-box nature, and rigid structure. These three shortcomings respectively (a) hinder their use for high dimensionality problems, (b) limit confidence in their approximations, and (c) prevent the transferability of trained models across different case studies.

4.3.1. Curse of Dimensionality—Metamodeling Time

The curse of dimensionality indicates that for a certain level of accuracy, there is an exponential increase in the required amount of data as the dimensions of a problem increase (Keogh & Mueen, 2017). Naturally, this problem can be addressed by reducing the number of input dimensions (i.e., fewer explanatory variables) using prioritization based on experience, knowledge of the task, or some automatic procedure such as principal component analysis (PCA). However, reducing the number of input variables may not be a satisfactory solution for real-world problems as it could exclude regions of the search space (Maier et al., 2014). Given this situation, improved metamodeling could help approximate complex response surfaces while retaining all the regions of the solution space.

The SMs have worked adequately so far but future metamodels are likely to increase in complexity. This is either due to an increase in the complexity of UWNs or an increase in the number of input (more design choices/explanatory variables) or output (more objectives) dimensions. Both drivers increase the size of the metamodels and consequently the number of training examples. Since the original models are already expensive to run, creating a large training dataset might be unfeasible in the first place. The metamodeling time would become the obstacle. This time is usually disregarded since some authors consider it not relevant compared to the posterior computational gain in the application. Nevertheless, this time is important in high dimensional search spaces, as noted by Razavi et al. (2012b), since the number of design samples required to train the metamodel could be already prohibitively large.

4.3.2. Black Box Nature—Deterministic and Obscure Outputs

Two of the most recurrent criticisms of ML models are their lack of uncertainty estimation and the lack of their transparency, that is, little or no ability to explain the results they obtain. Both are overlooked aspects of metamodeling in the context of UWNs. The MLSMs return a unique answer without uncertainty bands or possibilities to explain the combination of inputs that drove to the final outputs. For SMs, these issues are not major concerns; nevertheless, their inclusion aids the applications in which the SMs are used.

Regarding uncertainty estimation, a few articles (Raei et al., 2019; Rosin et al., 2021; She & You, 2019; W. Zhang et al., 2019) estimated the effect of including a metamodel in their respective application. Not accounting for this uncertainty can lead to bad approximations of the actual response surface and suboptimal or unfeasible solutions. Authors have dealt with this difficulty by performing sensitivity analysis (e.g., Raei et al., 2019) or training multiple models in parallel with slightly different datasets and averaging the outputs of the models. For example, Rosin et al. (2021) developed a committee of ANNs with this approach. However, this analysis requires extra considerations which may increase the metamodeling time. Some guidelines have been given for the pretreatment (Broad et al., 2015) and posttreatment (Broad et al., 2005) of these SMs but there is still a lack of focus on improving the management of uncertainty during treatment, that is, developing a model that directly considers uncertainty. Algorithms in the branch of robust ML may contribute to aid in the direct incorporation of metamodel uncertainty quantification whether it comes from the data (Wong & Kolter, 2019) or the model (Loquercio et al., 2020).

Although robust learning allows estimating the uncertainty of a result, it cannot explain why. This is the area of explainable ML. For water networks' SMs, being able to explain the results would help to understand the relationship between the decision variables and the objective function for the particular network that is being surrogated. For example, understanding which pipes (or a combination of them) play a key role in the resilience or flooding in a water network. There is a growing interest in the AI community toward explainable models to gain insights (Bhatt et al., 2020), ensure scientific value (Roscher et al., 2020), and develop trust in the outcomes of ML models (Dosilovic et al., 2018).

4.3.3. Rigid Architecture—Specific Case Use

One disadvantage of MLSMs is the high degree of specialization in the trained metamodel. As seen before, these metamodels achieve high accuracies in the data for which they were trained. However, once they are trained, they become specific and rigid. Their structure limits its use for other tasks in the same system or similar applications in other water networks. The metamodel can be run several times on the same water network but doing the same operation in a different system requires a new metamodel, which should be trained from scratch. This is not desirable since the training process could consume most of the computational budget, especially in large case studies.

One solution is to leverage the training process of other models with transfer learning to decrease the number of examples to train a new model. Situations for which transfer learning is desirable are changes in the water network composition, similar system metamodeling, and change in the behavior of the surrogated system. Changing components of the system accounts for scenarios when components (e.g., pipes, pumps, or tanks) are added to or removed from the system. Even though the system changes, it is still related enough to leverage a pretrained model on that water network. In a similar way, two networks can share enough resemblance (e.g., a subsystem of another network, two skeletonized networks, or two networks with similar topology or geography) that it makes sense to use an SM from one as a pretrained SM for the other. Likewise, the UWN can change so that the current metamodel no longer applies; this challenge is also known as concept drift.

4.4. Gaps in Knowledge

Based on the above critical analyses of metamodels and the issues identified, the key gaps in knowledge are summarized here:

1. Lack of depth on optimization of complex objectives and uncertainty analysis for water networks using MLSMs. There are still additional and more complex objectives that can be optimized with the aid of MLSMs, for instance, robustness and interventions under deep uncertainty.
2. Lack of benchmark water networks, especially for UDSs and complex cases. First, this hinders the development and comparison of algorithms across studies, and second, these metamodels still lack research on the

10.1029/2021WR031808

changes of the response surface with the increase in the complexity of the water system, especially for large systems.

3. Current MLSMs' limitations prevent advanced metamodeling applications. MLSMs can easily grow in size when the complexity of the response surface increases, most of the applications do not consider the uncertainty added by the metamodel, and its structure makes it rigid and not (re)useable for other cases.

5. Research Directions

Based on the identified gaps, three main lines for future research are suggested. They consider the current and future needs in applications on UWNs as well as the potential of MLSMs to meet them.

5.1. Advanced Applications

The current needs for adaptable water infrastructure are based on drivers such as growing demographics, urbanization, and climate change. As indicated in the UN-Water report "Water and Climate Change", taking adaptation and mitigation measures benefits water resources management and improves the provision of water supply and sanitation services. In addition, it contributes to combat both causes and impacts of climate change while contributing to meeting several of the Sustainable Development Goals (UNESCO, 2020). In UWNs, multi-objective optimization and uncertainty analysis play a key role in the search for adaptation measures and decision making, and MLSMs can help improve and accelerate their implementation.

Optimization applications will increase in the number and complexity of the inputs and outputs. Increasing the number of inputs, that is, decision variables and design interventions (e.g., nature-based solutions), allows to explore more alternatives, consider uncertainty, or assess multiple scenarios. On the other hand, the output of the optimization is leaning toward complex objectives such as multi-objective robustness (e.g., Kasprzyk et al., 2013), multiple technical performance metrics (e.g., Fu et al., 2013), pro-active maintenance (Kumar et al., 2018), complex water quality indicators (Jia et al., 2021), and human values (Doorn, 2021). Multi-objective optimization allows identifying solutions balancing trade-offs among objectives, for instance, cost and resilience (Wang et al., 2015). Naturally, when considering more objectives, the computational load increases, especially when those objectives are computationally expensive (e.g., robustness). In previous phases of research on optimization, metamodels were seen as an aid, but as optimization gradually evolves to consider additional and more complex objectives, metamodels become indispensable (e.g., Beh et al., 2017).

Regarding uncertainty analysis, it is necessary to have fast, reliable, and flexible metamodels that can adapt to the multiple conditions in which the systems are evaluated and under multiple criteria. Traditionally, simplified models have been preferred for this task; however, RS metamodels become appealing alternatives when dealing with more complex objective functions and original models. Metamodels should play a key role in the development of frameworks for robustness-driven design. This application has major implications for UDSs, since no MLSM study focused on uncertainty analysis, even when the evidence suggests the criteria for the design of these systems is not necessarily robust (Ng et al., 2020). Although uncertainty analysis entails an intrinsic increase in the computational effort, the benefits they bring outweigh the challenges it represents. According to the IPCC (2021b), UDSs are expected to receive more intense rainfall events based on climatic projections but considerable uncertainty remains.

The community should further research combined RS-LPFB applications, to further integrate MLSMs with physically-based models for accelerating the underlying hydrodynamic engines. Likewise, physically-based models could be hybridized by incorporating an ML model that corrects the outputs of the original model for higher accuracy accounting for the real behavior of the system. Looking ahead, ML algorithms could detach from the physically-based model and replace its functioning with a cheaper version to run based on increasingly available real-world data (e.g., digital twins for UWNs (IWA, 2021)).

5.2. Benchmarking and Large Network Behavior

Maier et al. (2014) had already identified the lack of benchmark models and determined the characteristics and recommendations of valuable benchmarks, including nontrivial real-world problems with a representative range of decision problems characteristic of the water systems. The review shows that UDSs lack such benchmarks. To overcome this issue, we recommend to implement a similar approach to that of the Kentucky database, with applications such as real-time control, outflow, and flood prediction. For WDSs, it is appropriate to enlarge the current databases to account for new objectives, interventions, performance metrics, and real case examples. Regarding metamodels, the benchmarks should also include a reference model to compare computational saving and accuracy, with suggested performance metrics, such as NSE, RMSE, or the number of model executions. Ideally, the benchmark could include an already running metamodel in practice, together with relevant information provided by their developers and users.

As Goodfellow et al. (2016) indicate, having benchmark databases with real cases is one of the reasons why DL has recently become a crucial technology in several disciplines. In AI, datasets went from hundreds or thousands of examples in the early 1980s up to datasets with millions of examples after 2010. Due to the current increase in connectivity and digitalization of our society, a large amount of ML algorithms can be fed with the information they require to achieve high accuracy. Since the ML and DL models are dependent on their training sets, their success goes hand in hand with the size and quality of available datasets, preferable with real information. The UWNs' research community is moving the first steps in this direction. One example concerns the UDS of the Bellinge dataset (Pedersen et al., 2021), a suburb to the city of Odense, Denmark that is now available for "independent testing and replication of results from future scientific developments and innovation within urban hydrology and urban drainage system research". This dataset includes 10 years of asset data (information from manholes and links), sensor data (level, flow, and power meters), rain data, hydrodynamic models (MIKE urban and EPA SWMM), and other information. Similar examples are needed to enable the exploration of metamodels' responses in networks of different characteristics (e.g., size, connectivity, slope).

As for the size of the networks, further research is required to assess the response surface of large networks. Specifically, new benchmark datasets should also include complex network cases for their study. These can be large networks or medium-size cases with high complexity. Considering that the larger the network the higher the required time to generate and use the training data, significant efforts are required on this matter. Metamodels could aid in reducing the computational times that obstruct studying the response surface of large and complex systems. Nonetheless, new metamodels are required to account for the complexity of these cases and use as few training scenarios as possible.

5.3. Unexplored Advanced Metamodeling Technologies

ML is one of the fastest growing fields. However, the field of MLSMs for UWNs has not yet considered the new tools and algorithms recently developed by researchers in fundamental AI or other applied disciplines. These advancements include DL architectures that express assumptions of the data in the ANNs for robust, interpretable, and transferrable models. This new wave of AI formalizes the attempts to add knowledge about modeled processes as well as extract knowledge from the results.

5.3.1. Inductive Bias—Deep Learning

The curse of dimensionality can be addressed by including inductive biases. Following the work of Battaglia et al. (2018), we define the inductive bias as the "expression of assumptions about either the data-generating process or the space of solutions". Inductive bias can be seen as well in the architecture of the model by leveraging the inner structure of the data, which could be spatial, temporal, or relational. Exploiting the structural information of the data can reduce the number of parameters, and consequently the required training examples by parameter sharing and sparsity of connections. In this way, it is possible to counteract the tendency of ML models to grow in size and data requirements. The data structure gives information about the similarity of the data points in a relevant dimension (e.g., distance, time, connection). In that sense, similar data can be treated analogously (parameter sharing) and dissimilar data can remain unrelated (sparse connectivity).

Inductive bias nudges a learning algorithm to prioritize some solutions over others. Ideally, involving inductive bias improves the search for solutions without compromising the performance, as long as the right inductive bias is chosen; otherwise, it can lead to suboptimal performance (Battaglia et al., 2018). For example, when surrogating the pressure at the nodes of a WDS with a neural network (e.g., Broad et al., 2005; Meirelles et al., 2017) there are multiple metamodel solutions, that is, architectures with specific parameter values that can approximate the response surface described by the training data. Nevertheless, when adding inductive bias, the set of possible solutions shrinks to a subset of solutions that comply with predefined characteristics, for example, having graph structure, following physical laws, or agreeing with measurements.

Deep learning is a subset of ML algorithms, mainly neural networks, that leverage inductive biases. The most common inductive bias in these algorithms is hierarchical processing (Battaglia et al., 2018). That is, the composition of elementary components (i.e., layers) to discover more complex interactions in the input data. The typical elementary components in DL are fully connected, convolutional, recurrent, and, more recently, graph layers. The fully connected layers have a weak inductive bias, while each of the remaining exploits some relation or invariance in the data. The convolutional layers typical of convolutional neural networks (CNNs) leverage the regular structures in grids, such as images, and connect information according to Euclidean closeness. RNNs consist of recurrent layers which consecutively process data sequences, such as time series, and connect information according to sequential similarity. Graph neural networks (GNNs) extend DL methods to non-Euclidean data, such as graphs, where entities are connected by relations or, in graph terminology, nodes connected by edges.

It is possible to use combinations of layers in problems that contain more than one structure such as in the case of UWNs, which have temporal, spatial, and topological variability. An example of the application of these in a civil infrastructure was developed by Sun et al. (2020) who included the topological and temporal relations in a road network for traffic forecasting. This infrastructure has multiple parallels with UWNs, including its graph connectivity, spatial-temporal variability, and human interaction.

Given their relational inductive bias, the newly developed GNN appears as a suitable DL architecture for applications in UWNs, since the natural structure of these systems can be represented using a graph. Researchers have already exploited graph theoretical concepts of UWNs (Deuerlein, 2008; Herrera et al., 2016; Meijer et al., 2018, 2020). Furthermore, some applications of GNNs in UWNs already exist. In WDSs, Tsiami and Makropoulos (2021) employed this architecture for cyber-physical attack detection using a graph created from sensors in the water system. Xing and Sela (2022) used the GNN to create a model for state estimation based on the layout of the WDS. Although showcased for simulated case studies, the ML models in these articles have been developed for usage in real scenarios. Therefore, these applications are not considered as surrogate models. In UDSs, Belghaddar et al. (2021) applied GNNs to complete missing values in databases of wastewater networks. Given its novelty and potential, further research on the GNN architecture is recommended to establish the benefits and limitations of this approach for surrogating UWN models, together with comparisons against already established MLSMs, for example, fully connected neural networks.

5.3.2. Third Wave of AI

The US Defense Advanced Research Projects Agency (DARPA, 2016) separates the different phases of AI into three waves. The first wave refers to the past approaches and the birth of AI, the second wave is the current and popular phase of high-performing black boxes, and lastly, the third wave is proposed for the future of AI with models leaning toward robustness and explainability.

Robustness refers to the ability to include uncertainty in the calculation of the outputs of a model, in this way the user receives both a deterministic answer and a range of possible values, usually represented by an expected value (e.g., mean) and a measure of uncertainty (e.g., variance). According to Gawlikowski et al. (2021), methods for estimating uncertainty in ANNs can be split into four types: single deterministic methods, Bayesian methods, ensemble methods, and test-time augmentation methods. Each of these lines offers an estimation of the degree to which the neural network is certain of the output. This aspect is relevant when quantifying how likely it is for the metamodel to detach from the response surface which may cause, depending on the application, to omit optimal solutions, miss outflows, or underestimate floods. Recommended methods for implementation on MLSMs include Bayesian neural networks (e.g., Zhu & Zabaras, 2018) or single deterministic methods, the latter is recommended based on the low additional computational burden they include.

Research in explainability has also gained popularity in recent years. In the case of MLSMs, an explainable model would improve understanding of the response surface of the original model or the solution space. An improved comprehension of the response surface would facilitate obtaining a better insight on the behavior of different algorithms (e.g., evolutionary methods); ultimately, contributing to what type of heuristic is best suitable in each application in water network which is a topic in which we have still very little understanding of (Maier et al., 2014). On the other hand, solution space explanation would allow gaining insight about which components in the real system affect its performance, but most importantly, how they affect it. This could drive the interventions in the physical water network to improve its performance. For example, Tsiami and Makropoulos (2021) were able to perform a removal analysis to quantify the contribution of each considered component (e.g., valves, tanks, and pumps) of the physical water network to the model's performance using GNNs. Since the GNN structure resembles the underlying system, it is possible to relate events on the metamodel to the actual system.

Although GNNs' outputs remain hardly explainable, there are efforts to generate explanations of them, for example, GNNExplainer (Ying et al., 2019). As noted by Battaglia et al. (2018), "the entities and relations that GNNs operate over often correspond to things that humans understand (such as physical objects), thus supporting more interpretable analysis and visualization". In this way, GNNs are not entirely explainable but they are more explainable than other DL architectures.

5.3.3. Transferrable AI Models

The reviewed studies in this article presented a methodology for training a metamodel to surrogate a computationally expensive model. Although the methodology is transferrable, meaning the steps can be followed and repeated to obtain a similar metamodel in another case study, the metamodel itself cannot be transferred to a new case study. This implies that all the metamodeling time spent on training is specific for every case. Transferrable models would allow training the metamodel with data not only from the case study at hand but also from other, real and synthetic cases. For example, from the previously discussed benchmark datasets. This increase in available information to train on is expected to improve the performance of the metamodel or even allow it to exist for cases in which data is scarce, for example, very computationally expensive UWNs in which training examples are costly. Through transferrable models, the authors may develop not only methodologies but also pretrained SMs, which can be adapted to other cases lowering the amount of training needed for a new network.

Once again, inductive bias plays a role, since the assumptions added to the algorithm delimit a smaller solution space, the ML models can be used as pretrained solutions for other tasks. In the AI domain, this practice is referred to as transfer learning. Transfer learning is mainly implemented for specialized DL methods, that is, architectures with strong inductive bias. It has been successfully implemented for applications such as diagnosis of medical images using CNNs (Vogado et al., 2018), prediction of air pollutants using RNNs (Hang et al., 2020), and bioinformatics as well as social-network classification tasks with GNNs (Verma & Zhang, 2019), among others (Weiss et al., 2016).

For transferrable SMs in UWNs, GNNs seem to be a natural option based on the agreement between the structure of the real system and the inductive bias corresponding to the GNNs. In an analogous way that CNNs learn filters that are independent of the input (i.e., images), GNNs could learn filters that can be used across cases (e.g., water networks). Adding the structure and physics to the metamodel allows including more domain knowledge in the ANN which improves generalization capabilities. A relevant example of a model like this is the mass conserving RNN for rainfall-runoff modeling developed by Hoedt et al. (2021) in which the parameters used in the model resemble the mass conservation principle, which increased the accuracy and improved the model's interpretability. At the same time, transferability opens the door to new applications, such as online optimization of interventions, by learning the effect of changes in the topology and components of the network.

Using physical information, such as the knowledge embedded in the hydrodynamic models, also allows generating hybrid and general models. These models allow bridging the best of two domains: physical-based and data-driven. On this: Vojinovic et al. (2003) indicated that "the major advantage of integrating both a deterministic (numerical) model and a stochastic (data-driven) model over using the stochastic data-driven model alone is that the already available deterministic model quality is exploited and improved, instead of starting from scratch and throwing away all knowledge." Furthermore, combining the domain knowledge with transferable models opens the possibility of creating general models. This type of model detaches from the training set in which it was trained so that its predictions can be applied in unseen scenarios. Following this trend, Kratzert et al. (2019) developed a recurrent ANN trained on basins from a continental dataset using meteorological time series data and static catchment attributes, and they were able to outperform hydrological benchmark models calibrated on individual catchments. The analogous application in UWNs would be an ML-based hydrodynamic model trained on a set of distribution or drainage systems which can generalize to independent unknown water networks.

5.4. Summary of Research Lines

Based on the above research directions, the recommendations for future research lines are summarized here:

1. We recommend additional efforts in two areas in which metamodels will be increasingly required for adaptation and mitigation measures on UWNs: multi-objective optimization and uncertainty analysis
2. We recommend consolidating valuable benchmarks that include large, nontrivial, and real-world problems with a representative range of decision problems characteristic of the UWNs along with reference models and performance metrics to compare computational saving and accuracy
3. Given the growth and resource availability in AI, we recommend to explore latest ML learning algorithms that are capable of expressing assumptions, that is, inductive bias, of the systems and data, as well as methods that increase robustness, interpretability, and transferability

6. Conclusions

This work reviews the current state of the application of MLSMs in urban water networks and proposes promising forward directions based on recent and successful developments in ML.

In terms of purpose, the main uses of MLSM in UWNs are optimization and real-time problems. Even though MLSM accelerate optimization algorithms by increasing the speed of individual iterations, these algorithms have multiple disadvantages. The training process can be time-consuming and the required size of that dataset cannot be known a priori as it depends on the complexity of the input-output mapping. For case study type, the UWNs in which MLSMs are applied vary in size and type. For analyzing the complexity of the case studies, we preferred to consider WDSs and UDSs separately. On the one hand, WDSs articles follow a clear pattern: development and implementation. The former use medium or small benchmark networks, and the latter a large real network. On the other hand, UDSs do not count with applications on benchmark networks due to their lack of availability. In terms of the metamodel, except for some applications of SVMs or RNNs, the vast majority of applications used MLP as SM. This method has been successfully implemented due to its high accuracy and flexibility regarding the inputs and outputs that it can map. Nevertheless, the MLSMs present multiple drawbacks that may even harm the development of an application. It is advisable to consider if an MLSM is worthwhile before deciding using one.

Based on the reviewed literature, the following issues and gaps in knowledge were identified in terms of limitations of existing MLSMs. These problems include limitations on the MLSMs, lack of depth in current applications, and insufficient benchmarking datasets.

1. Regarding metamodels' limitations, current MLSMs have the following issues: they can easily grow in size when the complexity of the response surface increases, most of the applications do not consider the uncertainty added by the metamodel, and its structure makes it rigid and not (re)useable for other cases.
2. In terms of applications, optimization is where most of the SMs are currently used; nevertheless, there are still additional and more complex objectives that can be optimized with the aid of MLSMs, for instance, robustness and interventions under deep uncertainty.
3. On case studies, the reviewed articles denote two main issues: first, there is a lack of UDSs benchmarks, which hinders the development and comparison of algorithms across studies, and second, these metamodels still lack research on the changes of the response surface with the increase in the complexity of the water system, especially for large systems.

The following research directions are suggested to address the above key gaps in knowledge:

1. Regarding metamodeling methods, further research is required on advanced metamodeling techniques that include: inductive bias, robustness, and transferability. The notion of inductive bias allows leveraging prior information to reduce the required training samples. Examples of this bias include adding physical laws, coherence with sensor data, or considering the underlying structure of the data—space, time, or topology—In this regard, the recently developed GNNs resemble the already existing architecture of the urban water networks and offer an useful inductive bias to consider in the metamodels. Furthermore, the new approach for AI models is to focus on the robustness and explainability of the models which offer insight into the applications and opportunities for improvement in the actual systems. Moreover, implementing the new architectures

of ML as an SM would allow transfer learning, which represents the ability to use pretrained models and save computational budget

2. On applications, additional efforts are encouraged in two areas in which metamodels will increasingly be more required: uncertainty analysis and multi-objective optimization, especially when robustness metrics are used as optimization objectives. Further research is required on other less developed applications, namely, real-time predictions, state estimation, and to a lesser extent, LFPB complements. These applications have been minimally explored and most of them have only been used for a specific type of water network
3. Regarding case study type, it is crucial to develop benchmark UWNs, especially of UDSs, and complex networks. This data will facilitate training, testing, and comparing new metamodels. These new benchmarks could incorporate information on leakages, demand patterns, cyber-attacks, rainfall, or surveillance data as well as performance metrics as reference points to compare performance

Exploring the potential of MLSMs for approximating UWNs' components and correcting predictions with real data can lead to independent ML models of the water networks that leverage the physical domain knowledge and the measurements. New MLSMs are encouraged to leverage the inductive bias offered by the increasing data to help UDS and WDS operators. The new advancements in ML have great potential to advance surrogate modeling for UWNs and strengthen its application in practice. Water network modelers can speed up calculations for larger and more complex cases, being able to design more robust and overall better urban water systems.

Data Availability Statement

No experimental data or code were produced for this manuscript.

Acknowledgments
This work is supported by the TU Delft AI Labs programme.

References

Andrade, M. A., Choi, C. Y., Lansey, K., & Jung, D. (2016). Enhanced artificial neural networks estimating water quality constraints for the optimal water distribution systems design. *Journal of Water Resources Planning and Management*, *142*(9), 04016024. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000663

Asher, M., Croke, B. F., Jakeman, A., & Peeters, L. J. (2015). A review of surrogate models and their application to groundwater modeling. *Water Resources Research*. https://doi.org/10.1002/2015wr016967

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., et al. (2018). *Relational inductive biases, deep learning, and graph networks*. *ArXiv*, 1–40. Retrieved from https://arxiv.org/abs/1806.01261

Baú, D. A., & Mayer, A. S. (2006). Stochastic management of pump-and-treat strategies using surrogate functions. *Advances in Water Resources*, *29*(12), 1901–1917. https://doi.org/10.1016/J.ADVWATRES.2006.01.008

Beeneken, T., Erbe, V., Messmer, A., Reder, C., Rohlfing, R., Scheer, M., et al. (2013). Real time control (RTC) of urban drainage systems – A discussion of the additional efforts compared to conventionally operated systems. *Urban Water Journal*, *10*(5), 293–299. https://doi.org/10.1080/1573062X.2013.790980

Beh, E. H. Y., Zheng, F., Dandy, G. C., Maier, H. R., & Kapelan, Z. (2017). Robust optimization of water infrastructure planning under deep uncertainty using metamodels. *Environmental Modelling & Software*, *93*, 92–105. https://doi.org/10.1016/j.envsoft.2017.03.013

Behzadian, K., Kapelan, Z., Savic, D., & Ardeshir, A. (2009). Stochastic sampling design using a multi-objective genetic algorithm and adaptive neural networks. *Environmental Modelling & Software*, *24*(4), 530–541. https://doi.org/10.1016/j.envsoft.2008.09.013

Belghaddar, Y., Chahinian, N., Seriai, A., Begdouri, A., Abdou, R., & Delenne, C. (2021). Graph convolutional networks: Application to database completion of wastewater networks. *Water (Switzerland)*, *13*(12), 1–19. https://doi.org/10.3390/w13121681

Bentley®. (n.d.). *WaterCAD*. Retrieved from https://www.bentley.com/en/products/product-line/hydraulics-and-hydrology-software/watercad

Berkhahn, S., Fuchs, L., & Neuweiler, I. (2019). An ensemble neural network model for real-time prediction of urban floods. *Journal of Hydrology*, *575*, 743–754. https://doi.org/10.1016/j.jhydrol.2019.05.066

Bermúdez, M., Ntegeka, V., Wolfs, V., & Willems, P. (2018). Development and comparison of two fast surrogate models for urban Pluvial flood simulations. *Water Resources Management*, *32*(8), 2801–2815. https://doi.org/10.1007/s11269-018-1959-8

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., et al. (2020). *Explainable machine learning in deployment* (pp. 648–657).

Bi, W., & Dandy, G. C. (2014). Optimization of water distribution systems using online retrained metamodels. *Journal of Water Resources Planning and Management*, *140*(11), 04014032. https://doi.org/10.1061/(ASCE)WR.1943-5452.0000419

Broad, D. R., Dandy, G. C., & Maier, H. R. (2005). Water distribution system optimization using metamodels. *Journal of Water Resources Planning and Management*, *131*(3), 172–180. https://doi.org/10.1061/(asce)0733-9496(2005)131:3(172)

Broad, D. R., Dandy, G. C., & Maier, H. R. (2015). A systematic approach to determining metamodel scope for risk-based optimization and its application to water distribution system design. *Environmental Modelling & Software*, *69*, 382–395. https://doi.org/10.1016/j.envsoft.2014.11.015

Broad, D. R., Maier, H. R., & Dandy, G. C. (2010). Optimal operation of complex water distribution systems using metamodels. *Journal of Water Resources Planning and Management*, *136*(4), 433–443. https://doi.org/10.1061/(asce)wr.1943-5452.0000052

Brown, R. R., Keath, N., & Wong, T. H. F. (2009). Urban water management in cities: Historical, current and future regimes. *Water Science and Technology*, *59*(5), 847–855. https://doi.org/10.2166/wst.2009.029

Chiang, Y.-M., Chang, L.-C., Tsai, M.-J., Wang, Y.-F., & Chang, F.-J. (2010). Dynamic neural networks for real-time water level predictions of sewerage systems-covering gauged and ungauged sites. *Hydrology and Earth System Sciences*, *14*(7), 1309–1319. https://doi.org/10.5194/hess-14-1309-2010

DARPA. (2016). *Perspective on AI*. Retrieved from https://www.darpa.mil/about-us/darpa-perspective-on-ai

Deltares. (n.d). *SOBEK*. Retrieved from https://www.deltares.nl/en/software/sobek/

Dempsey, P., Eadon, A., & Morris, G. (1997). Simpol: A simplified urban pollution modelling tool. *Water Science and Technology*, *36*(8–9), 83–88. https://doi.org/10.2166/wst.1997.0648

Deuerlein, J. W. (2008). Decomposition model of a general water supply network graph. *Journal of Hydraulic Engineering*, *134*(6), 822–832. https://doi.org/10.1061/(asce)0733-9429(2008)134:6(822)

Dini, M., & Tabesh, M. (2017). Water distribution network quality model calibration: A case study-Ahar. *Water Science and Technology: Water Supply*, *17*(3), 759–770. https://doi.org/10.2166/ws.2016.166

Dini, M., & Tabesh, M. (2019). Optimal renovation planning of water distribution networks considering hydraulic and quality reliability indices. *Urban Water Journal*, *16*(4), 249–258. https://doi.org/10.1080/1573062X.2019.1669185

Doorn, N. (2021). Artificial intelligence in the water domain: Opportunities for responsible use. *The Science of the Total Environment*, *755*, 142561. https://doi.org/10.1016/j.scitotenv.2020.142561

Dosilovic, F. K., Brcic, M., & Hlupic, N. (2018). Explainable artificial intelligence: A survey. *2018 41st International convention on information and communication technology, Electronics and Microelectronics. MIPRO 2018 - Proceedings, 210–215*. https://doi.org/10.23919/MIPRO.2018.8400040

Fernandez, G., Park, C., Kim, N. H., & Haftka, R. (2017). *Review of multi-fidelity models*. *ArXiv, March*. Retrieved from https://arxiv.org/abs/1609.07196

Fu, G., Kapelan, Z., Kasprzyk, J. R., & Reed, P. (2013). Optimal design of water distribution systems using many-objective visual analytics. *Journal of Water Resources Planning and Management*, *139*(6), 624–633. https://doi.org/10.1061/(asce)wr.1943-5452.0000311

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., et al. (2021). *A survey of uncertainty in deep neural networks*, (pp. 1–41). Retrieved from http://arxiv.org/abs/2107.03342

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press. Retrieved from http://www.deeplearningbook.org

Hadjimichael, A., Comas, J., & Corominas, L. (2016). Do machine learning methods used in data mining enhance the potential of decision support systems? A review for the urban water sector. *AI Communications*, *29*(6), 747–756. https://doi.org/10.3233/AIC-160714

Hang, I., Li, T., Fong, S., & Wong, R. K. (2020). Knowledge-based systems predicting concentration levels of air pollutants by transfer learning and recurrent neural network. *Knowledge-Based Systems*, *192*, 105622. https://doi.org/10.1016/j.knosys.2020.105622

Herrera, M., Abraham, E., & Stoianov, I. (2016). A graph-theoretic framework for assessing the resilience of sectorised water distribution networks. *Water Resources Management*, *30*(5), 1685–1699. https://doi.org/10.1007/s11269-016-1245-6

Hoedt, P.-J., Kratzert, F., Klotz, D., Halmich, C., Holzleitner, M., Nearing, G., et al. (2021). MC-LSTM: Mass-conserving LSTM. Retrieved from http://arxiv.org/abs/2101.05186

Huang, C.-L., Hsu, N.-S., Wei, C.-C., & Luo, W.-J. (2015). Optimal spatial design of capacity and quantity of rainwater harvesting systems for urban flood mitigation. *Water (Switzerland)*, *7*(9), 5173–5202. https://doi.org/10.3390/w7095173

IPCC. (2021). *IPCC: Climate change 2021: The physical science basis*. Cambridge University Press. Retrieved from https://www.ipcc.ch/report/ar6/wg1/

IWA. (2021). *Digital water - operational digital twins in the urban water sector: Case studies*. Retrieved from https://iwa-network.org/publications/operational-digital-twins-in-the-urban-water-sector-case-studies/

Jia, Y., Zheng, F., Maier, H. R., Ostfeld, A., Creaco, E., Savic, D., et al. (2021). Water quality modeling in sewer networks: Review and future research directions. *Water Research*, *202*, 117419. https://doi.org/10.1016/j.watres.2021.117419

Kapelan, Z. S., Savic, D. A., & Walters, G. A. (2005). Multiobjective design of water distribution systems under uncertainty. *Water Resources Research*, *41*(11), 1–15. https://doi.org/10.1029/2004WR003787

Kasprzyk, J. R., Nataraj, S., Reed, P. M., & Lempert, R. J. (2013). Many objective robust decision making for complex environmental systems undergoing change. *Environmental Modelling & Software*, *42*, 55–71. https://doi.org/10.1016/j.envsoft.2012.12.007

Keogh, E., & Mueen, A. (2017). Curse of dimensionality. *Encyclopedia of Machine Learning and Data Mining*, 314–315. https://doi.org/10.1007/978-1-4899-7687-1_192

Keum, H. J., Han, K. Y., & Kim, H. I. (2020). Real-time flood disaster prediction system by applying machine learning technique. *KSCE Journal of Civil Engineering*, *24*(9), 2835–2848. https://doi.org/10.1007/s12205-020-1677-7

Kim, H. I., & Han, K. Y. (2020). Urban flood prediction using deep neural network with data augmentation. *Water (Switzerland)*, *12*(3). https://doi.org/10.3390/w12030899

Kim, H. I., Keum, H. J., & Han, K. Y. (2019). Real-time urban inundation prediction combining hydraulic and probabilistic methods. *Water (Switzerland)*, *11*(2). https://doi.org/10.3390/w11020293

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., & Nearing, G. (2019). Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. *Hydrology and Earth System Sciences*, *23*(12), 5089–5110. https://doi.org/10.5194/HESS-23-5089-2019

Kumar, A., Rizvi, S. A. A., Brooks, B., Vanderveld, A., Wilson, K. H., Kenney, C., et al. (2018). *Using machine learning to assess the risk of and prevent water main breaks* (Vol. 2657, pp. 1–9). https://doi.org/10.1145/3219819.3219835

Langeveld, J. G., Benedetti, L., Klein, J. J. M. D., Nopens, I., Amerlinck, Y., Nieuwenhuijzen, V., et al. (2013). *Impact-based integrated real-time control for improvement of the Dommel River water quality* (p. 9006). https://doi.org/10.1080/1573062X.2013.820332

Latifi, M., Rakhshandehroo, G., Nikoo, M. R., & Sadegh, M. (2019). A game theoretical low impact development optimization model for urban storm water management. *Journal of Cleaner Production*, *241*. https://doi.org/10.1016/j.jclepro.2019.118323

Lima, G. M., Brentan, B. M., Manzi, D., & Luvizotto, E. (2018). Metamodel for nodal pressure estimation at near real-time in water distribution systems using artificial neural networks. *Journal of Hydroinformatics*, *20*(2), 486–496. https://doi.org/10.2166/hydro.2017.036

Liu, X., Tian, S., Tao, F., & Yu, W. (2021). A review of artificial neural networks in the constitutive modeling of composite materials. *Composites Part B: Engineering*, *224*, 109152. https://doi.org/10.1016/j.compositesb.2021.109152

Loquercio, A., Segu, M., & Scaramuzza, D. (2020). A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, *5*(2), 3153–3160. https://doi.org/10.1109/LRA.2020.2974682

Lund, N. S. V., Falk, A. K. V., Borup, M., Madsen, H., & Mikkelsen, P. S. (2018). Model predictive control of urban drainage systems: A review and perspective towards smart real-time water management. *Critical Reviews in Environmental Science and Technology*, *48*(3), 279–339. https://doi.org/10.1080/10643389.2018.1455484

Maier, H., & Dandy, G. (2000). Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environmental Modelling & Software*, *15*(1), 101–124. https://doi.org/10.1016/S1364-8152(99)00007-9

Maier, H., Kapelan, Z., Kasprzyk, J., Kollat, J., Matott, L. S., Cunha, M. C., et al. (2014). Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environmental Modelling & Software*, *62*, 271–299. https://doi.org/10.1016/j.envsoft.2014.09.013

Martínez, F., Alonso, M., Herna, V., Rao, Z., & Alvisi, S. (2007). *Optimizing the operation of the Valencia water- distribution network*. 65–78. https://doi.org/10.2166/hydro.2006.018

Meijer, D., Bijnen, M. V., Langeveld, J., Korving, H., Post, J., & Clemens, F. (2018). Identifying critical elements in sewer networks using graph-theory. *Water (Switzerland)*, *10*(2). https://doi.org/10.3390/w10020136

Meijer, D., Post, J., van der Hoek, J. P., Korving, H., Langeveld, J., & Clemens, F. (2020). Identifying critical elements in drinking water distribution networks using graph theory. *Structure and Infrastructure Engineering*, *17*(3), 347–360. https://doi.org/10.1080/15732479.2020.1751664

Meirelles, G., Manzi, D., Brentan, B., Goulart, T., & Luvizotto, E. (2017). Calibration model for water distribution network using pressures estimated by artificial neural networks. *Water Resources Management*, *31*(13), 4339–4351. https://doi.org/10.1007/s11269-017-1750-2

Modesto De Souza, R. G., Melo Brentan, B., & Meirelles Lima, G. (2021). Optimal architecture for artificial neural networks as pressure estimator. Brazilian. *Journal of Water Resources*, 1–9. https://doi.org/10.1590/2318-0331.262120210100

Ng, J. Y., Fazlollahi, S., & Galelli, S. (2020). *Do design storms yield robust drainage systems? How rainfall duration, intensity, and profile can affect drainage performance*, *146*(3), 1–13. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001167

Paluszczyszyn, D., Skworcow, P., & Ulanicki, B. (2013). Online simplification of water distribution network models for optimal scheduling. *Journal of Hydroinformatics*, *15*(3), 652–665. https://doi.org/10.2166/HYDRO.2013.029

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *30th International Conference on Machine Learning, ICML 2013, PART 3*, 2347–2355.

Pasha, M. F. K., & Lansey, K. (2014). Strategies to develop warm solutions for real-time pump scheduling for water distribution systems. *Water Resources Management*, *28*(12), 3975–3987. https://doi.org/10.1007/s11269-014-0721-0

Paszke, A., Lerer, A., Killeen, T., Antiga, L., Yang, E., Gross, S., et al. (2019). *PyTorch: An imperative style, high-performance deep learning library*.

Pedersen, A. N., Pedersen, J. W., Vigueras-Rodriguez, A., Brink-Kjær, A., Borup, M., & Mikkelsen, P. S. (2021). The Bellinge data set: Open data and models for community-wide urban drainage systems research. *Earth System Science Data Discussions*, *April*, 1–28. https://doi.org/10.5194/essd-2021-8

Peng, G. C. Y., Alber, M., Buganza, A., William, T., Suvranu, R. C., Dura, D. S., et al. (2020). Multiscale modeling meets machine learning: What can We learn. *Archives of Computational Methods in Engineering*, 0123456789. https://doi.org/10.1007/s11831-020-09405-5

Prasad, T. D., & Park, N.-S. (2004). Multiobjective genetic algorithms for design of water distribution networks. *Journal of Water Resources Planning and Management*, *130*(1), 73–82. https://doi.org/10.1061/(asce)0733-9496(2004)130:1(73)

Raei, E., Reza Alizadeh, M., Reza Nikoo, M., & Adamowski, J. (2019). Multi-objective decision-making for green infrastructure planning (LID-BMPs) in urban storm water management under uncertainty. *Journal of Hydrology*, *579*. https://doi.org/10.1016/j.jhydrol.2019.124091

Rao, Z., & Alvarruiz, F. (2007). Use of an artificial neural network to capture the domain knowledge of a conventional hydraulic simulation model. *Journal of Hydroinformatics*, *9*(1), 15–24. https://doi.org/10.2166/hydro.2006.014

Rao, Z., & Salomons, E. (2007). Development of a real-time, near-optimal control process for water-distribution networks. *Journal of Hydroinformatics*, *9*(1), 25–37. https://doi.org/10.2166/hydro.2006.015

Razavi, S., Jakeman, A., Saltelli, A., Prieur, C., Iooss, B., Borgonovo, E., et al. (2021). The future of sensitivity analysis: An essential discipline for systems modeling and policy support. *Environmental Modelling & Software*, *137*. https://doi.org/10.1016/j.envsoft.2020.104954

Razavi, S., Tolson, B. A., & Burn, D. H. (2012a). Numerical assessment of metamodelling strategies in computationally intensive optimization. *Environmental Modelling & Software*, *34*, 67–86. https://doi.org/10.1016/j.envsoft.2011.09.010

Razavi, S., Tolson, B. A., & Burn, D. H. (2012b). Review of surrogate modeling in water resources. *Water Resources Research*, *48*(7). https://doi.org/10.1029/2011WR011527

Roscher, R., Bohn, B., Duarte, M., & Garcke, J. (2020). Explainable machine learning for scientific insights and discoveries. *IEEE Access*, *8*, 42200–42216. https://doi.org/10.1109/ACCESS.2020.2976199

Rosin, T. R., Romano, M., Keedwell, E., & Kapelan, Z. (2021). A committee evolutionary neural network for the prediction of combined sewer overflows. *Water Resources Management*, *35*(4), 1273–1289. https://doi.org/10.1007/s11269-021-02780-z

Rossman, L. A. (2010). *Storm water management model user's manual, Version 5.0*. Retrieved from https://www.epa.gov/water-research/storm-water-management-model-swmm

Rossman, L. A., Tryby, M., & Janke, R. (2000). *EPANET 2.2 user manual*.

Salomons, E., Goryashko, A., Shamir, U., Rao, Z., & Alvisi, S. (2007). *Optimizing the operation of the Haifa-A water-distribution network* (pp. 51–64). https://doi.org/10.2166/hydro.2006.017

Sambito, M., Di Cristo, C., Freni, G., & Leopardi, A. (2020). Optimal water quality sensor positioning in urban drainage systems for illicit intrusion identification. *Journal of Hydroinformatics*, *22*(1), 46–60. https://doi.org/10.2166/hydro.2019.036

Sayers, W., Savic, D., & Kapelan, Z. (2019). Performance of LEMMO with artificial neural networks for water systems optimisation. *Urban Water Journal*, *16*(1), 21–32. https://doi.org/10.1080/1573062X.2019.1611886

Schultz, M. T., Small, M. J., Farrow, R. S., & Fischbeck, P. S. (2004). State water pollution control policy insights from a reduced-form model. *Journal of Water Resources Planning and Management*, *130*(2), 150–159. https://doi.org/10.1061/(ASCE)0733-9496(2004)130:2(150)

Seyedashraf, O., Bottacin-Busolin, A., & Harou, J. J. (2021). *A disaggregation-emulation approach for optimization of large urban drainage systems water resources research* (Vol. 2017, pp. 1–18). https://doi.org/10.1029/2020WR029098

Shamir, U., Asce, F., & Salomons, E. (2008). Optimal real-time operation of urban water distribution systems using reduced models. *Journal of Water Resources Planning and Management*, *134*(2), 181–185. https://doi.org/10.1061/(ASCE)0733-9496(2008)134:2(181)

She, L., & You, X.-Y. (2019). A dynamic flow forecast model for urban drainage using the coupled artificial neural network. *Water Resources Management*, *33*(9), 3143–3153. https://doi.org/10.1007/s11269-019-02294-9

Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, *54*(11), 8558–8593. https://doi.org/10.1029/2018WR022643

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, *104*(March), 333–339. https://doi.org/10.1016/j.jbusres.2019.07.039

Strubell, E., Ganesh, A., & McCallum, A. (2020). Energy and policy considerations for modern deep learning research. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence, 1*, 1393–13696. https://doi.org/10.1609/aaai.v34i09.7123

Sun, Y., Wang, Y., Fu, K., Wang, Z., Zhang, C., & Ye, J. (2020). Constructing geographic and long-term temporal graph for traffic forecasting. *Proceedings - International Conference on Pattern Recognition*, 3483–3490. https://doi.org/10.1109/ICPR48806.2021.9412506

Torres-Matallana, J. A., Leopold, U., Klepiszewski, K., & Heuvelink, G. B. M. (2018). EmiStatR: A simplified and scalable urban water quality model for simulation of combined sewer overflows. *Water (Switzerland)*, *10*(6). https://doi.org/10.3390/w10060782

Tscheikner-Gratl, F., Zeisl, P., Kinzel, C., Rauch, W., Kleidorfer, M., Leimgruber, J., & Ertl, T. (2016). Lost in calibration: Why people still do not calibrate their models, and why they still should - a case study from urban drainage modelling. *Water Science and Technology*, *74*(10), 2337–2348. https://doi.org/10.2166/wst.2016.395

Tsiami, L., & Makropoulos, C. (2021). Cyber—Physical attack detection in water distribution systems with temporal graph convolutional neural networks. *Water (Switzerland)*, *13*(9). https://doi.org/10.3390/w13091247

Ulanicki, B., Zehnpfund, A., & Martinez, F. (1996). Simplification of water distribution network models. *September*. https://doi.org/10.13140/RG.2.1.4340.8404

UNESCO. (2020). *United Nations world water development report 2020: Water and climate change*.

University of Kentucky. (2013). *Water distribution system research database*, University of Kentucky. https://doi.org/10.13023/kwrri.wdsrd

Van Bijnen, M., Korving, H., Langeveld, J., & Clemens, F. (2017). Calibration of hydrodynamic model-driven sewer maintenance. *Structure and Infrastructure Engineering*, *13*(9), 1167–1185. https://doi.org/10.1080/15732479.2016.1247287

Verma, S., & Zhang, Z.-L. (2019). *Learning universal graph neural network embeddings with aid of transfer learning*. *ArXiv*. Retrieved from https://arxiv.org/abs/1909.10086

Vogado, L. H. S., Veras, R. M. S., Araujo, F. H. D., Silva, R. R. V., & Aires, R. T. (2018). Engineering applications of artificial intelligence leukemia diagnosis in blood slides using transfer learning in CNNs and SVM for classification. *Engineering Applications of Artificial Intelligence*, *72*, 415–422. https://doi.org/10.1016/j.engappai.2018.04.024

Vojinovic, Z., Kecman, V., & Babovic, V. (2003). Hybrid approach for modeling wet weather response in wastewater systems. *Journal of Water Resources Planning and Management*, *129*(6), 511–521. https://doi.org/10.1061/(ASCE)0733-9496(2003)129:6(511)

Wang, Q., Guidolin, M., Savic, D., & Kapelan, Z. (2015). Two-objective design of benchmark problems of a water distribution system via MOEAs: Towards the best-known approximation of the true pareto front. *Journal of Water Resources Planning and Management*, *141*(3), 04014060. https://doi.org/10.1061/(asce)wr.1943-5452.0000460

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*. https://doi.org/10.1186/s40537-016-0043-6

Wolfs, V., & Willems, P. (2017). Modular conceptual modelling approach and software for sewer hydraulic computations. *Water Resources Management*, *31*(1), 283–298. https://doi.org/10.1007/s11269-016-1524-2

Wong, E., & Kolter, J. Z. (2019). *Learning perturbation sets for robust machine learning* (Vol. 1–32). Retrieved from https://arxiv.org/pdf/2007.08450.pdf

Wu, L., Zulueta, K., Major, Z., Arriaga, A., & Noels, L. (2020). Bayesian inference of non-linear multiscale model parameters accelerated by a deep neural network. *Computer Methods in Applied Mechanics and Engineering*, *360*, 112693. https://doi.org/10.1016/j.cma.2019.112693

Xing, L., & Sela, L. (2022). Graph neural networks for state estimation in water distribution systems: Application of supervised and semi-supervised learning. *Journal of Water Resources Planning and Management*, *148*(Accepted), 1–14. https://doi.org/10.1061/(ASCE)WR.1943-5452.0001550

Yates, D. F., Templeman, A. B., & Boffey, T. B. (1984). The computational complexity of the problem of determining least capital cost designs for water supply networks. *Engineering Optimization*, *7*(2), 143–155. https://doi.org/10.1080/03052158408960635

Ying, R., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). GNNExplainer: Generating explanations for graph neural networks. *Advances in Neural Information Processing Systems*, *32*(iii).

Yoon, S., Lee, Y.-J., & Jung, H.-J. (2020). Accelerated Monte Carlo analysis of flow-based system reliability through artificial neural network-based surrogate models. *Smart Structures and Systems*, *26*(2), 175–184. https://doi.org/10.12989/sss.2020.26.2.175

Zhang, D., Mishra, S., Brynjolfsson, E., Etchemendy, J., Ganguli, D., Grosz, B., et al. (2021). *AI Index report*. (pp. 1–222). Retrieved from https://aiindex.stanford.edu/report/

Zhang, W., Li, J., Chen, Y., & Li, Y. (2019). A surrogate-based optimization design and uncertainty analysis for urban flood mitigation. *Water Resources Management*, *33*(12), 4201–4214. https://doi.org/10.1007/s11269-019-02355-z

Zhu, Y., & Zabaras, N. (2018). Bayesian deep convolutional encoder – Decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, *366*, 415–447. https://doi.org/10.1016/j.jcp.2018.04.018