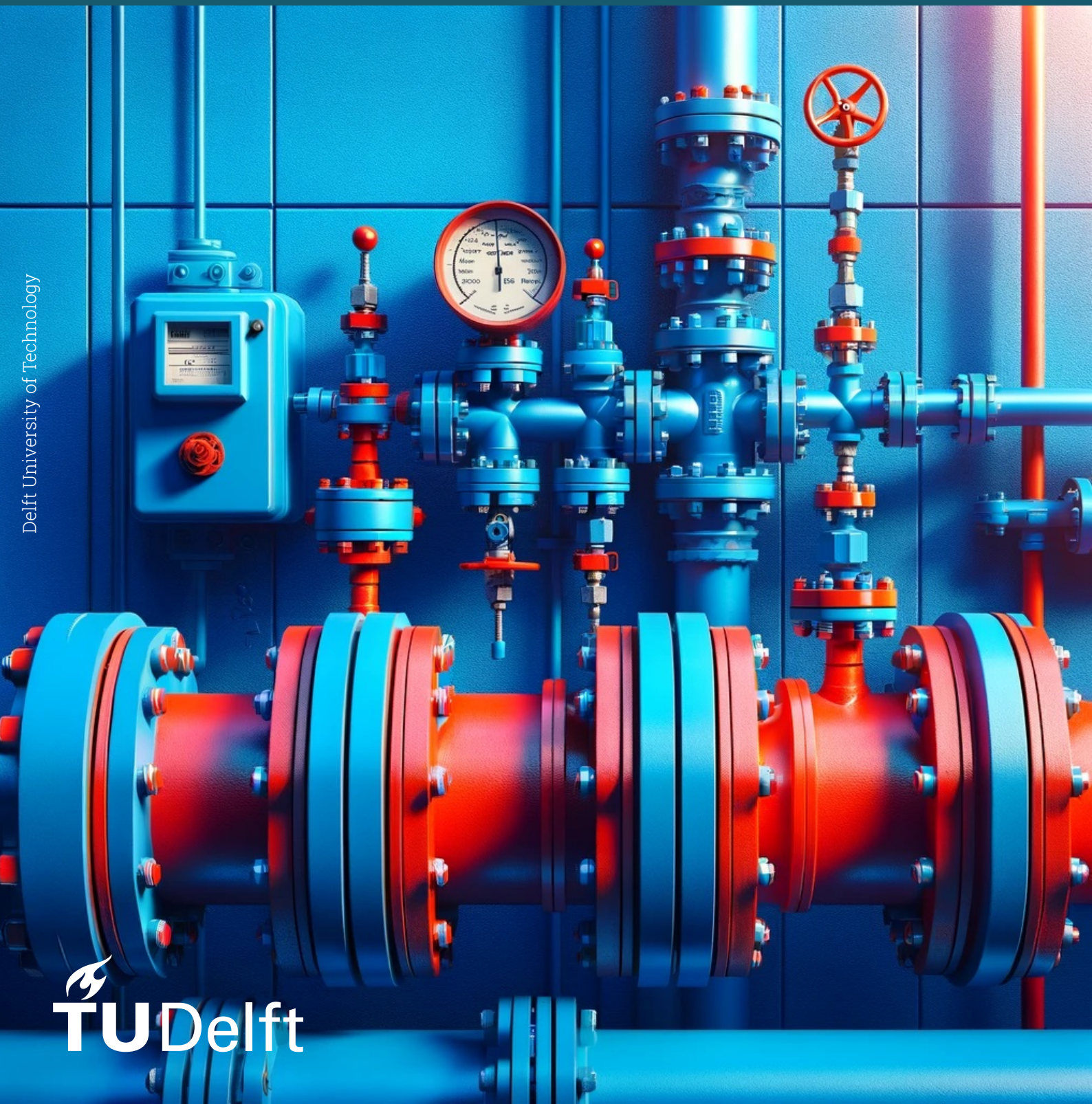
MSc Thesis

Optimizing the pump schedule of water distribution systems using a deep learning meta-model

CIE5060-09

Nikos Mertzanis



Delft University of Technology

TU Delft

Optimizing the pump schedule of water distribution systems using a deep learning meta-model

To what extent can algorithm unrolling optimize the pump schedule of an urban water distribution system?

by

Nikos Mertzanis

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Friday March 15, 2024 at 3:45 PM.

Student number:	4840256	
Faculty:	Civil Engineering & Geosciences	
Project duration:	May 1, 2023 – March 15, 2024	
Thesis committee:	Dr. ir. R. Taormina,	TU Delft, main supervisor
	Dr. ir. M. A. Schleiss,	TU Delft, secondary supervisor
	PhD. Candidate J. A. Garzón Díaz,	TU Delft, additional supervisor

Cover: Generated by DALL-E

An digital version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

To my mother. Who never stopped believing in me.

Nikos Mertzanis
Delft, March 2024

Abstract

This thesis investigates the integration of algorithm unrolling and genetic algorithms (GA) for optimizing pump scheduling in water distribution systems (WDS), a critical component for ensuring energy-efficient water delivery. In the context of modern civilization's reliance on clean, affordable water for diverse uses, the operation of a WDS, particularly through energy-intensive pumps, presents significant challenges. Traditional optimization techniques often resort to hydraulic solvers like EPANET, which, while accurate, are computationally intensive for large-scale applications. Our methodology introduces a meta-model based on algorithm unrolling, building upon prior work and extending it to address pump scheduling with a multi-objective function focusing on both cost and energy efficiency. This approach significantly reduces the computational load, offering a faster alternative to EPANET while maintaining considerable accuracy. The meta-model demonstrated promising results in the Fossolo network, achieving comparable schedules 20 times faster than traditional methods. However, its applicability to more complex networks and its ability to capture detailed system behaviors are limited, highlighting the need for further enhancements in model stability and reproducibility. Despite these limitations, the study emphasizes the potential of meta-models as a complementary tool to traditional methods, especially in scenarios requiring rapid decision-making under computational constraints. This research contributes to the broader field of water utility management, offering insights into more sustainable and efficient operation strategies.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
ACO	Ant Colony Optimization
AI	Artificial Intelligence
API	Application Programming Interface
AU	Algorithm Unrolling
ANN	Artificial Neural Network
BU	BaselineUnrolling
DDA	Demand Driven Analysis
GA	Genetic Algorithm
GGA	Global Gradient Algorithm
GHG	Greenhouse Gasses
GNN	Graph Neural Network
LFPB	Lower-Fidelity Physically-Based
LPS	Liters-Per-Second
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Square Error
NSGA	Non-Dominated Sorting Algorithm
PDA	Pressure Driven Analysis
PyMOO	Python Multi-Objective Optimization
PySIMDEUM	Python SIMulation of water Demand, and End-Use Model
RMSE	Root Mean Square Error
RS	Response Surface
SGD	Stochastic Gradient Descent
UM	UnrollingModel
WDN	Water Distribution Network
WDS	Water Distribution System
WNTR	Water Network Tool for Resilience

1

Introduction

Modern civilization relies on the provision of clean, affordable water, for both direct consumption, such as domestic use, and indirectly through irrigation, sanitation, and industrial usage. To transport and distribute this water we use water distribution systems (WDS). In general, the operation of a WDS refers to the management and control of all tasks required to deliver water from a reservoir, or another storage facility, to its consumers (Buenfil 2004). These include water collection, treatment, transportation, and delivery. When delivering clean water to houses, the system needs to be constantly pressurized so that if, at any moment, a user chooses to open a tap there will be water flowing out. A simple way to achieve this is by ensuring that all destinations where water is delivered are at a lower elevation from where the water is stored. The water is then delivered through the natural pressure gradient caused by height difference. However, this is not always a possibility, especially for areas with little topographical variation. In cases such as this, the combined use of pumps and tanks can help maintain pressure in the system.

Operating these pumps is the most expensive among the standard network components; as a result of their energy usage over time (Ormsbee et al. 1994). Pump operations represent 40% of total urban electricity consumption (Masłoń et al. 2020) and 2-3% of all energy usage worldwide (Sarbu 2016). The energy used by pumps can usually be reduced by more than 25%, while still delivering an adequate amount of pressure in the system (Moreira et al. 2013), thereby leading to savings in both costs and energy. To reduce energy usage and increase efficiency, pump operators of a WDS aim to determine the optimal times to switch pumps on or off. This problem is referred to as Pump Scheduling (PS) and its search space tends to be large for WDSs of a substantial size. An increase in efficiency is usually interpreted as a reduction of cost, with the cost of energy being variable throughout the day (Giacomello et al. (2013) and Bohórquez et al. (2015)). This variation in energy prices means that the most cost-effective pump operation schedule may, for example, favor nighttime operation when energy tends to be cheaper at the expense of higher energy consumption in total. Thus, the most cost-efficient operation is not necessarily the most energy-efficient. Besides optimizing operational expenses based on energy price, effective pump scheduling can take energy usage directly into account.

Determining an efficient pump schedule can be framed as an optimization problem. Optimization problems have three main components. An objective function, any number of decision variables, and constraints. The objective function encapsulates the goals of the optimization task. This could be, for example, to minimize or maximize a certain metric, like cost. Potential solutions to the problem are modeled through the decision variables. These can be discrete or continuous variables that describe a solution to the optimization problem. Checking whether a solution is valid involves ensuring that it satisfies the constraints of the optimization problem, such as ensuring one or more variables of the system stay within certain boundaries. Checking whether a valid solution is optimal against the rest, otherwise called a non-dominated solution, requires calculating its objective function values (Chong et al. 2023). The solution which is not dominated by any other solutions in the search space is the global optimum. In a multi-objective problem, there can be multiple such non-dominant solutions based on which objective is prioritized. The total of these solutions forms the Pareto frontier.

In the realm of optimization, conventional strategies typically model the problem's parameters via their mathematical relations, subsequently striving to identify the optimal solution in alignment with the optimization's objectives. Conversely, alternative methodologies skip this analysis, opting instead to directly generate and evaluate potential solutions. These are heuristic approaches, like Genetic Algorithms (GA), that are widely used and tend to have a good performance for optimization of complex problems (Yang 2014).

Genetic Algorithms (GA) is the heuristic approach selected for this thesis. These are algorithms that mimic natural selection to evolve near-optimal solutions (Michalewicz 1996). Most notably, they have also been successfully used to solve the pump scheduling problem, such as in Mackle et al. (1995). They were preferred due to their success in various fields such as "facility layout problem (FLP), supply network design, scheduling, forecasting, and inventory control" (Katoch et al. 2021). GAs can be beneficial for dealing with challenging problems, such as ones that have a large search space (Yang 2014). The search space of pump scheduling is all possible configurations/settings of the pumps in a WDS. Finding the optimal schedule is not straightforward due to the combinatorial growth of the search space with an increasing amount of pumps. Common approaches for operations problems in WDS tend to use heuristics (Maier et al. 2014), indicating that due to the complexity of obtaining an analytical solution, their solution should be approximated instead. As stated by Jowitt et al. (1992), "with the exception of very simple water-distribution systems, the problem of optimal pump scheduling is one of high complexity for which a formal approach is required". Finding the optimal pump schedule is "generally non-convex, and hence NP-hard" (Fooladivanda et al. 2015). This holds true for pump schedule optimization, which has been successfully achieved in the past using heuristic methods such as Genetic Algorithms (GA) and Ant Colony Optimization (ACO) (Bohórquez et al. (2015); López-Ibáñez et al. (2008)).

Traditional approaches to optimization of pump scheduling involve combining any of the numerous existing optimization algorithms with some hydraulic solver, like the one focused on in this thesis, EPANET, to perform optimization on this simulated counterpart. Nevertheless, "direct application of hydraulic solvers is computationally intensive when applied to models of large utilities" (Behandish et al. 2014).

Over recent decades, new approaches that leverage machine learning, especially artificial neural networks (ANNs), can replace software like EPANET for computationally intensive applications where the original model performance is not adequate, such as in pump scheduling of large WDSs. They provide an approximation of the original model's output in a fraction of the time (Razavi et al. 2012). This practice is known as surrogate modeling or meta-modeling, and it achieves a notable increase in processing speed at the expense of decreased accuracy when compared to the original model. The meta-models are not performing simulation but prediction, meaning that they are trained on data from the original model and learn how to predict its output. This can produce inaccurate results as it is not restricted by physical laws like numerical solvers such as EPANET. Still, with a rigorous training process, the output can be accurate enough for certain applications and it can be produced up to thousands of times faster, as shown by Broad et al. (2005). Besides not being constrained by physical laws, ANNs are also subject to other inherent limitations. They can be hard to train (curse of dimensionality), obscure when interpreting their output (black-box nature), and inflexible to modifications (rigid structure) (Garzón et al. 2022).

One promising approach proposed by Monga et al. (2021) is algorithm unrolling (AU) which improves upon commonly used ways of structuring ANNs, such as Multi-Layer-Perceptrons (MLP) (Balas 2009). Algorithm unrolling works by transforming iterative algorithms into deep neural networks. When designing ANNs by algorithm unrolling, the model's architecture is reflective of the actual structure of the iterative algorithm being modeled. "Specifically, each iteration of the algorithm step is represented as one layer of the network" Monga et al. (2021). Parameters of the algorithm also correspond to parameters of the ANN, effectively improving interpretability (black-box nature) and reducing reliance on large training sets (curse of dimensionality) (Monga et al. 2021).

One direct application of meta-modeling in the optimization of pump scheduling is Behandish et al. (2014). The study managed to obtain a 10-15% reduction in daily costs, which is close to the 25% margin stated by Moreira et al. (2013). A combination of meta-models was used to predict the head of the nodes in the network. The final meta-model is composed of multiple MLPs each dedicated to predicting specific parts in the network, like the tanks or a sub-section of nodes. An ensemble of those

predictions is used to obtain the final value. This works well in the case of the network examined but has two pitfalls. First, the regular pitfalls of MLPs, as detailed in section 2.4, are not addressed. Algorithm unrolling manages to alleviate some of those pitfalls, as explained in section 2.6. Second, the model architecture presented is highly case-dependent. Replicating the work on a new network would require constructing a different meta-model after analyzing the network, and training dedicated sub-ANNs for parts of the network, which would be time-consuming. It would be more resource-efficient if a meta-model could be directly trained on any given WDS's data, without custom modifications to its structure. In that case, the same model can be used for training after changing its input/output structure based on the WDS. No other network analysis is required. Finally, Behandish et al. (2014) used the single-objective of cost and limited the total possible pump switches in a schedule to four. Pump switches are used to represent the maintenance cost of the pumps since activation/deactivation of mechanical devices is detrimental to the wearing down of their components.

In this thesis, a multi-objective function is used, combining energy and cost. The global issues faced due to climate change are only increasing as mitigation remains limited, highlighted by the 2023 IPCC report (Lee et al. 2023). Thus, cost is no longer the only factor being minimized, but energy expenditure is considered equally important in terms of optimization. Energy here is referring indirectly to greenhouse gas emissions (GHG) since most of the energy grids worldwide are still powered by fossil fuels (IEA 2023). Additionally, no pump switch limitation is introduced with pump switches still being accounted for in the objective function. Our meta-model is based on algorithm unrolling, which manages to partially overcome the traditional ANN limitations. The process of algorithm unrolling also does not require analysis of the WDS being modeled.

The extent to which algorithm unrolling helps to optimize the pump schedule of an urban water distribution system can be explored through the following research questions.

- What is the performance of a meta-model of EPANET created via algorithm unrolling in terms of accuracy and speed?
  - How replicable is this procedure across different water distribution systems?
- To what extent can the meta-model be reliably used to optimize pump schedules?

In this thesis, we introduce an existing meta-model based on algorithm unrolling to accelerate the optimization of pump scheduling. We build our meta-model upon the previous work of Solà Roca (2023). To the best of our knowledge, no other work has investigated algorithm unrolling in this context. We adapted and extended the work of Solà Roca (2023) so that pump scheduling can be performed using the meta-model and a genetic algorithm. The performance of the pump schedules found during this optimization was compared against the obtained solutions from a GA, using EPANET as a hydraulic simulator. Our method using the meta-model is capable of finding comparable schedules, 20 times faster.

2

Background

2.1. Water Distribution Systems

Drinking water distribution systems (WDS) generally consist of numerous interconnected pipes, valves, pumps, and tanks. Water supply is sourced from reservoirs, including natural water bodies like lakes and rivers, or artificial storage facilities. It is then processed and made safe to drink by a treatment plant before being delivered to the users of the system. This thesis focuses on urban drinking water distribution systems. Characteristics of these systems include that they are closed, with no direct contact with the air, and that they deliver pressurized, treated water that is safe for human consumption. Moreover, systems with at least one pump are explored, where the pump schedule can be optimized. WDSs with pumping stations typically also incorporate tanks, acting as local buffers to address fluctuations in demand or enhance water availability.

Pumps have a different behavior to standard links of the network as they lead to an increase in the head and/or flow of the water traveling through the pipe. Adding to their complexity, their operation is controlled, which is how the pump schedule is reflected on the network. Some pumps are binary when it comes to their operation (on/off), while others can have multiple speed settings. There are also pumps whose speed curve can be continuous and adjusted with a dial.

2.1.1. Optimization of pump scheduling

Due to the mathematical nature of NP-hard problems leading to a computational burden as the complexity of the network increases, a global operational optimum can usually only be approximated. The possible network solutions are displayed as components of a multivariate analysis where cost is usually the main factor used to rank solutions, although, for this thesis, energy itself, will be of equal importance.

Optimization problems have three main components. An objective function, any number of decision variables, and constraints. For the definition of the pump scheduling problem in this thesis, the objective function to be minimized is energy and cost. The decision variables are 24 binary decision variables for every pump, representing every hour of the day that the pump is turned on or off. The constraint of the optimization problem is maintaining a minimum level of pressure for all users of the system, enough to ensure water delivery. Unless this restriction is satisfied, a solution is considered incorrect. The set of all non-dominated solutions, which cannot be further improved, forms the Pareto frontier, an example of which is displayed in Figure 2.1.

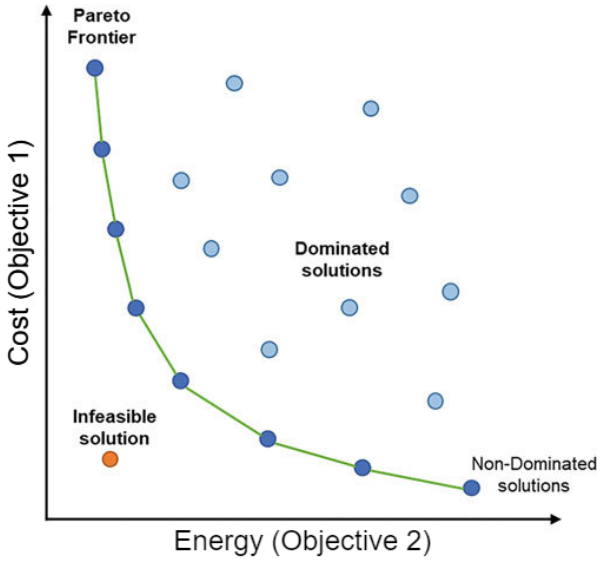


Figure 2.1: Pareto front adapted from Liu et al. (2021)

Referring to costs in the domain of pump scheduling can be misleading. It is not only the cost of the energy that a pump is consuming but also the cost of maintaining the pump itself. Depending on the manner it is operated, such as the amount of time it is run, a pump's life can be extended. To simplify this multi-faceted problem one of the most important factors, the amount of switches, is considered. Following Behandish et al. (2014), a switch is the change of a pump's state from off to on, but not vice versa. A high number of pump switches leads to increased pump maintenance (Makaremi et al. 2017).

2.1.2. EPANET and WNTR

EPANET was developed by the United States Environmental Protection Agency to model WDS (Rossman et al. 2020). It is a standalone open-source application that can analyze the hydraulics and the water quality throughout the network. The standalone application has a graphical user interface and is ideal for a water utility operator or an engineer seeking to conduct such analyses. Nevertheless, it is not possible to use it directly via computer code. For this, a Python package called WNTR, abbreviating the "Water Network Tool for Resilience" (Klise et al. 2018), was developed. WNTR is an application programming interface (API) that provides precisely this functionality.

2.1.3. Global Gradient algorithm

As stated in the EPANET documentation,*"the solution for heads and flows at a particular point in time involves solving simultaneously the conservation of flow equation for each junction and the head loss relationship across each link in the network"* (Rossman et al. 2020). EPANET employs the Global Gradient Algorithm (GGA) to reach a solution. This algorithm was created by Ezio Todini et al. (2013), who suggest that the global algorithm is "the most attractive approach out of the various formulations" of steady-state hydraulics. It has two forms, Newton-Raphson (NR) or Linear Theory (LT) with EPANET implementing the former.

Starting with the basic equations describing a water distribution system, first is the conservation of mass (Ezio Todini et al. 2013).

$$\sum_{k=1}^{n_i} q_{k,j} + s_i = 0 \tag{2.1}$$

where $q_{k_{i,j}}$ is the flow in the pipe $k_{i,j}$ connecting node i to j; n_i is number of pipes connected to node i; and s_i is the known demand at node i. The sign convention states that flow into a node is positive while flow out of a node is negative.

Then, the conservation of energy:

$$h_i - h_j - r_k|q_k|^{\alpha-1}q_k = 0 \tag{2.2}$$

where h_i and h_j are the hydraulic heads at nodes i and j respectively; q_k is the flow rate in pipe k; r_k is a coefficient that depends on the pipe characteristics and flow rate; and α is an exponent, whose value is 1.852 when using the Hazen-Williams equation.

E. Todini et al. (1988) restate equations 2.1 and 2.2 in the following matrix notation, which forms the basis of their GGA representation.

$$\begin{bmatrix} A_{11} & \cdots & A_{12} \\ \vdots & \ddots & \vdots \\ A_{21} & \cdots & 0 \end{bmatrix} \begin{bmatrix} q \\ \vdots \\ h \end{bmatrix} = \begin{bmatrix} -A_{10}h_0 \\ \vdots \\ -s \end{bmatrix} \tag{2.3}$$

where

$$\begin{aligned} q^T &= [q_1, q_2, \ldots, q_p], \text{the [1, p] unknown pipe discharges} \\ h^T &= [h_1, h_2, \ldots, h_n], \text{the [1, n] unknown nodal heads} \\ h_0^T &= [h_{n+1}, h_{n+2}, \ldots, h_N], \text{the [1, N - n] known nodal heads} \\ s^T &= [s_1, s_2, \ldots, s_n], \text{the [1, n] known nodal demands} \end{aligned}$$

with p as the number of pipes, n as the number of unknown head nodes, N as the total number of nodes in the network, and finally $N - n$ (or n_0) as the number of nodes with known head.

In Equation 2.3 A_{11} is a diagonal matrix whose elements are defined as:

$$A_{11}(k,k) = r_k|q_k|^{\alpha-1} \tag{2.4}$$

for $k \leq p$; $i \in 1, N$; $j \in 1, N$, where k is the index of the pipe connecting two generic nodes i, j. Then, A_{12} is a $[p; n]$ matrix relating the pipes to the unknown head nodes and A_{10} is a $[p; n_0]$ matrix relating the pipes to the known head nodes. The concatenation of these matrices is $\hat{A}_{12}$ and is defined as:

$$\hat{A}_{12}(i,j) = \begin{cases} -1 & \text{if pipe } i \text{ leaves node } j \\ 0 & \text{if pipe } i \text{ is not connected to node } j \\ +1 & \text{if pipe } i \text{ enters node } j \end{cases} \tag{2.5}$$

$$\hat{A}_{12} = \begin{bmatrix} A_{12} & \vdots & A_{10} \end{bmatrix} \tag{2.6}$$

In the same book section by E. Todini et al. (1988), the analytical iterative solution of the GGA is derived resulting in two fundamental iterative equations:

$$h^{(k+1)} = A^{(k+1)^{-1}} f^{(k+1)} \tag{2.7}$$

$$q^{(k+1)} = q^{(k)} - D^{(k+1)^{-1}} \left(A_{11}^{(k+1)^T} q^{(k)} + A_{12} h^{(k+1)} + A_{10} h_0 \right) \tag{2.8}$$

where $f^{(k+1)}$ is the net flow imbalance at the nodes and matrix D is defined as:

$$D(k_i, k_i) = \begin{bmatrix} \alpha_{k_i} r_{k_i} |q_{k_i}|^{\alpha-1} & 0 & \cdots & 0 \\ 0 & \alpha_{k_i} r_{k_i} |q_{k_i}|^{\alpha-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_{k_i} r_{k_i} |q_{k_i}|^{\alpha-1} \end{bmatrix}$$

These non-linear equations represented in matrix notation are solved iteratively, meaning that EPANET makes an initial guess at the solution and, through iterating over time, it converges at the correct state of the network where flow and pressure discrepancies are minimal (Rossman et al. 2020). This can be time-consuming, mostly for large or complex networks. In some cases, the method might not converge at all or converge to an incorrect solution if the initial guess is too far from the true solution. Thus, while EPANET is capable of accurately modeling a water distribution system it requires computational resources to do so, especially if the system is large. The amount of resources required becomes the bottleneck when iteratively examining system implementations.

2.2. Optimization strategies

For finding optimal pump schedules in WDSs, there are many optimization strategies such as ”Linear programming (LP), Non-Linear Programming (NLP), Mixed Integer Non-Linear Programming (MINLP), Neutral Search, Dynamic Optimization (DO), Dynamic Programming (DP), and also heuristic and evolutionary techniques such as Genetic Algorithms (GA) and Ant Colony Optimization (ACO)” (Castro-Gama et al. 2017). All of these strategies involve evaluating the pump schedule determined, meaning calculating the heads and flows in the WDSs and determining whether water demands are satisfied. This suggests faster evaluation can improve performance beyond EPANET’s capabilities. The first set of strategies, up to and including DP, are deterministic optimization techniques. The latter set of strategies is non-deterministic and has shown promise in the area of hard combinatorial optimization problems (Latin American Computing Conference et al. 2004).

2.3. Genetic Algorithms

The strategy selected for this thesis is Genetic Algorithms, which starts by selecting a random subset of solutions. The objective function is then evaluated for this subset to determine their performance. Evolutionary strategies are then followed such as mating and survival selection (Deb et al. 2002) to ensure solutions close to the optimum are retained while continuing with exploration so that the optimizer does not remain fixed in local optima. Genetic Algorithms have been proven to be good heuristic solvers for such problems (Makaremi et al. (2017); Niu et al. (2018); Do (2023)). In cases like pump scheduling, where the search space of possible pump configurations is usually too large to be exhaustively explored, GAs can provide a good solution. At the same time, as always with heuristic methods, the best solution found is not guaranteed to be the global optimum.

This strategy can highly benefit from a faster scenario evaluation since its performance relies directly on the ability to evaluate as many possible pump scheduling configurations as possible. The solution found by a GA improves as more generations of it are run. When formulating pump scheduling as a constrained multi-objective problem, any multi-objective genetic algorithm, that allows for an arbitrary number of decision variables could suffice. We selected NSGA-II, or otherwise Non-dominated Sorting Genetic Algorithm (Deb et al. (2002); Blank et al. (2020)). It is considered to be one of the best GAs in terms of accuracy and performance (Deb et al. 2002).

2.4. Meta-modeling

Speeding up the ability to examine different scenarios for a given hydraulic system greatly increases the potential to solve problems regarding design and operations. Thus, meta-modeling or surrogate modeling ”is concerned with developing and utilizing cheaper-to-run surrogates of the original simulation models” (Razavi et al. 2012). There are two main types of meta-models; lower-fidelity physically-based meta-models (LFPB) and response surface (RS). LFPB meta-models are a simplification of the original physically-based model, while RS meta-models replace the model completely with some different, faster technique to calculate the pressure differences (Razavi et al. 2012). There are multiple techniques to construct an RS meta-model, with some of the most computationally expensive ones being Kriging, artificial neural networks (ANNs), and support vector regression (Fernández-Godino 2023).

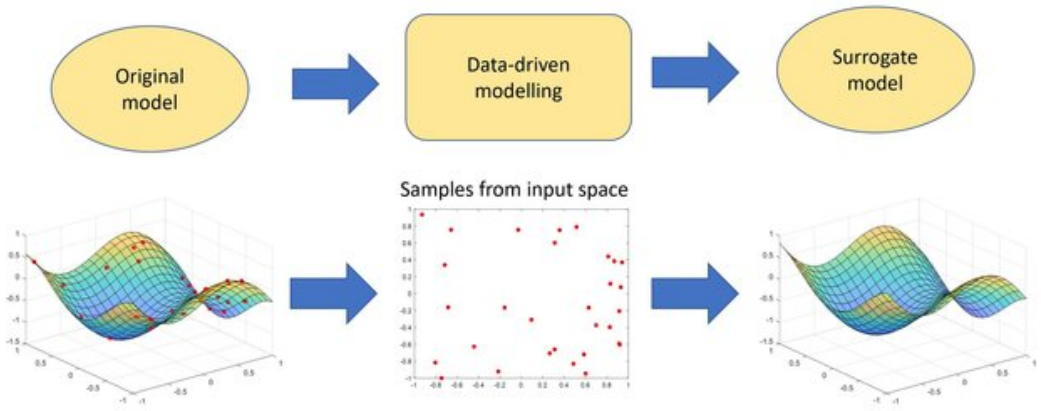


Figure 2.2: Example construction of RS surrogate model (Kocijan et al. 2022)

As shown in Figure 2.2, the meta-model is provided with data samples and, through training, learns the structure of the input space. Then, instead of running the original model, the meta-model can be probed to provide an answer from this input space in a fraction of the time. Thus, in a problem of optimization, the outcome of the meta-model can be used to filter through numerous scenarios and remove the ones that are not shown to be close to ideal for the problem being addressed. Then, this small remaining subset of the search space can be directed back to the original physical model to ensure that it is comprised of near-optimal, physically valid solutions.

2.5. Machine-learning based meta-models

In this thesis, the meta-model used falls within the Response Surface (RS) category of machine learning-based (ML) models, specifically employing a deep-learning artificial neural network (ANN) approach. Deep learning, a subfield of machine learning, refers to ANNs that incorporate inductive biases, which are designed to emulate the hierarchical structure inherent in the data, here referring to water distribution networks (Battaglia et al. 2018). While ANN-based meta-models are among the most promising techniques in this area, their development, particularly the training process ”is severely handicapped in limited computational budgets” (Razavi et al. 2012). These meta-models are trained using data from an established physical simulation model, like EPANET, aiming to replicate its output with high fidelity and enhanced speed for system applications post-training (Alizadeh et al. 2020). Although they may not match the precision of the original simulation models, meta-models typically offer sufficient accuracy and significantly faster performance for evaluating various scenarios, as demonstrated by Behandish et al. (2014) and Solà Roca (2023).

There are multiple types of artificial neural networks such as Convolutional, Recurrent, Multi-layer Perceptrons, and others, each optimized for different sets of applications (Madhiarasan et al. 2022). Garzón et al. (2022) state that multi-layer perceptrons (MLP) are the most popular for machine learning-based surrogate modeling in urban water networks due to them being easy to implement, combined with multiple successes in previous applications. This has led to MLPs being considered a one-size-fits-all solution, even though they suffer from some of the classical issues of AI-based models. These include the curse of dimensionality, a black-box nature, and a rigid structure, each with implications in the resulting model (Garzón et al. 2022).

Curse of dimensionality The curse of dimensionality indicates that, as the dimensions of a problem increase, there can be an exponential increment in the data required to model the problem. When surrogate modeling EPANET, this roughly translates to; the number of data samples necessary to model a WDS increases exponentially with its size. Besides size, this property also relates to the complexity of the WDS. The data required to capture the behavior of a WDS with pumps, tanks, and valves is exponentially larger than one without, as these elements represent another dimension of the input space.

Black-box nature The property of having a black-box nature refers to the lack of transparency in the predictions of ANNs (Buhrmester et al. 2021). Simple ANNs can usually not be broken down into their constituents so that the factors of the input can be linked to the prediction. Effectively, this means that when an ANN is making a prediction, we cannot interpret the reasoning behind it. This can be problematic ranging from decreasing confidence in a network's predictions to being genuinely dangerous, such as in the case of self-driving cars and clinical applications (Z. Zhang et al. 2018), (Utesch et al. 2020).

Rigid structure This disadvantage of traditional machine-learning-based surrogate models can be framed as a lack of generalizability. Since training an ANN on a WDS of EPANET essentially teaches the Global Gradient Algorithm to EPANET, then it is natural to expect that this model applies to other WDSs as well. There is a false assumption underlying this logic. Constructing a surrogate model using an ANN does not necessarily make the ANN learn the underlying algorithm used, but only how a specific WDS behaves using this algorithm and its inputs. This leads to surrogate models based on ANNs being case-specific and requiring a different training process for every WDS being meta-modelled (Garzón et al. 2022).

2.6. Algorithm Unrolling

Algorithm unrolling (or unfolding) is a technique of transferring an iterative algorithm into an ANN architecture (Monga et al. 2021). Unlike traditional ANNs, where the number and function of layers are often determined through trial-and-error during hyper-parameter optimization, algorithm unrolling purposefully constructs each ANN layer to mirror an iteration step of the original algorithm. This method, detailed by Monga et al. (2021), involves aligning the layers of the ANN with the iterations of the algorithm, thereby creating a deep neural network whose architecture inherently reflects the algorithmic process it is modeling. The resulting model preserves the inductive biases of the iterative process, enabling the direct optimization of network parameters against real-world data. An example of how algorithm unrolling works is displayed in Figure 2.3. During learning via backpropagation, the network's layers learn to behave like the original model's iterative algorithm, bringing the output one step closer to the solution with every layer evaluated. Errors in the output can be directly linked to components of the network. In other words, the process of unrolling overcomes the limitation of being a black-box to an extent as "the trained network can be naturally interpreted as a parameter-optimized algorithm, effectively overcoming the lack of interpretability in most conventional neural networks" (Monga et al. 2021).

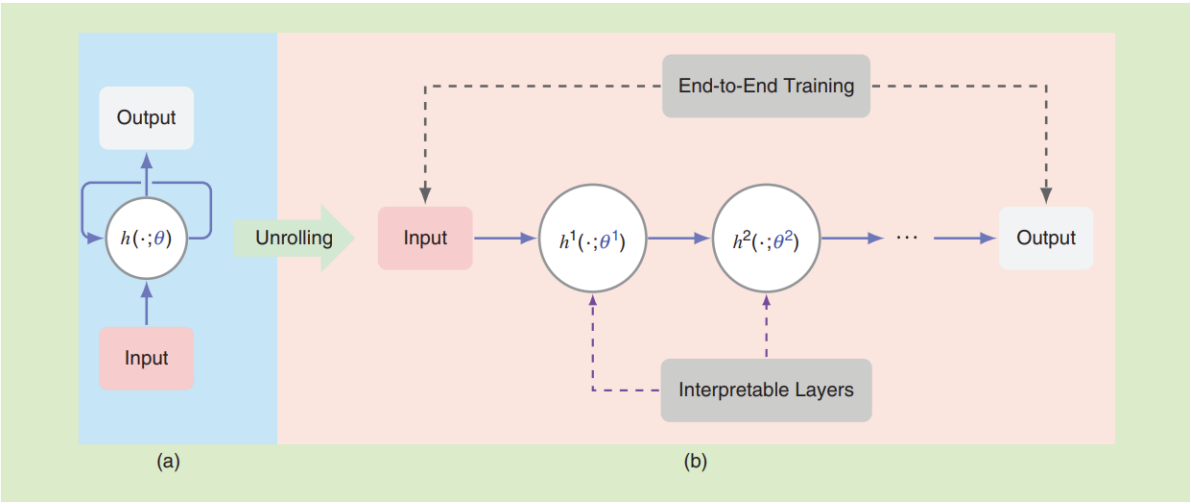


Figure 2.3: A high-level overview of algorithm unrolling by Monga et al. (2021). (a) represents an iterative algorithm and (b) the corresponding deep network generated by cascading the algorithm's iterations. The blue θ refers to the algorithm's parameters in (a) and to network parameters learned through end-to-end training in (b)

2.6.1. GGANet - Steady state meta-modeling of WDS

Our research builds on the work presented in the thesis of Solà Roca (2023). This thesis experimented with creating different models, aiming to do better than current methods of meta-modeling EPANET. Among these developed models, one followed the traditional Multi-Layer Perceptron (MLP) framework. The rest, however, employed innovative strategies for building neural networks. One of these innovative models is a Graph Neural Network, which showed modest promise in the context of Water Distribution System (WDS) meta-modeling. Another model, characterized by its deep unrolled ANN structure, produced promising results. It is this latter model, due to its potential, that we have chosen as the starting point for our current study.

In the work of Solà Roca (2023), there were two unrolling approaches implemented. The first one is called BaselineUnrolling (BU) and largely follows the paradigm presented in Figure 2.3. The main difference is that instead of creating one meta-model layer per iteration, there are two. One represents the heads and another the flows. These are deliberate choices that aim at reproducing these intermediate variables, in the same way as the GGA does. The algorithm of GGA requires the heads (2.7) and flows (2.8) to be calculated iteratively.

After having identified these two stages of the GGA, the matrix f in Equation 2.7 and D in 2.8 are treated as learnable parameters Φ and Ψ.

$$h^{(k+1)} = \Phi(q^{(k)}, x_s) \tag{2.9}$$

$$q^{(k+1)} = q^{(k)} - \Psi(h^{(k+1)}, x_s) \tag{2.10}$$

where $h^{(k+1)}$ and $q^{(k+1)}$ correspond to the approximations of the heads and flows at step $k+1$ of the iterative algorithm, x_s are the system features of the WDS and $\Phi(\cdot)$ and $\Psi(\cdot)$ represent the functions that update the hydraulic variables q and h, respectively. The heads are updated from the flows and vice versa (Solà Roca 2023).

The complexity of the learnable parameters Φ and Ψ can vary depending on the unrolling implementation. For the BU model implemented in Solà Roca (2023), Φ and Ψ are represented by single-layer MLPs. Two layers were constructed per iteration step, predicting the output of the flows and heads. Hence, the BaselineUnrolling (BU) network ends up having double the amount of layers compared to the iterations required by EPANET. Thus, the unrolling networks used in this thesis are not implemented by exactly following the algorithm unrolling approach outlined by Monga et al. (2021), which indicates constructing one layer per iteration of the original algorithm. In the case of EPANET, where the algorithm tends to converge after around 4-6 iterations (Ezio Todini et al. 2013), the BU implemented requires 8-12 layers.

For the second approach, UnrollingModel, the GGA was reproduced even more closely. This method is composed of first constructing a layer for all of the components of the GGA. Then, the computationally expensive components were identified and a machine learning model, in the form of an MLP, learns a matrix representation of each. This matrix is only an approximation of the original matrix calculated by the GGA. The algorithm is then applied as is, using tensors, but instead of manually calculating these computationally expensive parts, their modeled approximations are used. This allows the model to converge at a solution faster than EPANET. Figure 2.5 illustrates the meta-model's architecture. It is clear that this second approach is more complex and requires a significantly higher amount of layers per iteration than BaselineUnrolling.

In summary, Solà Roca (2023) constructed 3 working models. These are:

- Multilayer Perceptron (MLP)
- BaselineUnrolling (BU)
- UnrollingModel (UM)

each with an increasing degree of complexity and adaptation of the mathematically-based structure of EPANET.

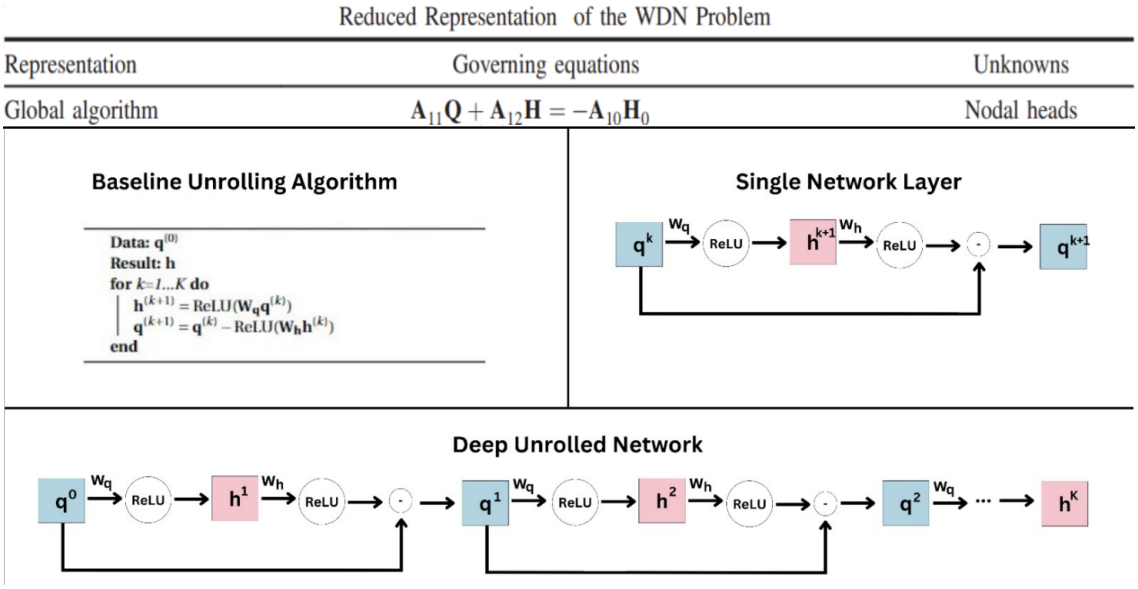


Figure 2.4: BaselineUnrolling based on Ezio Todini et al. (2013), implemented by Solà Roca (2023)

Different approaches to constructing ANNs, such as algorithm unrolling, aim to address the three limitations of traditional ANNs; their black-box nature, rigid structure, and the curse of dimensionality. The resulting networks used in this thesis (BU, UM) manage to mitigate the curse of dimensionality. The rigid structure and the black-box nature are also lessened, but only to a limited extent.

The implications of a black-box nature are the inability to interpret and explain the network's behavior as, traditionally, most ANN parameters do not have a corresponding counterpart in the algorithm they are modeling. This is not exactly the case for the models of Solà Roca (2023). In both models, the layers for heads correspond to the heads of the nodes in the network, but the flows do not correspond to the flows through the pipes and, similarly, the parameters of the layers do not correspond to any parameters of the GGA. In this way, the meta-models' functionality is interpretable only to a certain degree and some internal processes can be linked to physically based ones indicated by the GGA. For example, for every node of the network, there is a node in the unrolled model whose head is predicted iteratively until the final prediction. However, if a prediction is far from the real value it cannot be interpreted, only traced to the components that contribute to its inaccuracy.

This architecture also relates to the rigid structure of ANNs, meaning the inability to generalize. Unrolled artificial neural networks entail domain knowledge in their structure. In our case, this includes the EPANET equations and the network's structure itself. Models built through algorithm unrolling "tend to generalize better than generic networks, and they can be computationally faster as long as each algorithmic iteration is not overly expensive" (Monga et al. 2021). The UnrollingModel (UM) was not shown to generalize by Solà Roca (2023) but its structure is not fixed. Depending on the WDS it is being trained on, the meta-model's size adapts to the system's. This still means that the UM is case-specific and is highly unlikely to ever become a generalizable model but it could prove to be reproducible, a property distinct to generalizability but related to it.

The third limitation is the curse of dimensionality, the implication of which is that data increases exponentially with the WDS's size. Monga et al. (2021) state that an unrolled iterative algorithm manages to overcome the curse of dimensionality by having fewer parameters than typical ANNs. This is not the case for the meta-model constructed in the research of Solà Roca (2023). The method of unrolling was implemented in such a manner that the network requires more parameters beyond a certain WDS size when compared with most traditional response surface meta-models for EPANET, as is shown in Table 2.1. This also means that training these models requires an extensive training dataset. Nevertheless, because the model's architecture is designed to reflect the GGA's structure, there is a strong inductive bias embedded which mitigates this effect, leading to improved accuracy.

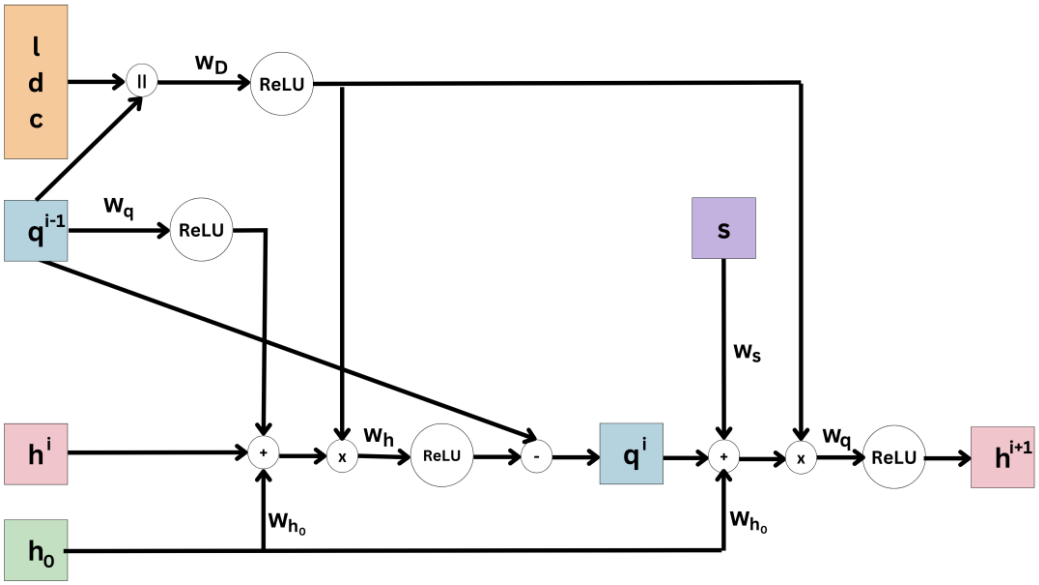


Figure 2.5: Unrolling the GGA, one iteration (Solà Roca 2023)

	RMSE (m)			Speedup			# Parameters		
Network	MLP	UM	GNN	MLP	UM	GNN	MLP	UM	GNN
BAK	0.481	0.339	0.502	1421	732	75	190K	85K	111K
FOS	3.677	1.391	1.572	1752	814	67	261K	88K	247K
PES	5.813	5.079	4.921	2086	2063	82	177K	159K	222K
MOD	1.195	1.010	1.318	2103	841	62	224K	1.34M	247K
RUR	1.105	0.885	1.198	3633	1195	63	297K	3.11M	111K

Table 2.1: Results from Solà Roca (2023). Comparable speedups to an MLP, but a larger number of parameters is required for certain networks

Solà Roca (2023) implemented a hybrid ANN architecture combining elements of data-based ANNs and the technique of algorithm unrolling for the global gradient algorithm. The results in Table 2.1 indicate that the algorithm unrolling (UM) ANN outperforms standard MLP-based approaches for most network examples, while still having comparable speedups.

Currently, all the models constructed by Solà Roca (2023) are only capable of handling three elements of hydraulic systems. Nodes, pipes, and reservoirs. While these can describe a simple hydraulic system, they are the bare minimum and more complicated systems tend to involve multiple other components such as pumps, valves, and tanks. Thus, to optimize a pump schedule, pumps and tanks need to be implemented in the models. For the case of the MLP, which functions simply by associating the input with the output, a simple extension and modification of this data is all that is required. For the other two networks, and especially UnrollingModel, the structure also needs to be adjusted.

Notably, the models simulate a steady-state simulation. This means that the water demand of the output nodes and the state of the WDS are assumed to be fixed. Conversely, pumps and tanks are continuous, meaning that their behavior and state, like the pump power throughput and the tank level, change during the day. The same holds for the demand over a day. To adjust for this, the models should be modified so that they simulate the flows and heads of the WDS considering a dynamic demand pattern. Following EPANET, the dynamic behavior arises from a series of steady-state simulations. Demands and pump states can then vary per element of this series, while the original working model is still used as is. Apart from incorporating this dynamic behavior, the models should also produce output for every time step requested. Meaning that, if a network is simulated for 24 hours with a time step length of one hour, then

the output should include 24 series. Thus, while the models based on algorithm unrolling are promising, they require multiple adaptations to be used for pump scheduling.

3

Methodology

The first step of this thesis is to adapt the models so that they can conduct continuous simulation. The next step is to extend the network with tanks and pumps and, afterward, ensure the adapted meta-model remains capable of modeling hydraulic systems using these components. More specifically, after selecting a network and determining a water demand pattern, the search space of its combined pump and tank operation schedule is explored. A dataset is created to train our meta-model on, and its performance is analyzed. After training, optimization of the network's pump schedule is performed. Finally, the extent to which pump scheduling was successful is discussed, and the reproducibility of results is explored on a set of different WDSs.

3.1. Meta-model development

3.1.1. Meta-model adaptations

Continuous simulation

To extend the capability of the steady state meta-model, we implemented an iterative approach. The difference from each iteration of the loop to the next is the input of the meta-model. There are now 24 entries, one for every hour, where previously there was one. This holds for the input demand data, for the pump configuration introduced later on (on/off), as well as the predicted pressures. Below is a graphic illustrating two iterations of this loop.

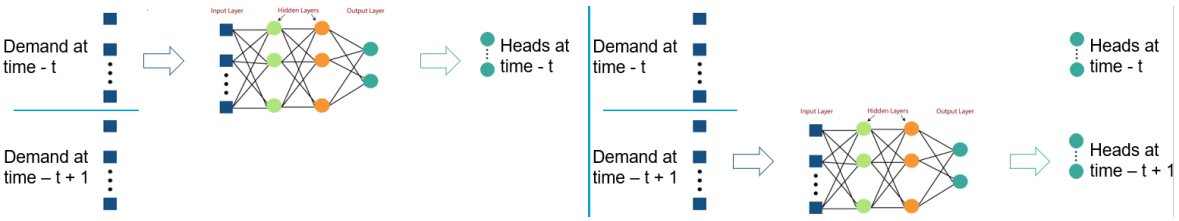


Figure 3.1: Graphic depiction of Extended Period Analysis loop (left to right)

The predicted pressures are subsequently appended to the previous ones to form a final array containing the pressure evolution of the network over the whole day. Training is also adapted to take this into account, as the loss function is now calculated over the whole day.

Tanks

Modeling the tanks is done by considering them as a special case of a junction of the network whose pressure is provided in the output data and then predicted and trained on by the meta-model. The only difference in terms of data structure is that the initial tank level is appended to the known heads in the input data. Preferably, this initial level should also vary between samples to provide training examples of how the tank levels affect the system but it was not done so.

To include tanks in equations 2.9 and 2.10 of the meta-model, the unknown nodal heads - n are increased by the number of tanks t since we want to predict their head. Simultaneously, the total number of known heads n_0, which earlier only represented the reservoirs, is also increased by t, since the initial tank levels are now known. Then, the nodal demands of the tank are set to zero.

Thus, the tank is a special case of a node with known and unknown nodal heads simultaneously. This is because the head of the tank is initially known but becomes unknown as the simulation is run.

Pumps

Pipes and pumps in a WDS are represented by the flowrate value of the water going through them. WNTR requires the flowrate of the pump to calculate the energy it has consumed. This energy is used to optimize the schedule of the pump, which is why it needs to be valid numerically.

Previously, flows were not trained for in the loss function of the meta-models. This led to the variable modeling flows (q) in the meta-models not corresponding to the calculated flows through the network by EPANET. This was changed, albeit only in the case of the pumps and not for all the pipes in the system. Making this change depends on the way algorithm unrolling is implemented.

Concisely, the unrolling of Solà Roca (2023) follows the Newton-Raphson algorithm as elaborated on in section 2.6. There, the diagonal matrix D is estimated (2.1.3) as it is of importance to modeling the flows in the network, including the flows through pumps. This matrix estimates the first derivative of the head loss for every link (pipe or pump). These are the derivatives of the values in matrix A_{11} (2.4).

The estimation of this matrix is contained in the definition of Φ in equation 2.9. We modified the D matrix definition inside $\Psi(\cdot)$ because this is where the flows through the pumps are represented in the original GGA algorithm. We opted to concatenate the pump schedule to its input and leave everything as is, to allow for the pump flow value to be estimated by the layer.

This modification translates to a simple addition of the operational state of the pumps in the input data in equations 2.9 and 2.10. The values are used by the estimator of D, which is $\Psi(\cdot)$. The q values corresponding to the pump flowrates are then appended to the output so that their loss can be calculated during training. The final equations of the meta-model are then almost identical to the original equations:

$$h^{(k+1)} = \Phi(q^{(k)}, x_s, x_d) \tag{3.1}$$

$$q^{(k+1)} = q^{(k)} - \Psi(h^{(k+1)}, x_s, x_d) \tag{3.2}$$

with $h^{(k+1)}$ and $q^{(k+1)}$ corresponding to the approximations of the heads and flows at step $k+1$, now including the heads of the tanks and the flows of the pumps. The placeholder x_s remains, standing for the static features of heads and diameters, while x_d is added, representing the dynamic time series of demands and pump schedules. Finally, $\Phi(\cdot)$ and $\Psi(\cdot)$ are modified to use the appropriate data corresponding to the predicted timestep, as depicted in Figure 3.1.

Figure 3.2 is an example of how the input and output dimensions of our adapted meta-model are structured.

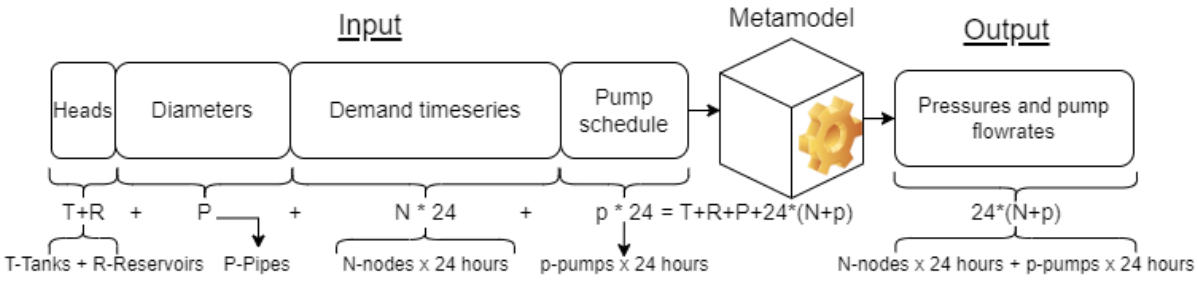


Figure 3.2: Input/Output structure of adapted meta-model

After receiving the input data x_s and x_d, the meta-model processes them according to what is required per timestep. For the first step, the diameters of the pumps (d) are used to make an informed guess (q-guess) for the flow through the pipes. This guess is then refined using the rest of the input data as the value iterates through the layers of the meta-model. This includes the reservoir heads (R), the initial tank levels (T_0), and the demands and pump schedules for the first hour ($s[0]$, $ps[0]$). The layers of the meta-model then calculate sequentially values h, q. This is a head value for every node and a flow value for every pipe. The pipe flows are called q but they do not correspond to the flow values calculated during the GGA. This is why, in the final output, only the flow values of the pumps are included. The heads and the flows are then provided back into the model, along with the demands and pump schedules of the next hour, so that the next timestep can be calculated. This process is depicted in Figure 3.3.

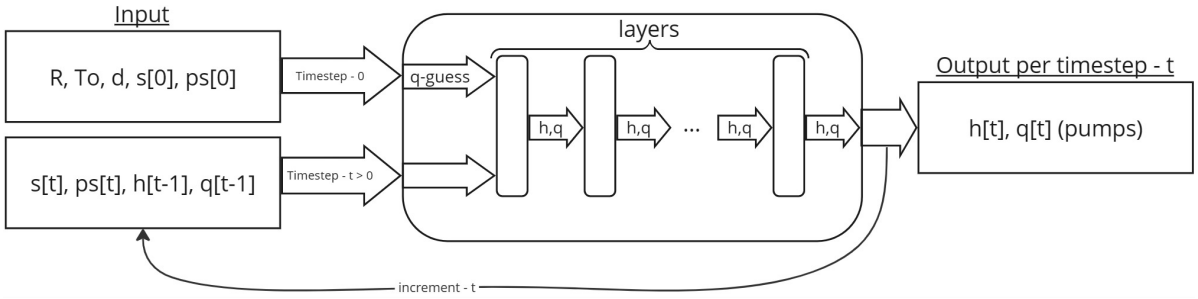


Figure 3.3: Prediction loop for the UM meta-model

3.1.2. Meta-model training & assessment

Hyper-parameter optimization

Training neural networks is a meticulous process due to the vast number of parameters that can be tuned, such as the number of hidden layers and learning rate. The increase of combinations as more parameters are considered is exponential, but simultaneously the more parameters explored, the more likely it is to reach an optimum. This is why there are tools that assist in this process of parameter tuning. One such tool is named Weights & Biases, abbreviated as WandB. Inside this cloud computing toolkit, the user can define a sweep, meaning a set of options for the parameters, and then search through them. This is how all of the models showcased were selected, using R^2 as a ranking metric.

Accuracy metrics

Two metrics were chosen to provide an overview of the model's performance. R-squared (R^2) and Mean Squared Error (MSE). The values for these metrics included in the results are only calculated using the test set.

$$R^2 = 1 - \frac{\sum (y_i - y_{i,\text{pred}})^2}{\sum (y_i - \bar{y})^2} \tag{3.3}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_{i,\text{pred}})^2 \tag{3.4}$$

where y_i is the value from EPANET, $\bar{y}$ is the mean of y_i, $y_{i,\text{pred}}$ is the predicted value, and n the number of samples.

The R^2 metric, otherwise called the coefficient of determination, is a measure of goodness-of-fit. As mentioned by Di Bucchianico (2007), R^2 can be a reliable indicator for the goodness-of-fit, as long as it is carefully interpreted. The MSE is simply an error metric indicating the difference between the predicted and simulated values.

For both metrics, the calculation is performed for the whole output, as well as per node for the final model. This is because the models constructed here predict values for every single junction, along

with the flow rate of the pumps. The errors of these values can vary depending on factors such as the placement of a node/link in the network and its function. It could be the case that junctions' heads are captured well but pumps' flowrates are not. The calculation of R^2 per node is useful because it offers a more detailed insight per component predicted.

Aside from the metrics, performance is also compared with the MLP meta-model.

Multi-Layer Perceptron The MLP used in this thesis to compare with the algorithm unrolling models is the same as the one used by Solà Roca (2023), after the adaptations of continuous simulations, tanks, and pumps. The parameters of this model were optimized using WandB, and it was found that 2 layers with 128 hidden channels provide the best R^2 for the network explored. The activation function is ReLU, as was defined in Solà Roca (2023).

Speedup metric

Speedup was also selected as a metric since it is one of the main reasons for surrogate modeling. The evaluation of speedup is two-parted. First, the speedup of the time taken to perform a hydraulic simulation with the meta-model versus EPANET is compared. Afterward, the overall time taken to optimize the pump schedule using the NSGA-II combined with the meta-model against the NSGA-II combined with EPANET is compared.

3.2. Meta-model aided optimization

3.2.1. Pump scheduling

Formulating an optimization problem involves creating an objective function. The objective function used is a combination of cost, energy, and pump switches. The goal is to minimize cost and energy use while retaining control of how often the pumps switch on and off. To solve this, the WNTR package is used to calculate the total energy used by all pumps in the network and the total cost of this energy. Optimizing only based on cost can favor energy-inefficient schedules by over-pressurizing the network at night when energy is cheaper. This is which is why energy and cost are both considered. The cost is calculated based on a given price time series shown in Figure 3.4. By incorporating both objectives, some balance will be sought between these two factors using Pareto curves. The pumps should be operated as often as possible during troughs of energy demand, while at the same time, they should be operated as least as possible while matching the water demand of the network. The water demand to be satisfied is assumed to be known, even though as stated by Walski (2001), one of the biggest hurdles in water distribution system research is accurately modeling demand.

Pumps are to be operated on a binary basis at this stage, even though in reality the power throughput of a pump can have many different speed settings or be continuous. Thus, the decision variables of this problem are the binary values of the pump operation schedule. This is an array of variables whose value is either 1 or 0 representing the on and off state respectively. The length of this array relates to the way time is discretized. If one discretizes time into very small intervals, it could be the case that for some reason the optimal schedule involves turning a pump on and off 30 times in an hour. This is not desirable, since the switch itself from these modes incurs a cost to the mechanism of the pump and it leads to a constantly shifting and unstable system. This is why it was decided to simply divide the day into 24-hour intervals and allow for any number of pump state changes between those intervals only. This leads to a binary list of length equal to 24 for every single pump in the network, representing its schedule. All of these lists combined form the decision variables of the optimization problem.

Finally, constraints in this optimization problem are represented by satisfying water demand. If there is any node in the network for which pressure is not enough to deliver water, the constraint has been violated and the pump schedule is considered invalid. Additionally, ensuring that demand is satisfied includes confirming that a certain amount of pressure is maintained for every node in the network. This latter part is called criticality analysis since nodes need to maintain pressure above a certain critical state. The minimum required pressure value of 20 psi is followed as a guideline from the *Texas Commission on Environmental Quality* (2023), but others can exist. Since WNTR was set to use metric units, it uses meters of head when determining the required pressure and 1psi $\hat{=}$ 0.7 meters of head. Therefore, we convert 20 psi to around 14 meters of head. The *Texas Commission on Environmental Quality* (2023) states that this value is the absolute minimum through which water delivery can still

occur, with 35psi being the ideal value for standard operation. The exact magnitude of the value could differ between network operators, which is why it is not of importance. Regardless of the value, we are interested in an algorithm capable of producing a satisfactory schedule.

In reality, when using the meta-model with the minimum allowed pressure value set to 14 meters of head, it struggles to produce valid schedules. The discrepancy between the meta-model and EPANET leads to schedules that tend to violate the constraint, at least by a little, when analyzed by the physical meta-model. This is why we decided to increase the minimum pressure requirement of the meta-model to try and create a safety margin for it. In this way, even with a small discrepancy, the meta-model is optimized for a schedule with stricter requirements. Different, higher values were tested, with 15 meters of head providing enough of a margin to start producing valid schedules without being too far from the optimum.

The price of the energy used to deliver pressure is also varying. Electricity price patterns indicated in Figure 3.4 are obtained from a conference paper about power grid scheduling (Parisio et al. 2011). There, real-life trends can be discerned such as the increase of energy prices during midday.

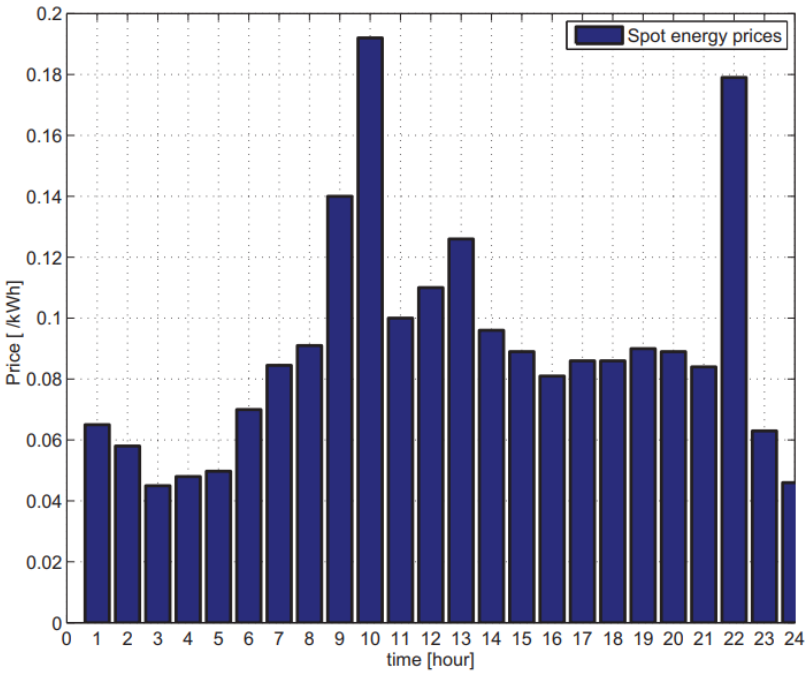


Figure 3.4: Energy prices over 24 hours Parisio et al. (2011)

WNTR already has a function where energy in kWh is multiplied by cost per kWh, but has no energy price pattern functionalities, which is why it had to be adapted for this thesis. The function where the cost of energy is calculated is adapted for the case of variable energy price. The array of the variable energy price is checked to have the same length as the array of total energy consumed by the pumps of the WDS, which is 24 due to the time discretization. Afterwards, the two arrays are multiplied and the sum is returned, equaling the total cost of operation.

Apart from the direct cost of operating the pumps, there is also an indirect cost factor associated with their maintenance. To account for this, the pump switches are modeled as an objective. We generate the optimal solutions and maintain the freedom to select a pump schedule with more pump switches, but a cheaper energy/cost combination, if that turns out to be a possibility. In this way, we obtain the best solutions for a certain amount of switches. For example, the best solution with 4 switches is obtained along with the best solution with 7, if there is one. Pump operators could then decide which schedule is more beneficial for the pump in a real-life scenario.

In total, the objective function has three parts and includes the expenditure of energy, cost, and the number of pump switches that occurred in the schedule.

$$\text{PEE}_{p,t} = \left(\frac{9.81 \cdot (\text{h}_{\text{end},p,t} - \text{h}_{\text{start},p,t}) \cdot \text{q}_{p,t}}{\text{e}_p \cdot 3600} \right) \tag{3.5}$$

$$\text{Total Energy (kWh)} = \sum_{p=1}^{P} \sum_{t=1}^{24} PEE_{p,t} \cdot \Delta t_p \tag{3.6}$$

$$\text{Total Cost (€)} = \sum_{p=1}^{P} \sum_{t=1}^{24} PEE_{p,t} \cdot \text{C}_t \cdot \Delta t_p \tag{3.7}$$

Where:

- $PEE_{p,t}$ stands for Pump Energy Expenditure for pump p at timestep t
- 3600 is the factor used to convert the calculated energy from kJ to kWh
- p indexes the pumps
- t indexes the timesteps
- $h_{end,p,t}, h_{start,p,t}$ are the hydraulic heads at the end and start nodes of pump p at timestep t
- $q_{p,t}$ is the flow rate through pump p at timestep t
- e_p is the efficiency of pump p
- Δt_p is the duration of each timestep for pump p in seconds
- P is the total number of pumps
- C_t is the cost per kilowatt-hour at timestep t

To model the constraints, a list of the minimum pressures that occurred for every node in the network is obtained. This list is then compared with our selected minimum value and, if any of the nodes' minimum pressure is lower than this value, the solution is rejected. Formally:

For junction nodes in a water network, the pressure surplus when compared to the smallest acceptable value at each node can be expressed as a list - S where:

$$\text{S (m)} = Pr_{\text{min}}(n) = \min(pr_n) - pr_{\text{req}} \quad \forall n \in \text{J} \tag{3.8}$$

Where:

- J is the set of all junction nodes in the water network
- pr_n is the set of all pressures recorded at junction node n over a specified time period
- min(pr_n) is the minimum pressure at junction node n
- pr_{req} is the minimum required pressure value, which is set to be 14m for EPANET and 15m for the meta-model
- $Pr_{\text{min}}(n)$ is the minimum pressure at junction node n minus the minimum required pressure for water delivery. This should be positive for all nodes in a valid schedule.

Finally, placing everything together as a constrained minimization problem:

$$\text{min : Total Energy (kWh), Total Cost (€), Pump Switches} \tag{3.9}$$

$$\tag{3.10}$$

$$\text{subject to : } \min(\text{S}_n) \geq 0 \quad \text{for n = 1,....,N-nodes} \tag{3.11}$$

$$\tag{3.12}$$

3.2.2. Genetic Algorithm (NSGA-II)

For implementing the GA we used a package called PyMOO (Blank et al. 2020). The abbreviation stands for Python Multi-Objective Optimization. Multi-objective involves having multiple optimization goals, which are all represented in the objective function's setup. More specifically, there are three criteria; energy, cost, and pump switches. This is what makes the problem multi-objective. The algorithm selected is NSGA-II (Non-Dominated Sorting Algorithm), which is benchmarked in Deb et al. (2002) and is deemed to be one of the best-performing optimizing algorithms for a range of multi-objective problems.

The population of the algorithm, meaning the number of samples to be evaluated per generation, is set to 100. The seed used to generate these samples is set so that the optimized schedules can be reproduced if need be. When determining the performance of the algorithm and calculating the speedup, multiple optimizations are performed using a range of pre-set seeds, and the average of their output is calculated.

The discrepancy between the heads and pump flow values predicted propagates to the energy and cost calculations of the scheduling algorithm. Therefore, after the evolutionary algorithm finishes and identifies the optimum schedules, these have to be checked with EPANET. This is done by evaluating the schedule again, this time using EPANET instead of the meta-model. The first point of interest is the head since it is often the case that the schedules produced are not capable of delivering the minimum required water demand. Then, after checking that the schedule is valid, the energy and cost values are re-calculated based on the output of EPANET. This does little to affect the overall speedup since it is only done in the end, after the schedules are found by the NSGA-II. It also ensures that the real objective function values of a schedule that is considered optimal are recorded since the final evaluation is done through EPANET.

3.3. Case Study

3.3.1. Selection of WDS

The main water distribution network explored in this thesis is the Fossolo network, which was originally constructed to be used as an example in a network design optimization paper for D'Ambrosio et al. (2008). Based on a real neighborhood located in Bologna, Italy, it simulates the average demand, which is 3000 cubic meters per day (CMD). In total, it has 37 nodes and 58 pipes.

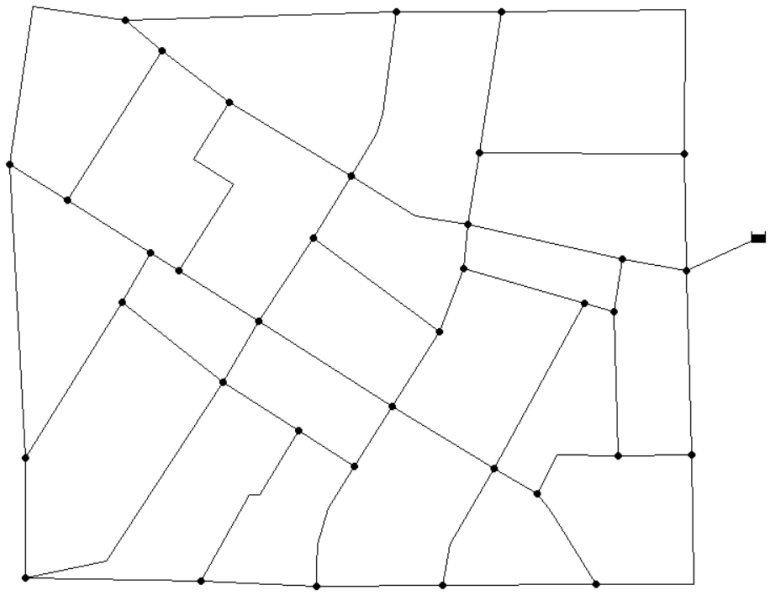


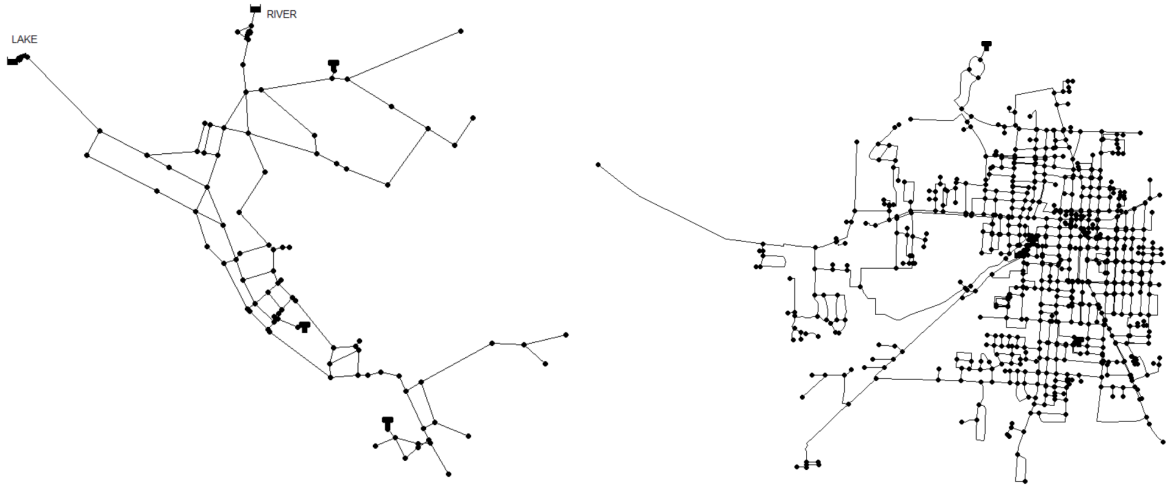
Figure 3.5: The Fossolo network, containing one reservoir and 8.4 kilometers of pipe (D'Ambrosio et al. 2008)

3.3.2. Reproducibility

Besides Fossolo, we also want to explore the performance of the meta-model in a couple of different networks to determine the extent to which the results are reproducible. Ideally, these networks selected to train on should satisfy certain characteristics. Firstly, they cannot have valves since their functionality is not implemented in the models. This is why the valves were removed from potential networks. Control rules were removed for the same reason. For most networks, pump scheduling was not possible since any closing of the pumps led to an immediate drop in the head and an inability to satisfy demands for the hour. Testing was performed after manually reducing demands to check whether they could be set low enough to be satisfied by operating the pumps partially within a day. Some were successfully adapted.

Hence, after a thorough exploration of all the small and medium-scale networks contained in the database, two were selected. These are:

- EPANET Net 3 (Net3): Example water network provided with the software manual of EPANET. It is based on a real-world water district in California. It has a total demand of 2.11 MGD (million gallons per day), one reservoir, three tanks, two pumps, 100 junctions, and 6.8 miles of pipe (Rossman et al. 2020).
- KY2: Based on a real-world water system in Kentucky. It has a total demand of 2.09 MGD, one reservoir, three tanks, one pump, 815 junctions, and 87.9 miles of pipe (Kentucky Water Resources Research Institute, University of Kentucky 2024).



(a) Net3 layout **(b)** KY2 layout

Figure 3.6: Overview of additional networks

3.3.3. Case study modifications

The first dataset is created by extending Fossolo from a steady-state simulation simulating a full day and randomizing its demands. Then, the second dataset created is concerned with the tank adaptation. Since the original Fossolo network has no tanks, a tank was manually inserted at a node close to the reservoir and the pump, as shown in Figure 3.7. The dimensions of the tank were determined manually, after randomly testing a few values and ensuring that the network's demand could be completely satisfied, at least for a few hours, using only water incoming from the tank. This led to an elevation of 100m with an initial level of 20m bringing the total height to 120m, whereas the reservoir's height was lowered to 100m to prioritize water intake from the tank. The diameter was set to 5m. The pipe connecting the tank to the network was also set with a diameter and length matching the ones of the pipe connecting the reservoir to the system.

For the final dataset, a pump is added to Fossolo after further lowering the reservoir from 100 to 40 meters to ensure that the pump, and not gravity, is the only reason for a head increase in the network. The pump's curve, similar to the tank's dimensions, is determined manually through trial and error to

ensure that the pump can provide adequate pressure to satisfy the demands of all nodes. This final version of Fossolo is depicted in Figure 3.7.

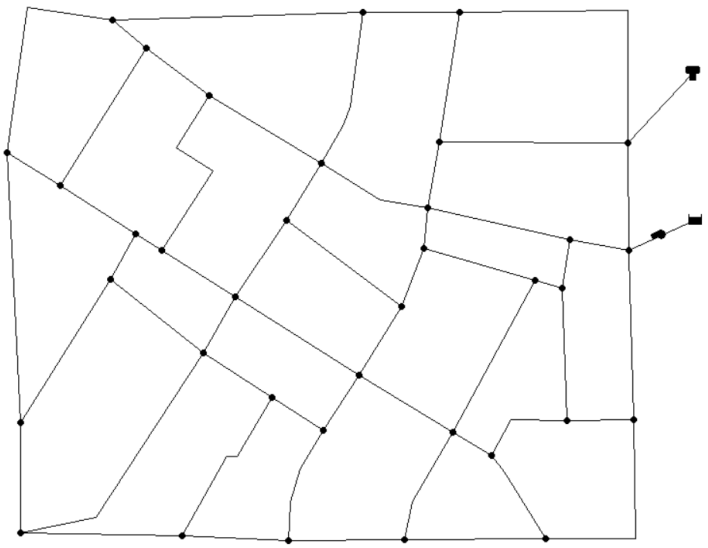


Figure 3.7: The modified Fossolo network

3.3.4. Dataset structure

To effectively train the network, the dataset should encompass a broad range of scenarios reflecting the network's real-world applications and challenges. It must be structured to accurately represent the complexities and variabilities encountered in daily operations, such as fluctuations in water demand and variations in pump schedules. In the context of ANNs, the dataset structure is vital for learning patterns and making accurate predictions. It typically includes input features that reflect the conditions under which the network operates and output labels that the model aims to predict.

Demand generation

Since the networks selected to train represent urban water distribution systems, the water demand that is used to generate the data should be representative of residential water consumption. This is usually represented daily, with an hourly interval, meaning that there are 24 entries representing the demand to be satisfied per hour. This is also how the continuous simulation of EPANET was set to be discretized. To generate such data, we use a package called PySimdeum. It is capable of "modeling and simulating residential stochastic water demand at the end-use level" (Steffelbauer et al. 2022), which is precisely what pump scheduling deals with. An example of a generated demand series is presented in Figure 3.8. Demand values in the Figure are adjusted with a random multiplier to meet network limitations and the demand time-series itself is periodically updated, ensuring diverse yet manageable demand scenarios.

The overall effect of these changes is that the daily water demand of the network is significantly decreased. This is acceptable since the point of modifying Fossolo is not to solve Fossolo's specific pump scheduling problem, which is of high, constant demands, but to show how a network of varying realistic demands can be pump scheduled with the help of meta-modeling.

After randomizing demands, the WNTR package is used to simulate the network and determine whether the run was successful. Even after reducing demands, there are still plenty of networks generated (around 10%) whose demand exceeds the capability of the water distribution system. This is the desired behavior since it means that the scenarios generated are testing the boundaries of the possible water amount to be delivered to the network. This is especially useful considering pump optimization. In a network with very low demands, any pump schedule will suffice even if the pumps are barely turned

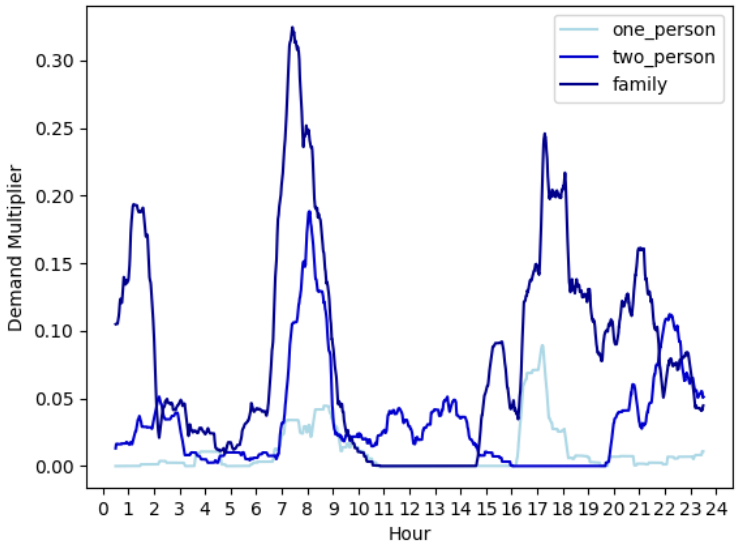


Figure 3.8: Simple example of pump demand generation scenario with three types of houses

on during the day. Optimizing a pump schedule is preferably done for a network that approaches the limit of what can be possibly delivered, without being equal to it since that would imply that the pumps are always turned on. On the other hand, if demands are too low, it could be the case that the network can fully satisfy its demands with the pumps only being turned on for one or two hours a day. In this latter case, there is little optimization to be performed other than identifying the cheapest and most energy-efficient hour.

Pump schedule generation

Regarding the pump of Fossolo, a similar process as with demands is followed during data generation. The model has to learn from examples of pump schedules so that it can make the association between the inputted binary pattern representing the pump schedule and the flow delivered to the system through the pump. This is why a random assortment of patterns is generated along with the randomization of the demands. The function generating these schedules tends to select a value of 1 with a probability of 80%. This is because in the network designed, even in a case where demands are low, the pump still needs to be set to on for most of the day to deliver a sufficient amount of water. Effectively, the schedules generated tend to have around 19-20 on values and 4-5 off ones. Since the selection is random, exceptional cases still occur if demands are met. The resulting output, where thousands of combinations of pump schedules and demand series are simulated and run on EPANET, becomes the dataset used to train our meta-models.

Training, testing and validation split

Every sample in the dataset generated represents one day of operation. To select how many samples are required to train on, we tested multiple values ranging from 100 to 10000. It was found that around 1000 samples can provide adequate variety to obtain high-performing models without complications due to the memory of the local machine. This value is dependent on the size and complexity of the WDS. Out of these values, 10% is used as a test set and another 10% as the validation set, so 100 days each, with the remaining 80%, or 800 days, being used as the training set.

4

Results

The Results & Discussion section is structured following the research questions. We start by analyzing meta-model performance through accuracy and speedup measurements. Then, we progress to a short reproducibility assessment and, finally, we assess the quality and speedup of pump scheduling.

4.1. Model performance

4.1.1. Accuracy

Continuous simulation and Tank

The adaptation of the meta-models to perform pump scheduling was a step-wise process that first required their modification for continuous simulation and, subsequently, tanks before pumps could be added. During these intermediate steps, multiple meta-models were adapted, trained, and evaluated. The Table below is a summary of the performance of these meta-models. The MLP serves as a measure of comparison for the two unrolling meta-models, BU and UM. The best-performing meta-model for the continuous and tanked versions of Fossolo is BU, while also being the one requiring the least parameters. All of the metrics are measured with the test datasets.

	Continuous Version			**Version with tank**		
	MLP	**BU**	**UM**	**MLP**	**BU**	**UM**
R^2	0.69	0.762	0.761	0.92	0.953	0.948
MSE	1.85	0.744	0.745	1.4	0.79	0.87
# of parameters	25892	12880	34417	26021	13493	35878

Table 4.1: Comparison metrics on test data between meta-models for continuous and tanked versions of Fossolo

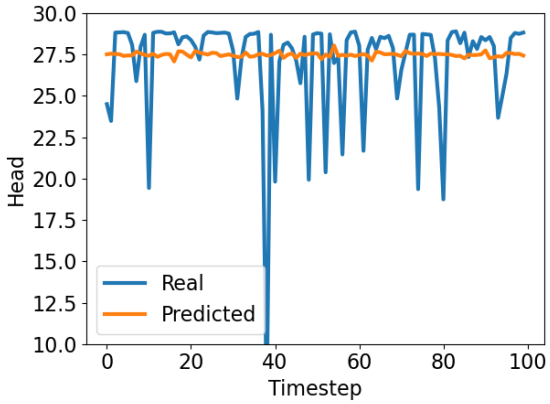
BU, the simpler version of unrolling when compared to UM, is the best-performing model. It could be the case that its decreased amount of parameters combined with the inductive bias of the meta-model enables it to train more effectively. Thus, the curse of dimensionality is a smaller problem for this meta-model compared to the other two. Still, the difference in accuracy between the two meta-models is minimal.

Pump addition

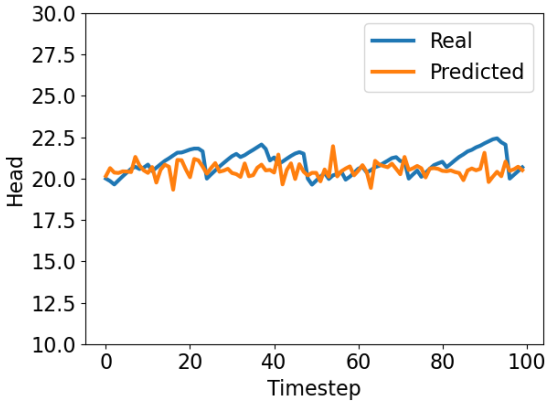
Besides predicting heads, the meta-models are now also forecasting the pump flow, as detailed in Section 3.1.1. The analysis identifies two primary sources of fluctuation in the graphical representations: variations in demand and the operational state of the pump.

Initially, we analyze the outcomes of the three selected meta-models (MLP, BU, UM). Graphs generally feature Head in meters on the y-axis, except for pump graphs where Liters-Per-Second (LPS) is the metric, and time in hours on the x-axis, with the first 96 hours (or four days) displayed. These plots compare real values from EPANET (in blue) against our meta-model predictions (in orange), with input demands and pump schedules varying daily. A metrics Table for all meta-models is provided, with a detailed focus on the UM meta-model used for pump scheduling. Later, we discuss reproducibility and pump scheduling optimization, maintaining the same color scheme for meta-model outputs and EPANET results, but adjusting axes and units as necessary. All of the results included are generated with the test datasets.

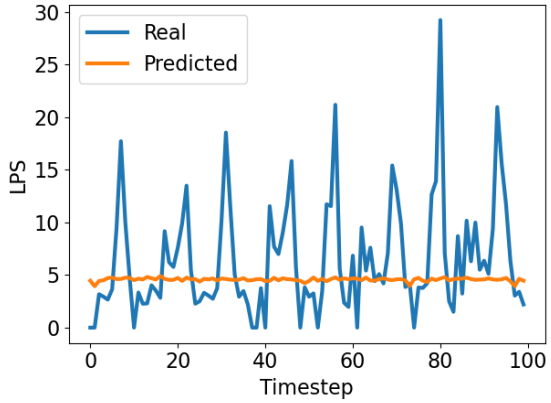
Multi-Layer Perceptron Figure 4.1 showcases the head of a random node, the tank, and the flowrate of the pump, describing the overall prediction of the network by the MLP. For figures 4.1a and 4.1c, it is clear that the meta-model is predicting a constant value, close to the average value of the network. It is incapable of capturing the fluctuations in the nodal heads and the pump flow. Figure 4.1b does exhibit some variation in the prediction but it appears to be random and does not follow the tank levels.



(a) Random - Node 2



(b) Tank - Node 37

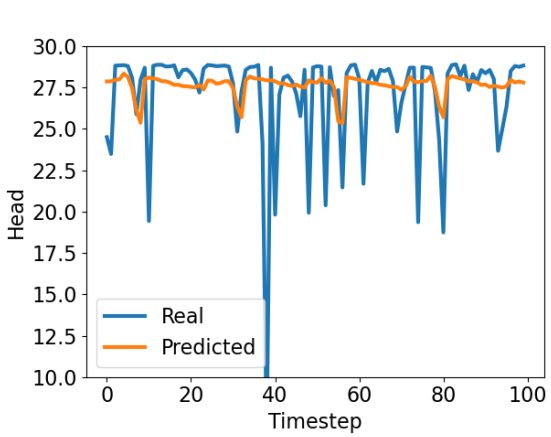


(c) Pump - Link 38

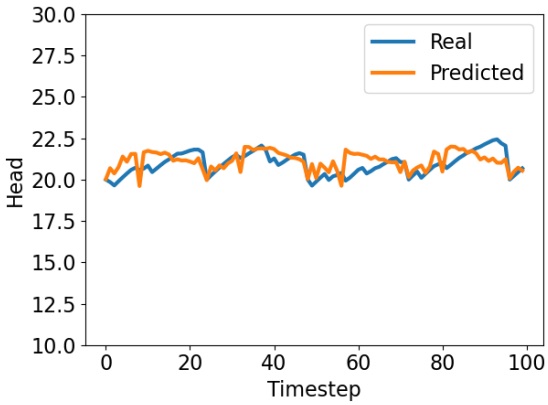
Figure 4.1: MLP predicting Fossolo network with a pump and a tank on test data (4 days)

The MLP meta-model, with an R^2 of 0.62 seen in Table 4.2, appears incapable of meta-modeling the system beyond finding a satisfying average value for its elements. The implementation of pump scheduling is infeasible with the MLP as a proposed meta-model, owing to its inability to accurately account for the variations in the head caused by the operational states of the pump. All pump schedules would, more or less, perform the same according to the MLP.

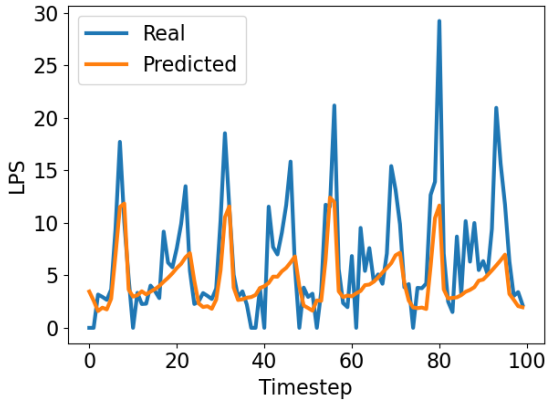
BaselineUnrolling As seen in Figure 4.2c, the BU meta-model manages to capture the pump's flow to an extent and retain some of its performance when predicting tanks but is not able to capture a generic node's head, as in Figure 4.2a. Specifically, it is unable to capture most of the nodes' head drops due to the pump schedule and barely captures peaks in demand. The tank prediction is also better than the MLP, but the tank pattern remains erratic due to the pump's operation. Drops in the pump's flow only occasionally correctly translate to drops in the node's head, and are represented too steeply in the tank's level, which varies slowly throughout the day.



(a) Random - Node 2



(b) Tank - Node 37

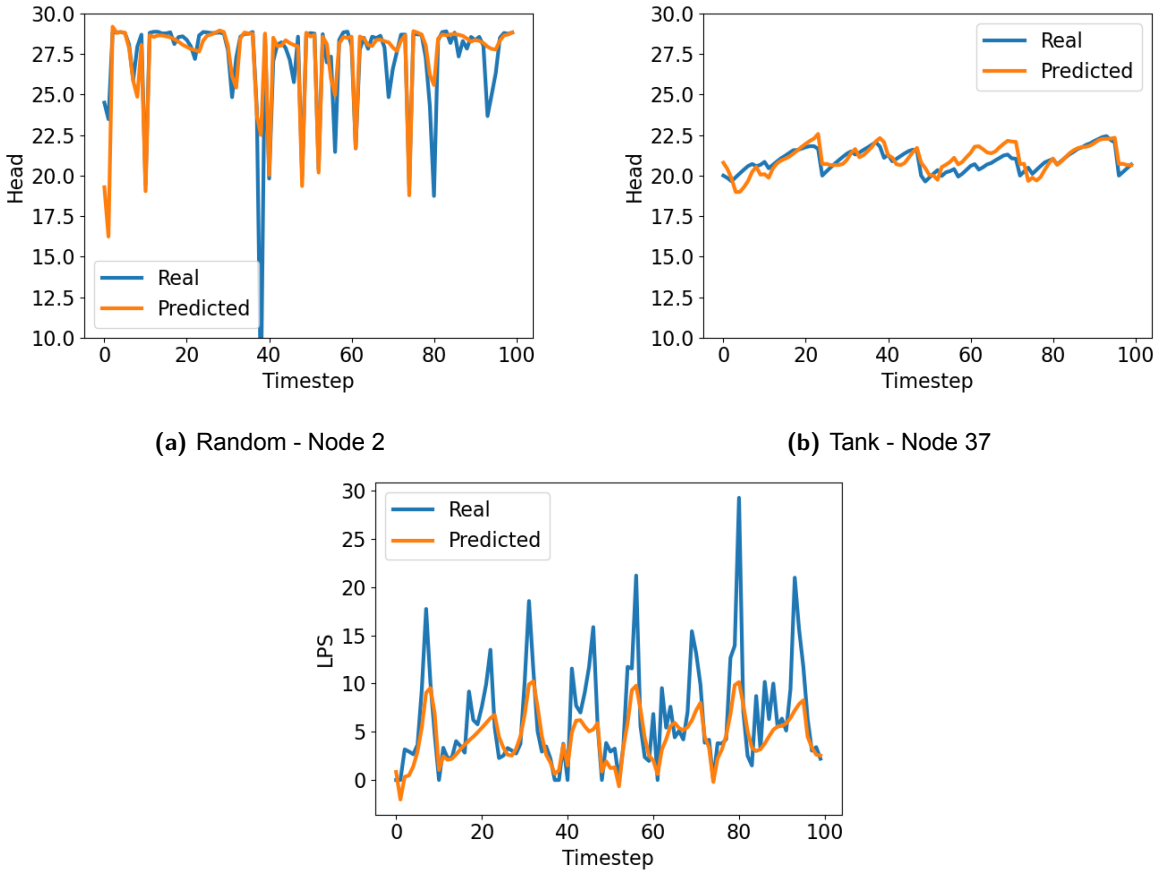


(c) Pump - Link 38

Figure 4.2: BU predicting Fossolo network with a pump and a tank on test data (4 days)

Thus, the added complexity of the pump is captured but at the cost of the previous nodes' performance. This means that, much like with the MLP, BU would be a sub-par candidate for surrogate meta-modeling. Manual examination combined with a low R^2 score (0.64) is enough to exclude it.

UnrollingModel Finally, the UM meta-model is presented.



(a) Random - Node 2

(b) Tank - Node 37

(c) Pump - Link 38

Figure 4.3: UM predicting Fossolo network with a pump and a tank on test data (4 days)

This final meta-model performs well considering that the previous UM meta-model trained on the tanked version of Fossolo was unable to capture variability in nodal heads. This meta-model captures not only the effects of the pump schedule but also of changing demand. The large drops in the head are a result of the pump being turned off but there are also small ones due to demand increase, like in node 2 around hour 20, that are captured.

	MLP	**BU**	**UM**
R^2	0.62	0.64	0.831
MSE	10.9	10.3	4.74
# of parameters	26278	13454	35278

Table 4.2: Comparison metrics on complete test dataset between meta-models for Fossolo with a pump and a tank

Examining only the metrics does not provide the full picture. Calculating R^2 on the whole dataset is an indicator of goodness-of-fit (Di Bucchianico 2007), but it can also be deceiving unless components are independently examined. In this case, the components are the nodes for which the R^2 can also be calculated per node (junction, tank, or pump) instead of the aggregated output. This is calculated on the whole of the test dataset (100 days) and plotted over the network in Figure 4.4

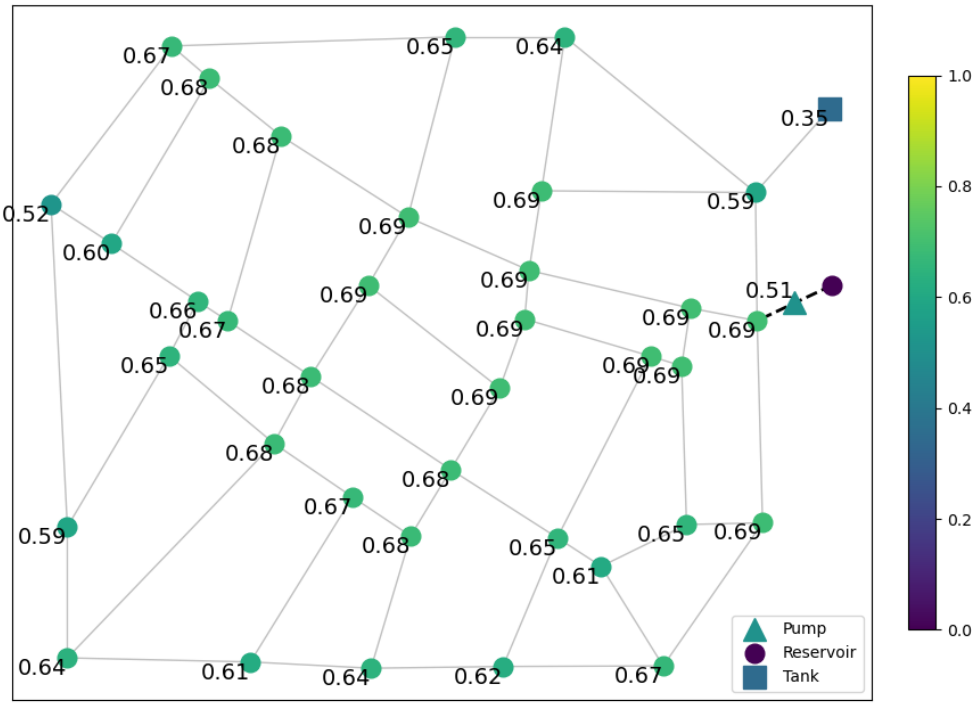


Figure 4.4: UM - R^2 metric per node measured on the complete test dataset

The UM meta-model's R^2 is equal to or above 0.5 for all nodes except for the tank. This is a sign that the meta-model manages to capture the behavior of the system as a whole, instead of simply fitting to specific elements. Overall, the UM is the best meta-model out of the three and the one with which pump scheduling is implemented.

The sudden improvement in the performance of the UM after the pump addition is in line with typical network behavior. Capturing complexity is not necessarily a linear process moving from simple to more complex predictive tasks (Abbott 1994). Nevertheless, a future investigation would be useful to understand the differences between the networks and why these lead to such different meta-model behaviors.

Error distribution

We select a day and examine the spatial distribution of errors at three specific hours—0, 1, and 10—to understand the dynamics of the system under different pump operation conditions. These hours were intentionally chosen based on the operational schedule of the pump, as outlined in Table 4.3. This schedule indicates the pump's operational status at each hour, where '1' signifies the pump is on and '0' indicates it is off.

Hour	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	...
Pump Status	1	0	1	1	1	1	1	1	1	0	1	1	1	1	1	0	...

Table 4.3: Pump operational schedule with key hours highlighted.

Hour 0 represents the initial condition of the system, showing its performance with the pump activated. Hour 1 provides insight into the system's behavior in a state of equilibrium after the pump has been

operational for a duration. Hour 10 is selected to illustrate the system's response after the pump is turned off, highlighting a period where the pressure drops, as seen in Figure 4.5. The meta-model does manage to capture the effect of the pump being off, observed as a noticeable decrease in the head, but it falls short of matching the actual magnitude of the drop. This inability to fully account for head drops significantly contributes to inaccuracies. More specifically, hours 10 and 16 show a pressure drop of the same magnitude in our meta-model while the EPANET value during hour 10 is much lower.

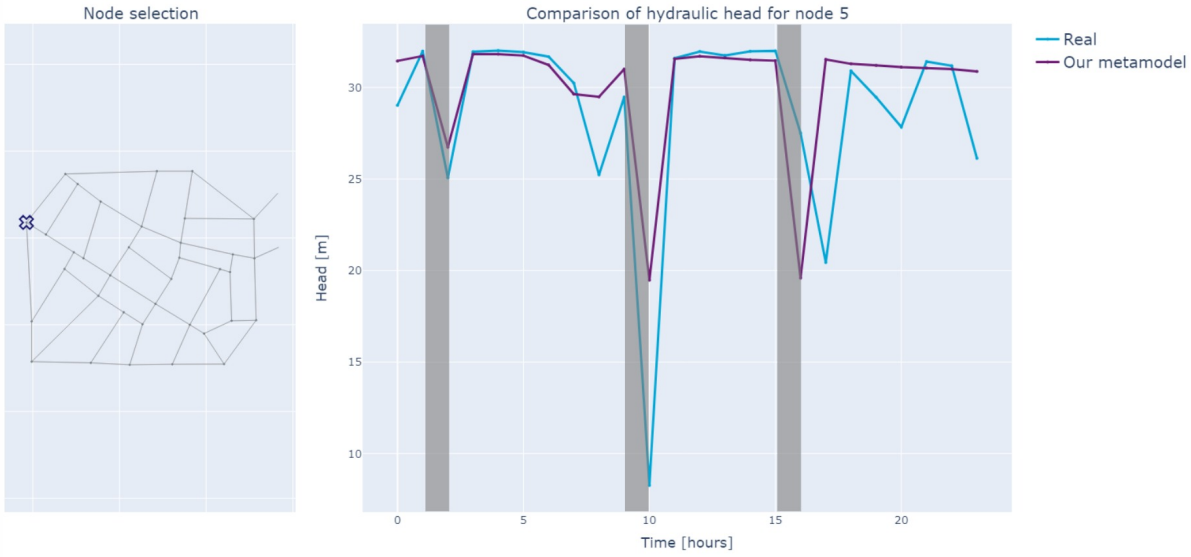


Figure 4.5: Prediction time-series for a random node at selected hours reflecting the pump's operational impact. Gray areas indicate the times when the pump is off

To assess the accuracy of the UM meta-model, the spatio-temporal error distribution is plotted in Figure 4.5, ensuring that there are no nodes with consistently high errors. Errors are visually represented through two dimensions: color and circle size. The color scale is absolute, ranging from 0 to 1 meter of head error, where a darker color signifies an error of 1m or more. The circle size is proportional to the magnitude of errors, with larger circles indicating errors that are relatively high compared to other prediction errors. Key components of the network, such as the reservoir (denoted as 0), the tank (37), and the pump (38), are highlighted in blue for clear identification.

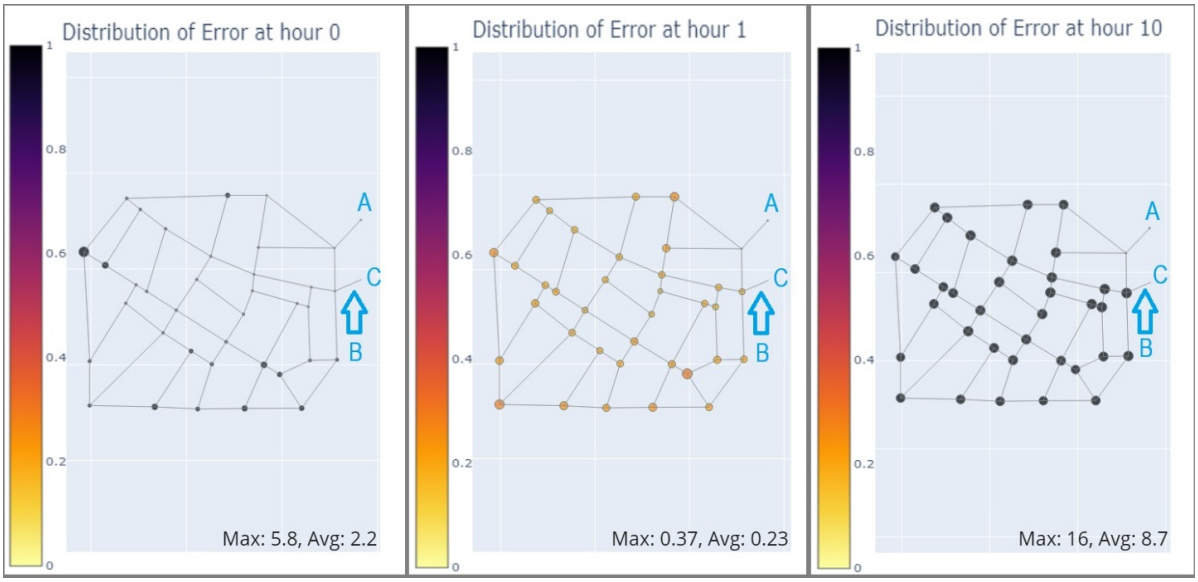


Figure 4.6: Spatially distributed error on three selected hours. Hour 0: Initial operation. Hour 1: Equilibrium state. Hour 10: Pump shutdown. Tank (A), pump (B), and reservoir (C) denoted

Figure 4.6 illustrates the system's spatial distribution error at critical hours. Initial operation (hour 0), equilibrium state (hour 1), and during pump shutdown (hour 10). These moments are pivotal for analyzing the system's response to changes in pump status, with a marked performance decline observed during off cycles. Errors during initial operation are concentrated away from the pump, tank, and reservoir but are later distributed uniformly across the network. This is both the case for the equilibrium state, as well as pump shutdown, except for the tank node and the one adjacent to it.

4.1.2. Speedup

Table 4.4 displays a comparison of the time taken by EPANET and the meta-model to simulate pump scenarios. Simulating a single pump schedule with EPANET and the meta-model takes approximately the same time. Nevertheless, the meta-model can process multiple schedules simultaneously due to parallelization. The time increase taken to process schedules in parallel is not linear, which leads to the meta-model becoming faster per schedule as the size of the evaluated schedules increases, as indicated in Table 4.4. Processing of up to 10,000 schedules is possible in about 0.68 seconds on average, effectively reducing the prediction time per scenario to 68 microseconds. Consequently, the meta-model achieves a speedup of about 220 times compared to EPANET, which does not support parallel processing. This was measured by executing every evaluation size 10 times for the meta-model and calculating the average. As for EPANET, 100 simulations were executed once with a single scenario, since parallelization is not possible.

	Average computation time per scenario (s)		Speedup
Scenarios evaluated	EPANET	meta-model	EPANET/meta-model
1	0.015 ± 0.007	0.012 ± 0.02	1.25
10	0.015 ± 0.007	0.0017 ± $5 * 10^{-4}$	8.8
100	0.015 ± 0.007	0.00034 ± $8 * 10^{-5}$	44
1000	0.015 ± 0.007	0.0001 ± $35 * 10^{-6}$	150
10000	0.015 ± 0.007	0.000068 ± $12 * 10^{-6}$	220

Table 4.4: Average speedup of meta-model due to parallelization

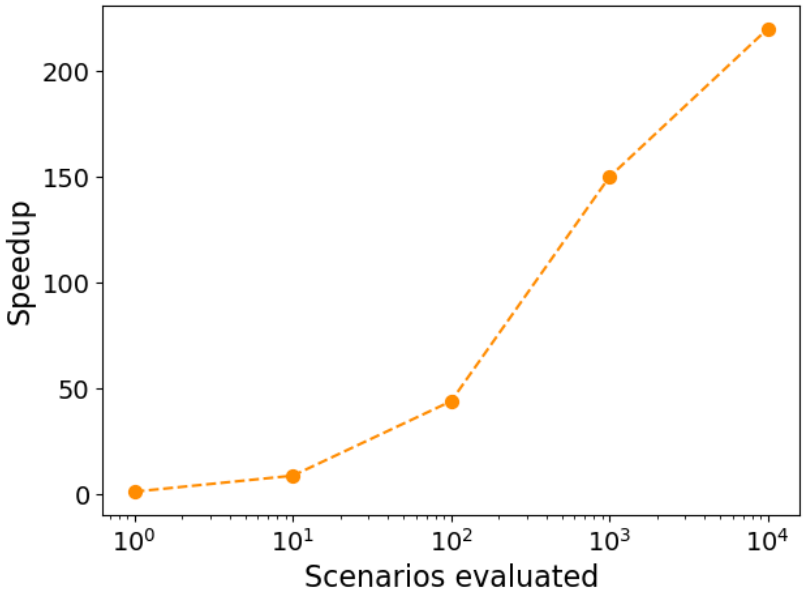
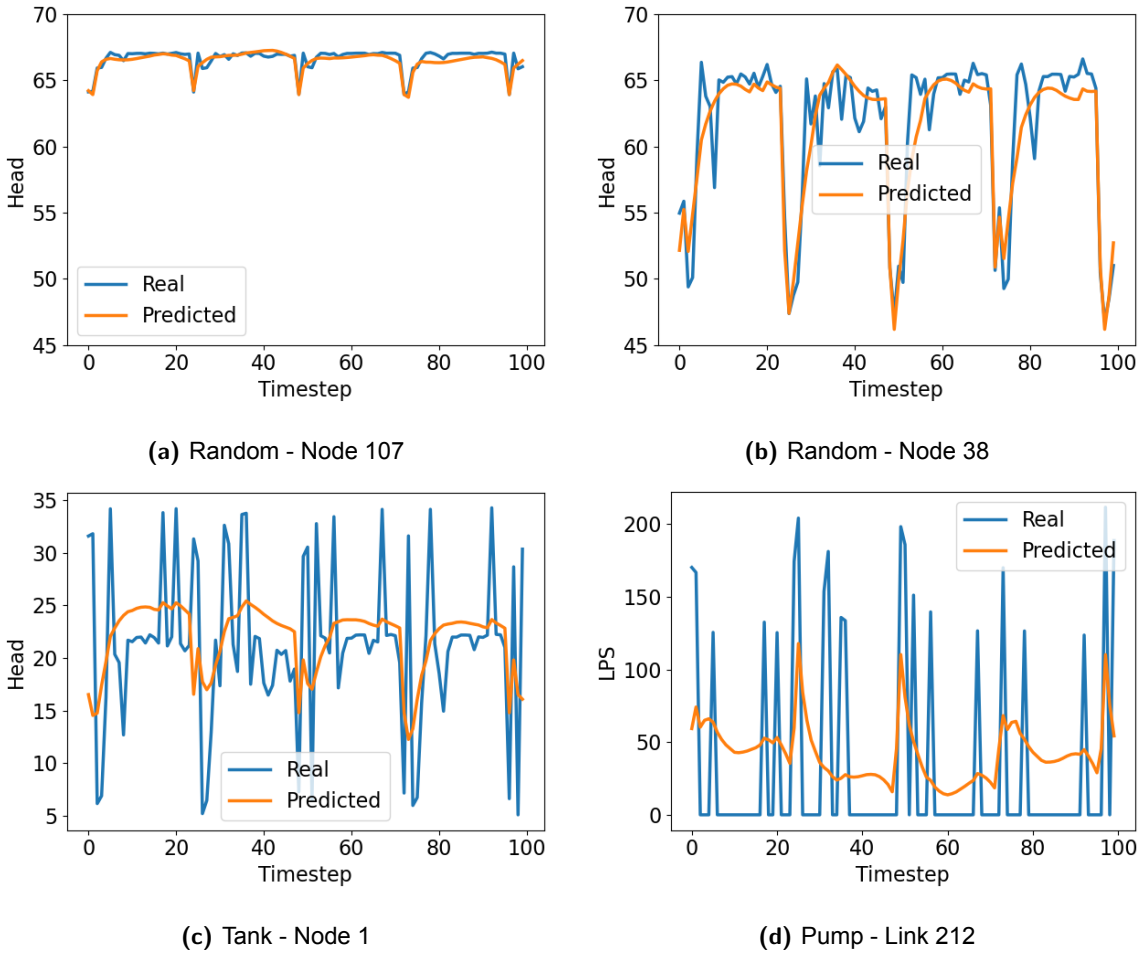


Figure 4.7: Graph indicating speedup obtained using the meta-model

Therefore, with low amounts of scenarios evaluated the speedup is marginal. The more one increases the scenarios to be evaluated, the higher the possible speedup attained, as shown in Figure 4.7. Simultaneous evaluation of 10^4 was the highest amount that could be checked due to the memory limitations of the local machine. In the case of a supercomputer, this number could be further increased leading to, possibly, even higher speedup values.

4.2. Extended set of networks and reproducibility

It would be ideal if a meta-model could be an accurate predictor for any network after training. This would mean that, while not generalizable, the process is reproducible to other types of networks. To test this, different networks are examined. Namely, Net3 and KY2 are introduced in 3.3.1. Hyper-parameter optimization was performed using the most accurate meta-model, UnrollingModel. Both of the networks are larger than Fossolo in terms of node and pump count as well as water delivery. Analysis on other nodes is included in appendix A.4, and A.5.



(a) Random - Node 107

(b) Random - Node 38

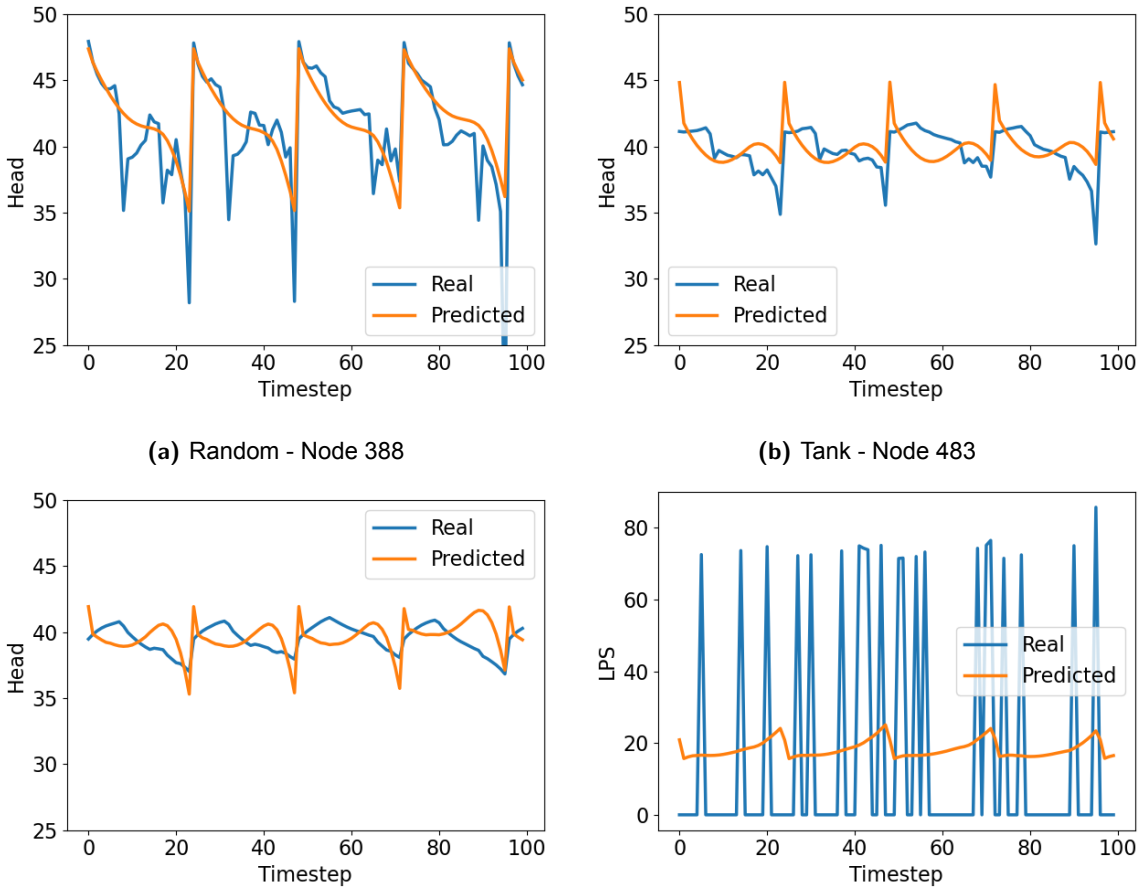
(c) Tank - Node 1

(d) Pump - Link 212

Figure 4.8: UM metamodel predictions for selected locations in Net3 on test data (4 days)

The 4 sub-figures in 4.8 give an overview of the accuracy for the Net3 meta-model. The only prediction that can be considered accurate is that of the first node, in sub-figure 4.8a. The second sub-figure 4.8b resembles more closely the typical prediction for most nodes, showing that variability is captured well. The R^2 for most of these nodes is around 0.8, which is a good fit. At the same time, the meta-model is not able to fit drops in the head due to pump shutdown. Additionally, and even more detrimental to performing pump scheduling, the tank and the pump fits are poor, with negative R^2 values indicating no correlation. The tank behavior, and specifically the sudden increases and decreases in its head caused by the pump, are not captured by the meta-model. Similarly, the predictions of the pump are not able to make the connection between the inputted pump schedule and the pump's flowrate, beyond an increase close at the beginning of each day.

Hwang et al. (2017) describe Net3 as a transmission-dense network, meaning it has a high concentration of large pipes designed for moving water over long distances. On the other hand, KY2 is labeled as distribution-dense, indicating it has many smaller pipes focused on delivering water directly to users within a localized area. KY2 has ten times as many pipes and consumer nodes. Looking at the overview of the network structures in figures A.12 and A.14 makes it clear that going from Fossolo to Net3, to KY2 we are observing networks of increasing complexity.



(a) Random - Node 388

(b) Tank - Node 483

(c) Next to Pump - Node 812

(d) Pump - Link 813

Figure 4.9: UM metamodel predictions for selected locations in KY2 on test data (4 days)

	Net3-UM	KY2-UM
R^2	0.78	0.87
MSE	698	8.7
# of parameters	180k	8.8m

Table 4.5: Metrics for additional networks calculated on the complete test dataset

Figure 4.9 is similar to the one of Net3, only with the fits being worse, when examined manually. Analogously, standard nodes are captured up to an extent, but tanks, pumps, and nodes next to them are not. In the case of networks much more complex than Fossolo, the UnrollingModel struggles to capture this complexity using effectively the same logic with just an increased amount of parameters. In Table 4.5, the exponential increase of parameters, which was also included in Table 2.1.

Despite the inability to fit these networks, there are some indications that UM could eventually manage to capture their behavior. Most notably, the nodal predictions remain relatively close to their real values and the tank (4.8c) and pump (4.8d) nodes of Net3 are poorly fit but there are small indications that they can be predicted accurately. For instance, around hour 20 of day 1, there are two bumps for both the tank and the pump. With more rigorous training and a larger dataset construction, the curve fit could resemble the accuracy of Fossolo's results.

4.3. Optimization of pump schedule

The UM method yielded a highly accurate meta-model for predicting water network behavior, achieving an R^2 rating above 0.8 for the Fossolo network. To implement pump scheduling, the model utilizes the WNTR framework alongside PyMOO and the NSGA-II algorithm. This approach involves setting up an objective function, as outlined in the methodology section 3.2.1, to guide the selection and evolution of pump schedules. These schedules are then assessed using EPANET and the best are selected, culminating in the generation of optimized pump schedules for the Fossolo network.

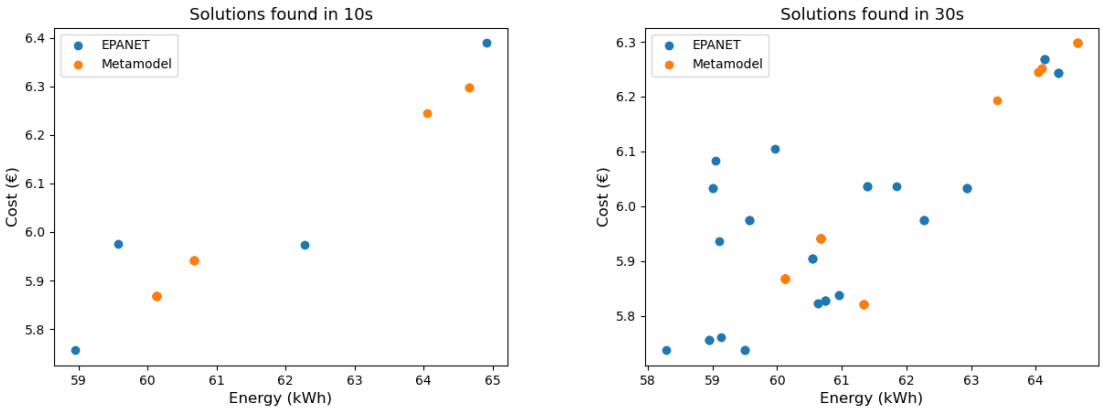
4.3.1. Quality of solutions

In this analysis, we select a sample day from the test dataset and use the meta-model UM to generate potential pump schedules for Fossolo. Out of 620 schedules generated in total, 474 were invalid while 144 were valid. The ratio of valid to invalid schedules is 0.3, or around one out of three. This means that out of every four successful schedules produced by the meta-model, only one meets the criteria set by EPANET. This is caused by the meta-model's inability to accurately predict steep decreases in water pressure, leading to overestimated pressure levels. This overestimation results in nodes being incorrectly considered as having met their water demands. This effect was observed in the meta-model accuracy section 4.1.1, where in Figures 4.3a and 4.5 it was shown that drops in the heads of nodes are not fully captured.

The meta-model's response surface tends to be smoother compared to that of EPANET, meaning it often fails to detect sharp decreases in pressure or quick increases in flow rate. Consequently, the meta-model is more likely to deem a pump schedule as successfully fulfilling water demands, despite potential inaccuracies. Furthermore, the model's difficulty in accurately capturing peaks in the pump's flow rate (as shown in4.3c) results in a consistent underestimation of both cost and energy usage. However, this is considered a minor problem because the primary concern is whether the model can correctly identify trends in cost and energy efficiency between different schedules, rather than pinpointing precise values. In other words, as long as the meta-model can reliably distinguish one schedule as being more cost-effective or energy-efficient than another, the exact degree of underestimation is less critical. This is because EPANET will ultimately perform the final assessment, ensuring the real values for both objectives are shown.

To compare the quality of the solutions produced, the two available optimization pipelines, EPANET with GA versus the meta-model with GA, are given the same computational resources by limiting the time they are allowed to produce solutions in. The time limits selected were 1, 3, 10, and 30 seconds. The small magnitude of time reflects the fact that the network has a moderate size compared to real-life water networks.

When it comes to the 1 and 3-second tests, the meta-model outperformed EPANET, which was unable to find any solutions in this amount of time. As for the 10 and 30 seconds, EPANET was the clear winner, coming up with better-performing solutions considering both cost and energy.



(a) Solutions generated in 10 seconds

(b) Solutions generated in 30 seconds

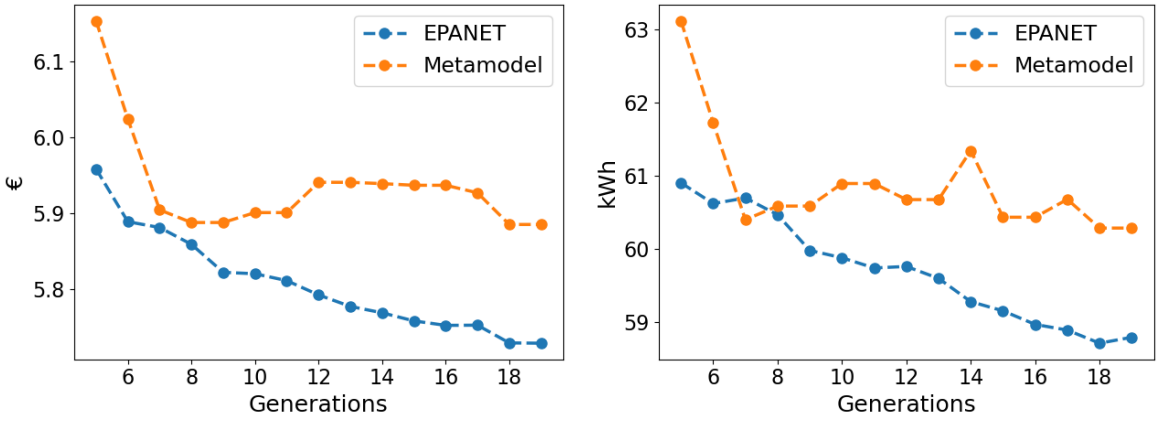
Figure 4.10: Cost over energy for timed experiments of pump schedule optimization for the Fossolo network

In this time-restricted setting, EPANET discovered slightly better solutions than the meta-model. When comparing the solutions generated in 10s, the meta-model's optimum has a higher consumption of cost by around 0.1€ and energy by 1.1kWh for a daily schedule. For both values, this is equal to around a 2% reduction in efficiency. The magnitude of this percentage is deceptive since in a real-life system, over a long period, the difference could accumulate and result in large quantities of wasted money and energy. This means that without any time or computational resource restrictions EPANET should be preferred to perform pump scheduling. In cases of time limitations, the usefulness of such a meta-model becomes apparent.

For the 1 and 3-second scenarios, EPANET could not find any schedules to match against, but our algorithm with the meta-model did manage to find valid solutions. These time differences seem minor, but in a large real-life network, they could be significant enough to make using the original meta-model impractical due to slow convergence. To provide an example, our optimization pipeline assumes that water demand is known a priori. This is not the case in real life, where short-term demand forecasting, what is preferred from an operations management perspective (Donkor et al. 2014), can have horizons as short as the next day (Namdari et al. 2023). This becomes even more prevalent as research is revealing the connection between water demand and weather (Bakker et al. 2014), for which a shorter horizon also translates to a more accurate forecast. Effectively, for an accurate short-demand forecast the time available to determine a pump schedule is 24 hours. In such an example, and the case of a colossal network, it could be that the physical meta-model simply never manages to converge to a solution within this time criterion. One would then be more likely to opt for a sub-optimal scenario generated by the meta-model, instead of a scenario that is constantly lagging behind the real-world conditions.

4.3.2. Meta-model speedup

To evaluate the speedup when optimizing the pump schedule using the meta-model, a different experiment is performed. The initial populations of 100 solutions for the GA are randomized multiple times but set to be the same among the two optimization pipelines. 620 schedules are generated by conducting 5 separate runs of the genetic algorithm, each initiated with a unique seed. These runs span from 5 to 20 generations, with the results recorded at the end of each generation. The performance of the schedules found by the meta-model is evaluated using EPANET at the end of every generation to determine, first, whether they are valid schedules and, then, their cost and energy consumption. An average of the cost and energy expenditure of the solutions is then calculated and plotted along the number of generations that the pipelines are run for. The solutions are comparable, with the meta-model's being slightly worse than EPANET's, excepting energy consumption for the seventh generation. However, after the eighth generation, we observe a plateau and subsequent deterioration in the meta-model's solution quality, in contrast to the continuous improvements observed using EPANET.



(a) Cost of the optimal schedules

(b) Energy consumption of the optimal schedules

Figure 4.11: Pump scheduling objective function results. EPANET completes in 34' minutes vs. the meta-model in 2'

If the energy and cost expenditure determined using the meta-model's output were included, and not EPANET's evaluation, a similar constant improvement as with EPANET would be observed in the form of the cost/energy lines steadily declining for the meta-model. The divergence stems from a gradual misalignment between the optima identified by the meta-model and those of EPANET. Solutions that appear optimal to the meta-model become less effective according to EPANET's evaluation beyond the eighth generation. This misalignment suggests a potential limitation in the meta-model's ability to accurately predict the physical meta-model's optimal solutions over extended iterations. Improving the meta-model through enhanced meta-model construction and dataset creation may reduce, but not eliminate, this deviation. Recognizing this limitation is crucial for accurately interpreting the meta-model's long-term performance and reliability.

Running the GA with EPANET takes 34 minutes while running it with the meta-model takes 2. The factor of this speedup is 17, notably smaller than 220, the highest possible speedup attained through the meta-model. It follows that the speedup mentioned in section 4.1.2 is not to be confused with the overall speedup of the pump scheduling algorithm. The calculation of the heads and the flows of the network, which is how the factor of 220 was determined, is not the only task performed during each generation of the optimization algorithm. The values produced, either through prediction or simulation, are used to calculate the energy and cost expenditure of the specific scenario, which is another computationally costly task.

The real speedup is the total time taken to run the algorithm. This is partially limited by how WNTR calculates energy and cost by using the pump flow rates and the heads. These package functions were not optimized for performance, but readability, leading to their bottle-necking performance. This means that while there is a potential to vectorize them and obtain even higher speedups, it was opted to use them as-is, to avoid introducing another point of possible errors in the calculation pipeline. The rest of the function executions were vectorized and optimized as much as possible.

In addition, 10 single-generation optimizations were performed with an initial population of 10^4, instead of 100 used during optimization, to take full advantage of parallelization. The average time required for those was then calculated for both EPANET with the GA and the meta-model with the GA. This led to an effective speedup of 20.

Only 20 times faster appears meager in comparison to the 200-fold speedup of the meta-model. Nevertheless, in a real-life scenario, it could be more than enough to produce pump schedules on time. For example, what previously took 7 days to calculate would now take around 8.5 hours.

5

Conclusion

In this thesis, we have explored and combined two distinct areas of research. These are surrogate modeling by algorithm unrolling and pump scheduling of water distribution systems using a Genetic Algorithm. The two areas have been researched extensively over the years but, to the best of our knowledge, have never been combined before. We will now state the conclusions of the research from the scope of the initial research questions posed.

5.1. What is the performance of a meta-model of EPANET created via algorithm unrolling in terms of accuracy and speed?

The first half of the research was focused on creating the best possible metamodel of EPANET for the network of Fossolo, a network representing a neighborhood in Italy. During this part, we focused on improving and extending the existing surrogate models created by Solà Roca (2023). Initially, these meta-models were designed for simple water systems and were limited to modeling reservoirs and standard nodes. We improved the functionality of three models by incorporating tanks and pumps. The UnrollingModel(UM), which was the most complex among them, showed superior performance. It was the only model that could accurately represent the behavior of both tanks and pumps to a certain degree. This was evident from its overall R2 value exceeding 0.8, and R2 values over 0.5 for individual nodes, indicating a strong match. Therefore, UM was chosen for pump scheduling tasks. Moreover, this meta-model delivers predictions over 200 times faster than traditional physical simulations.

Ideally, the process of unrolling should adhere more closely to the original algorithm (GGA) implemented in EPANET. Apart from the possibility of accuracy improvements, it is likely that doing so would help ensure that more variables follow their GGA-counterparts, such as the flows through the pipes, which are currently intermediary variables whose value does not relate to the actual flows. In this case, the links between the nodes of the WDS would be known and further enhancements could be made, like linking the tank to its incoming and outgoing water volume and implementing measures to mitigate overflow or emptying. Similar advancements could be made for the pumps, incorporating the equations of EPANET handling pumps in the meta-model's architecture.

5.2. How replicable is this procedure across different water distribution systems?

Reproducibility was explored by training the model on two different networks; *Net3* and *KY2*. The model was unsuccessful at predicting EPANET's output accurately enough to perform pump scheduling for either network. They are both structurally more complex networks in comparison to Fossolo. Additionally, no correlation between performance and network complexity is observed as overall performance does not degrade in network *KY2*, which has an increased number of nodes and pipes compared to Net3. This is in line with the findings of Solà Roca (2023). It is highly unlikely that valid schedules could be produced using these models and, it is definite, that the energy and cost calculations based on the

pump flowrates would have little to no correlation to the actual values.

It cannot be concluded whether the process is reproducible as there are small indications in the form of peaks and drops at the correct spots, suggesting that improving the training process could cause the model to be capable of fitting to the networks. The main issue that prevented us from performing this training was model instability. A model based on algorithm unrolling can be made more complex by increasing the amount of its layers. The model was unstable beyond a certain small amount of layers when compared with the iterations of EPANET. Exploding gradients prevented the model from converging to the network. Therefore, instead of merely recommending the construction of a more thorough training pipeline with more data samples and a higher amount of parameters/layers, we recommend future researchers revisit the basics of the model. Understanding why algorithm unrolling works so well and then breaks beyond a certain number of layers would be key to improving the models. Implementing a stricter version based on the algorithm unrolling paradigm is another possible approach. Reconstructing a model more closely following the original EPANET equations could lead to models capable of fitting larger and more complex networks.

5.3. To what extent can the meta-model be reliably used to optimize pump schedules?

During the second half of the research, an optimization pipeline using the Genetic Algorithm NSGA-II was set up. This pipeline is two-parted since it can either use the WNTR package to optimize the network's pump schedule or our meta-model. The objectives of the optimization were cost and energy. At this stage, a speedup of upwards of 20 times was observed when using the meta-model instead of the physical-based method for optimization. One magnitude less than what is empirically proven to be possible, but still fast enough to make a considerable difference during optimization. It was also observed that only around 30% of schedules evolved using the meta-model were valid schedules when served back to EPANET. It is theorized that vectorization of the energy calculations of WNTR and better model creation could improve the speedup attained and the ratio of valid schedules respectively.

Finally, a comparison between the two approaches was performed. Placing the optimization results of each approach side-by-side, proves that the physical-based method is preferred when it comes to generating the most optimal pump schedule. What the physical-based method cannot do is generate a schedule in a short amount of time. This is where meta-modeling through algorithm unrolling holds the potential to become a valuable tool for operation managers of water systems.

5.4. Limitations & broader application

One of the primary limitations observed in this thesis relates to the meta-model's ability to accurately predict and optimize pump schedules across different water distribution systems. While the model showed promising results for the Fossolo network, its applicability to more complex networks like Net3 and KY2 was limited, indicating a potential constraint in the model's reproducibility. This limitation is further compounded by the meta-model's difficulty in capturing intricate behaviors of the water distribution system, such as steep drops in pressure or accurate flow rates through pumps, which are crucial for precise pump scheduling. Additionally, the instability of the model with an increased number of layers, as highlighted by the phenomenon of exploding gradients, presents a significant challenge to scaling the model for larger and more complex networks. These limitations underscore the need for further research to enhance the model's stability, accuracy, and reproducibility.

Despite the limitations identified, this thesis contributes valuable insights into the optimization of pump scheduling, with particular emphasis on the distinction between traditional methods like EPANET and the innovative application of meta-models. The process of optimization using EPANET and a GA serves as the groundwork upon which the meta-model, combined with a GA, offers an alternative approach when dealing with large water distribution networks or when faced with constraints in computational resources. Specifically, the meta-model presents an advantageous option for rapid pump scheduling in scenarios characterized by sudden changes or short-term demand forecasting, where EPANET's detailed simulations may not be as feasible due to time or computational limitations.

This distinction emphasizes the complementary nature of the two approaches: EPANET remains the

preferred method for achieving the most accurate and optimal pump schedules, particularly in networks of a small to intermediate size. However, in situations where the network's complexity has large proportions, or when immediate scheduling decisions are required without extensive computational processing, the meta-model emerges as a practical tool. This strategic utilization of the meta-model, leveraging its speed and efficiency, can aid operation managers and researchers in navigating the intricate balance between satisfying operational demands and minimizing resource expenditure.

6

Future Research Directions

Throughout this thesis, several important areas have been noted that deserve extra attention in an added section. These areas could provide a starting point for future studies aiming to build on this work and overcome the limitations we have found.

6.1. Model architecture

The unrolling process in modeling usually aims to closely follow the original equations. However, the models discussed encounter challenges in directly copying EPANET's equations because they do not use the network's graph structure, such as how nodes and edges are connected. Instead, the meta-model learns from the relationships within the network data. Solà Roca (2023) tried using Graph Neural Networks to mirror EPANET's structure, but they could not match the performance of algorithm unrolling models in accuracy or speed.

Building upon the foundational unrolling approach, an ideal enhancement would incorporate the tank and pump equations as detailed in Chapter 12 of Rossman et al. (2020), after figuring out how to include the network's graph structure in an efficient manner inside the meta-models. Such an integration would provide the meta-model with a more sophisticated architectural framework that mirrors EPANET's global gradient algorithm more closely. For tanks, this would involve accurately modeling the dynamics of volume and flow, including the capability to reflect the tank's filling and draining processes. For pumps, the model would encapsulate the complex interplay between the pump's curve, the flowrate, and the pressure, fundamental to determining the pump's effect on the rest of the system. A meta-model equipped with these specific functionalities would not only extend the learning process to more complex aspects of the water network but also potentially offer a truer representation of the physical processes at play, thereby enhancing both the predictive power and utility of the model in practical applications.

6.2. Model instability

While refining the BaselineUnrolling and UnrollingModel based on algorithm unrolling, model instability due to complexity, such as increased learning rates or layer counts, frequently led to exploding gradients during backpropagation, a problem not fully understood. Solà Roca (2023) managed this by reducing layers and parameters, but the persistent nature of this instability suggests it requires more investigation.

Efforts to address exploding gradients included trying various optimization strategies, with Adam optimization being chosen (Kingma et al. 2017), despite its inability to fully prevent the issue. Subsequently, gradient clipping was employed, which limits the gradient vector's norm to a specified threshold, preventing gradient explosion without altering weights (J. Zhang et al. 2019). Yet, setting this threshold proved challenging; a value too low hindered learning, and too high did not curb the gradients. This ongoing instability, despite applying multiple strategies, indicates a potential flaw either in the application of the strategies, the models' structure, or the training process. Future work should focus on refining

the training approach and the models to enhance stability and performance.

6.3. Greenhouse Gas Emissions

In this thesis, energy usage is directly accounted for when performing optimization. As is highlighted by Lee et al. (2023), the emission of fossil fuels is the main contributor to human-caused climate change. Hence, for the duration of this thesis, energy has indirectly been implying greenhouse gases (GHG), although this is not necessarily the case. GHGs, or fossil fuels, are the main energy source for most energy grids worldwide IEA (2023). Still, there are other energy sources such as renewables providing an increasing percentage of this energy. Solar and wind energy are becoming more prevalent in developed countries seeking to reduce their carbon footprint. The percentage of the total energy production of these sources is closely correlated with the hour of the day. At noon is when there is usually the most sun and wind so the percentage of the total energy provided by these sources is the highest. Hence, optimization of pump scheduling could directly take into account the percentage of energy generated through GHGs and try to optimize based on this value. In this case, overnight operation would be disfavored compared to mid-day. For the Fossolo network, which is located in Italy, the Terna energy grid operator directly provides this data (Terna 2024).

6.4. Time discretization

Another related point of discussion is the fidelity through which time is discretized. Once again, for the sake of limiting the combinatorial explosion of the possible solutions/schedules of the pumps, the day was divided into 24 hourly intervals. This was deemed simple enough for schedule generation, while detailed enough to capture the variability of demands and electricity costs, which were preset hourly. There is an added advantage to this, which is that there is a limit to the possible pump switches per day (12), too many of which are harmful to the pumps' operation. In real life, pump operators appreciate flexibility when generating a schedule since demands can vary widely within any hour. Thus, it would be ideal to opt for intervals of higher fidelity, like 30 minutes or quarterly Lansey et al. (1994).

6.5. Ensemble Forecasting

In combination with the rise of short-term demand forecasting, the use of probabilistic forecasting is expanding in research, offering not just a single forecast but a range of possible outcomes within an uncertainty interval (Gagliardi et al. (2017), Hutton et al. (2015)). This approach, known as ensemble forecasting, has been applied in weather prediction for over a decade, as stated by Leutbecher et al. (2008). The connection between meta-modeling and pump scheduling lies in the ability of a rapid meta-model to investigate a vast array of future scenarios through ensemble forecasting, thereby enabling the pre-generation of multiple pump schedules. This approach allows for a more informed selection as the prediction horizon narrows and real-time demand measurements indicate which scenario is unfolding.

References

Abbott, L. F. (Aug. 1994). “Decoding neuronal firing and modelling neural networks”. en. In: *Quarterly Reviews of Biophysics* 27.3, pp. 291–331. ISSN: 0033-5835, 1469-8994. DOI: 10.1017/S0033583500003024. URL: https://www.cambridge.org/core/product/identifier/S0033583500003024/type/journal_article (visited on 02/13/2024).

Alizadeh, Reza, Janet K. Allen, and Farrokh Mistree (July 2020). “Managing computational complexity using surrogate models: a critical review”. en. In: *Research in Engineering Design* 31.3, pp. 275–298. ISSN: 0934-9839, 1435-6066. DOI: 10.1007/s00163-020-00336-7. URL: http://link.springer.com/10.1007/s00163-020-00336-7 (visited on 06/07/2023).

Bakker, M., H. van Duist, K. van Schagen, J. Vreeburg, and L. Rietveld (Jan. 2014). “Improving the Performance of Water Demand Forecasting Models by Using Weather Input”. In: *Procedia Engineering*. 12th International Conference on Computing and Control for the Water Industry, CCWI2013 70, pp. 93–102. ISSN: 1877-7058. DOI: 10.1016/j.proeng.2014.02.012. URL: https://www.sciencedirect.com/science/article/pii/S1877705814000149 (visited on 02/06/2024).

Balas, Valentina Emilia (July 2009). “Multilayer perceptron and neural networks”. In: *Wseas Transactions on Circuits and Systems*. URL: https://www.academia.edu/52398369/Multilayer_perceptron_and_neural_networks (visited on 02/09/2024).

Battaglia, Peter W., Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Caglar Gulcehre, Francis Song, Andrew Ballard, Justin Gilmer, George Dahl, Ashish Vaswani, Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matt Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu (Oct. 2018). *Relational inductive biases, deep learning, and graph networks*. arXiv:1806.01261 [cs, stat]. DOI: 10.48550/arXiv.1806.01261. URL: http://arxiv.org/abs/1806.01261 (visited on 01/17/2024).

Behandish, M. and Z.Y. Wu (2014). “Concurrent Pump Scheduling and Storage Level Optimization Using Meta-models and Evolutionary Algorithms”. en. In: *Procedia Engineering* 70, pp. 103–112. ISSN: 18777058. DOI: 10.1016/j.proeng.2014.02.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877705814000150 (visited on 11/16/2023).

Blank, Julian and Kalyanmoy Deb (2020). “Pymoo: Multi-Objective Optimization in Python”. In: *IEEE Access* 8, pp. 89497–89509. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.2990567. URL: https://ieeexplore.ieee.org/document/9078759 (visited on 10/26/2023).

Bohórquez, Jessica, Juan Saldarriaga, and Daniel Vallejo (2015). “Pumping Pattern Optimization in Order to Reduce WDS Operation Costs”. en. In: *Procedia Engineering* 119, pp. 1069–1077. ISSN: 18777058. DOI: 10.1016/j.proeng.2015.08.936. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877705815026065 (visited on 06/07/2023).

Broad, D. R., G. C. Dandy, and H. R. Maier (May 2005). “Water Distribution System Optimization Using Metamodels”. EN. In: *Journal of Water Resources Planning and Management* 131.3, pp. 172–180. ISSN: 0733-9496. DOI: 10.1061/(ASCE)0733-9496(2005)131:3(172). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9496%282005%29131%3A3%28172%29 (visited on 03/30/2023).

BUBER, Ebubekir and Banu DIRI (Oct. 2018). “Performance Analysis and CPU vs GPU Comparison for Deep Learning”. In: *2018 6th International Conference on Control Engineering & Information Technology (CEIT)*, pp. 1–6. DOI: 10.1109/CEIT.2018.8751930. URL: https://ieeexplore.ieee.org/abstract/document/8751930 (visited on 01/01/2024).

Buenfil, Mario O. Rodriguez (Oct. 2004). “Water Distribution System Operation”. en. In: *Water Encyclopedia*. Ed. by Jay H. Lehr and Jack Keeley. 1st ed. Wiley, pp. 200–204. ISBN: 9780471441649. DOI: 10.1002/047147844X.mw551. URL: https://onlinelibrary.wiley.com/doi/10.1002/047147844X.mw551 (visited on 02/09/2024).

Buhrmester, Vanessa, David Münch, and Michael Arens (Dec. 2021). “Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey”. en. In: *Machine Learning and Knowl-*

edge Extraction 3.4, pp. 966–989. ISSN: 2504-4990. DOI: 10.3390/make3040048. URL: https://www.mdpi.com/2504-4990/3/4/48 (visited on 02/11/2024).

Castro-Gama, Mario, Quan Pan, Emilio Attilio Lanfranchi, Andreja Jonoski, and Dimitri P. Solomatine (2017). "Pump Scheduling for a Large Water Distribution Network. Milan, Italy". en. In: *Procedia Engineering* 186, pp. 436–443. ISSN: 18777058. DOI: 10.1016/j.proeng.2017.03.249. URL: https://linkinghub.elsevier.com/retrieve/pii/S1877705817314029 (visited on 06/07/2023).

Chong, Edwin K. P., Wu-Sheng Lu, and Stanislaw H. Zak (Oct. 2023). *An Introduction to Optimization: With Applications to Machine Learning*. en. Google-Books-ID: uEDUEAAAQBAJ. John Wiley & Sons. ISBN: 9781119877639.

D'Ambrosio, C., C. Bragalli, Jon Lee, Andrea Lodi, and P. Toth (2008). "Water Network Design by MINLP". In: URL: https://www.semanticscholar.org/paper/Water-Network-Design-by-MINLP-D%E2%80%99Ambrosio-Bragalli/39b2ec4d0736041b0fe2c4de4c4ea5bf522a7b8b (visited on 01/01/2024).

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (Apr. 2002). "A fast and elitist multiobjective genetic algorithm: NSGA-II". In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197. ISSN: 1941-0026. DOI: 10.1109/4235.996017. URL: https://ieeexplore.ieee.org/document/996017 (visited on 02/07/2024).

Di Bucchianico, Alessandro (Dec. 2007). "Coefficient of Determination". en. In: *Encyclopedia of Statistics in Quality and Reliability*. Ed. by Fabrizio Ruggeri, Ron S. Kenett, and Frederick W. Faltin. 1st ed. Wiley. ISBN: 9780470018613. DOI: 10.1002/9780470061572.eqr173. URL: https://onlinelibrary.wiley.com/doi/10.1002/9780470061572.eqr173 (visited on 02/08/2024).

Do, Nhu (2023). "Optimal scheduling of variable speed pumps in water distribution systems using genetic algorithms". In: (). URL: https://www.academia.edu/60524907/Optimal_scheduling_of_variable_speed_pumps_in_water_distribution_systems_using_genetic_algorithms (visited on 12/02/2023).

Donkor, Emmanuel A., Thomas A. Mazzuchi, Refik Soyer, and J. Alan Roberson (Feb. 2014). "Urban Water Demand Forecasting: Review of Methods and Models". en. In: *Journal of Water Resources Planning and Management* 140.2, pp. 146–159. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)WR.1943-5452.0000314. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0000314 (visited on 01/23/2024).

Fernández-Godino, M. Giselle (Apr. 2023). *Review of multi-fidelity models*. arXiv:1609.07196 [stat]. URL: http://arxiv.org/abs/1609.07196 (visited on 10/19/2023).

Fooladivanda, Dariush and Joshua A. Taylor (Dec. 2015). "Optimal pump scheduling and water flow in water distribution networks". In: *2015 54th IEEE Conference on Decision and Control (CDC)*, pp. 5265–5271. DOI: 10.1109/CDC.2015.7403043. URL: https://ieeexplore.ieee.org/document/7403043 (visited on 11/13/2023).

Gagliardi, Francesca, Stefano Alvisi, Zoran Kapelan, and Marco Franchini (July 2017). "A Probabilistic Short-Term Water Demand Forecasting Model Based on the Markov Chain". en. In: *Water* 9.7, p. 507. ISSN: 2073-4441. DOI: 10.3390/w9070507. URL: https://www.mdpi.com/2073-4441/9/7/507 (visited on 02/29/2024).

Garzón, Alexander, Zoran Kapelan, J. Langeveld, and Riccardo Taormina (2022). "Machine Learning-Based Surrogate Modeling for Urban Water Networks: Review and Future Research Directions". In: *Water Resources Research* 58.5, e2021WR031808.

Giacomello, Carlo, Zoran Kapelan, and Matteo Nicolini (Mar. 2013). "Fast Hybrid Optimization Method for Effective Pump Scheduling". en. In: *Journal of Water Resources Planning and Management* 139.2, pp. 175–183. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)WR.1943-5452.0000239. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0000239 (visited on 03/10/2024).

Hutton, Christopher J. and Zoran Kapelan (Apr. 2015). "A probabilistic methodology for quantifying, diagnosing and reducing model structural and predictive errors in short term water demand forecasting". In: *Environmental Modelling & Software* 66, pp. 87–97. ISSN: 1364-8152. DOI: 10.1016/j.envsoft.2014.12.021. URL: https://www.sciencedirect.com/science/article/pii/S1364815214003788 (visited on 02/29/2024).

Hwang, Hwee and Kevin Lansey (Dec. 2017). "Water Distribution System Classification Using System Characteristics and Graph-Theory Metrics". en. In: *Journal of Water Resources Planning and Management* 143.12, p. 04017071. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)WR.1943-5452.

0000850. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29WR.1943-5452.0000850 (visited on 02/13/2024).

IEA (Oct. 2023). *World Energy Outlook 2023*. en-GB. Tech. rep. Paris: IEA. URL: https://www.iea.org/reports/world-energy-outlook-2023 (visited on 03/04/2024).

Jowitt, Paul W. and George Germanopoulos (July 1992). "Optimal Pump Scheduling in Water□Supply Networks". en. In: *Journal of Water Resources Planning and Management* 118.4, pp. 406–422. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)0733-9496(1992)118:4(406). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9496%281992%29118%3A4%28406%29 (visited on 06/07/2023).

Katoch, Sourabh, Sumit Singh Chauhan, and Vijay Kumar (2021). "A review on genetic algorithm: past, present, and future". In: *Multimedia Tools and Applications* 80.5, pp. 8091–8126. ISSN: 1380-7501. DOI: 10.1007/s11042-020-10139-6. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7599983/ (visited on 02/07/2024).

Kentucky Water Resources Research Institute, University of Kentucky (2024). *Water Distribution System Research Database*. DOI: 10.13023/KWRRI.WDSRD. URL: https://uknowledge.uky.edu/wdsrd/ (visited on 01/25/2024).

Kingma, Diederik P. and Jimmy Ba (Jan. 2017). *Adam: A Method for Stochastic Optimization*. arXiv:1412.6980 [cs]. DOI: 10.48550/arXiv.1412.6980. URL: http://arxiv.org/abs/1412.6980 (visited on 01/08/2024).

Klise, Katherine A., Regan Murray, and Terra Haxton (Apr. 2018). *AN OVERVIEW OF THE WATER NETWORK TOOL FOR RESILIENCE (WNTR).* English. Tech. rep. SAND2018-4462C. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States). URL: https://www.osti.gov/biblio/1510389 (visited on 01/29/2024).

Kocijan, Jus, Nadja Hvala, Matija Perne, Primoz Mlakar, Bostjan Grasic, and Marija Zlata Boznar (May 2022). *Surrogate modelling for the forecast of Seveso-type atmospheric pollutant dispersion*. preprint. In Review. DOI: 10.21203/rs.3.rs-1640949/v1. URL: https://www.researchsquare.com/article/rs-1640949/v1 (visited on 10/19/2023).

Lansey, Kevin E. and Kofi Awumah (Jan. 1994). "Optimal Pump Operations Considering Pump Switches". en. In: *Journal of Water Resources Planning and Management* 120.1, pp. 17–35. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)0733-9496(1994)120:1(17). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9496%281994%29120%3A1%2817%29 (visited on 11/23/2023).

Latin American Computing Conference, Mauricio Solar, David Fernández-Baca, Ernesto Cuadros-Vargas, Centro Latinoamericano de Estudios en Informática., and Congreso Iberoamericano de Educación Superior en Computación (2004). *CLEI 2004, XXX Conferencia Latinoamericana en Informática: Arequipa, Perú, 27 Septiembre-1 Octubre : resúmenes*. eng. OCLC: 57174940. [Arequipa, Perú]: CLEI. ISBN: 9789972987625.

Lee, Hoesung et al. (2023). *IPCC, 2023: Climate Change 2023: Synthesis Report, Summary for Policymakers. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland.* en. Tech. rep. URL: https://doi.org/10.59327/IPCC/AR6-9789291691647.001 (visited on 02/09/2024).

Leutbecher, M. and T. N. Palmer (Mar. 2008). "Ensemble forecasting". In: *Journal of Computational Physics*. Predicting weather, climate and extreme events 227.7, pp. 3515–3539. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2007.02.014. URL: https://www.sciencedirect.com/science/article/pii/S0021999107000812 (visited on 02/29/2024).

Liu, Xuhan, Adriaan P. IJzerman, and Gerard J. P. van Westen (2021). "Computational Approaches for De Novo Drug Design: Past, Present, and Future". en. In: *Artificial Neural Networks*. Ed. by Hugh Cartwright. Methods in Molecular Biology. New York, NY: Springer US, pp. 139–165. ISBN: 9781071608265. DOI: 10.1007/978-1-0716-0826-5_6. URL: https://doi.org/10.1007/978-1-0716-0826-5_6 (visited on 02/23/2024).

López-Ibáñez, Manuel, T. Devi Prasad, and Ben Paechter (July 2008). "Ant Colony Optimization for Optimal Control of Pumps in Water Distribution Networks". en. In: *Journal of Water Resources Planning and Management* 134.4, pp. 337–346. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)0733-9496(2008)134:4(337). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9496%282008%29134%3A4%28337%29 (visited on 03/03/2024).

Mackle, G., G.A. Savic, and G.A. Walters (Sept. 1995). “Application of genetic algorithms to pump scheduling for water supply”. In: *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*. ISSN: 0537-9989, pp. 400–405. DOI: 10.1049/cp:19951082. URL: https://ieeexplore.ieee.org/abstract/document/501705 (visited on 12/02/2023).

Madhiarasan, Manogaran and Mohamed Louzazni (Apr. 2022). “Analysis of Artificial Neural Network: Architecture, Types, and Forecasting Applications”. en. In: *Journal of Electrical and Computer Engineering* 2022, e5416722. ISSN: 2090-0147. DOI: 10.1155/2022/5416722. URL: https://www.hindawi.com/journals/jece/2022/5416722/ (visited on 02/21/2024).

Maier, H.R., Z. Kapelan, J. Kasprzyk, J. Kollat, L.S. Matott, M.C. Cunha, G.C. Dandy, M.S. Gibbs, E. Keedwell, A. Marchi, A. Ostfeld, D. Savic, D.P. Solomatine, J.A. Vrugt, A.C. Zecchin, B.S. Minsker, E.J. Barbour, G. Kuczera, F. Pasha, A. Castelletti, M. Giuliani, and P.M. Reed (Dec. 2014). “Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions”. en. In: *Environmental Modelling & Software* 62, pp. 271–299. ISSN: 13648152. DOI: 10.1016/j.envsoft.2014.09.013. URL: https://linkinghub.elsevier.com/retrieve/pii/S1364815214002679 (visited on 06/07/2023).

Makaremi, Yasaman, Ali Haghighi, and Hamid Reza Ghafouri (Mar. 2017). “Optimization of Pump Scheduling Program in Water Supply Systems Using a Self-Adaptive NSGA-II; a Review of Theory to Real Application”. en. In: *Water Resources Management* 31.4, pp. 1283–1304. ISSN: 1573-1650. DOI: 10.1007/s11269-017-1577-x. URL: https://doi.org/10.1007/s11269-017-1577-x (visited on 12/02/2023).

Masłoń, Adam, Joanna Czarnota, Aleksandra Szaja, Joanna Szulżyk-Cieplak, and Grzegorz Łagód (Jan. 2020). “The Enhancement of Energy Efficiency in a Wastewater Treatment Plant through Sustainable Biogas Use: Case Study from Poland”. en. In: *Energies* 13.22, p. 6056. ISSN: 1996-1073. DOI: 10.3390/en13226056. URL: https://www.mdpi.com/1996-1073/13/22/6056 (visited on 03/03/2024).

Michalewicz, Zbigniew (1996). “GAs: What Are They?” en. In: *Genetic Algorithms + Data Structures = Evolution Programs*. Ed. by Zbigniew Michalewicz. Berlin, Heidelberg: Springer, pp. 13–31. ISBN: 9783662033159. DOI: 10.1007/978-3-662-03315-9_2. URL: https://doi.org/10.1007/978-3-662-03315-9_2 (visited on 02/09/2024).

Monga, Vishal, Yuelong Li, and Yonina C. Eldar (Mar. 2021). “Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing”. In: *IEEE Signal Processing Magazine* 38.2, pp. 18–44. ISSN: 1558-0792. DOI: 10.1109/MSP.2020.3016905.

Moreira, Daniel F. and Helena M. Ramos (2013). “Energy Cost Optimization in a Water Supply System Case Study”. en. In: *Journal of Energy* 2013, pp. 1–9. ISSN: 2314-615X. DOI: 10.1155/2013/620698. URL: http://www.hindawi.com/journals/jen/2013/620698/ (visited on 06/02/2023).

Namdari, Hossein, Ali Haghighi, and Seyed Mohammad Ashrafi (Sept. 2023). “Short-term urban water demand forecasting; application of 1D convolutional neural network (1D CNN) in comparison with different deep learning schemes”. en. In: *Stochastic Environmental Research and Risk Assessment*. ISSN: 1436-3259. DOI: 10.1007/s00477-023-02565-3. URL: https://doi.org/10.1007/s00477-023-02565-3 (visited on 02/06/2024).

Niu, Dan, Zhenguo Kuang, Xisong Chen, Shuang Wei, Jun Yang, and Xiangyu Wang (Sept. 2018). “Optimizing Pump Scheduling for Water Supply through Improved Multiple Population Genetic Algorithm”. In: *2018 24th International Conference on Automation and Computing (ICAC)*, pp. 1–6. DOI: 10.23919/IConAC.2018.8748955. URL: https://ieeexplore.ieee.org/document/8748955 (visited on 12/02/2023).

Ormsbee, Lindell E. and Kevin E. Lansey (Mar. 1994). “Optimal Control of Water Supply Pumping Systems”. en. In: *Journal of Water Resources Planning and Management* 120.2, pp. 237–252. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)0733-9496(1994)120:2(237). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9496%281994%29120%3A2%28237%29 (visited on 10/19/2023).

Parisio, Alessandra and Luigi Glielmo (Oct. 2011). “A mixed integer linear formulation for microgrid economic scheduling”. In: *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 505–510. DOI: 10.1109/SmartGridComm.2011.6102375. URL: https://ieeexplore.ieee.org/document/6102375/ (visited on 01/03/2024).

Razavi, Saman, Bryan A. Tolson, and Donald H. Burn (2012). “Review of surrogate modeling in water resources”. en. In: *Water Resources Research* 48.7. ISSN: 1944-7973. DOI: 10.1029/2011WR01

1527. URL: https://onlinelibrary.wiley.com/doi/abs/10.1029/2011WR011527 (visited on 03/30/2023).

Rossman, L., H. Woo, M. Tryby, F Shang, R. Janke, and T. Haxton (2020). *EPANET 2.2 User Manual*.

Sarbu, Ioan (Dec. 2016). "A Study of Energy Optimisation of Urban Water Distribution Systems Using Potential Elements". en. In: *Water* 8.12, p. 593. ISSN: 2073-4441. DOI: 10.3390/w8120593. URL: http://www.mdpi.com/2073-4441/8/12/593 (visited on 06/02/2023).

Solà Roca, Albert (2023). "GGANet: Algorithm Unrolling for Water Distribution Networks Metamodelling". en. In: URL: https://repository.tudelft.nl/islandora/object/uuid%3A535f5494-bf0c-4a8c-bb27-116a7cf1918b (visited on 02/02/2024).

Steffelbauer, D.B., B. Hillebrand, and E.J.M. Blokker (July 2022). "pySIMDEUM: An open-source stochastic water demand end-use model in Python". In: *Proceedings of the 2nd joint Water Distribution System Analysis and Computing and Control in the Water Industry (WDSA/CCWI2022) conference, Valencia (Spain)*.

Terna (Mar. 2024). *Transparency Report: Actual Generation*. en. URL: https://www.terna.it/en/electric-system/transparency-report/actual-generation (visited on 03/07/2024).

Texas Commission on Environmental Quality (Oct. 2023). en. Official Government Organization of the U.S. URL: https://www.tceq.texas.gov/drinkingwater/technical_guidance/staff_guidance/capacity (visited on 10/26/2023).

Todini, E. and S. Pilati (June 1988). "A gradient algorithm for the analysis of pipe networks". In: *Computer applications in water supply: vol. 1—systems analysis and simulation*. GBR: Research Studies Press Ltd., pp. 1–20. ISBN: 9780471917830. (Visited on 03/07/2024).

Todini, Ezio and Lewis A. Rossman (May 2013). "Unified Framework for Deriving Simultaneous Equation Algorithms for Water Distribution Networks". en. In: *Journal of Hydraulic Engineering* 139.5, pp. 511–526. ISSN: 0733-9429, 1943-7900. DOI: 10.1061/(ASCE)HY.1943-7900.0000703. URL: https://ascelibrary.org/doi/10.1061/%28ASCE%29HY.1943-7900.0000703 (visited on 06/02/2023).

Utesch, Fabian, Alexander Brandies, Paulin Pekezou Fouopi, and Caroline Schießl (July 2020). "Towards behaviour based testing to understand the black box of autonomous cars". In: *European Transport Research Review* 12.1, p. 48. ISSN: 1866-8887. DOI: 10.1186/s12544-020-00438-2. URL: https://doi.org/10.1186/s12544-020-00438-2 (visited on 02/11/2024).

Walski, Thomas M. (Aug. 2001). "The Wrong Paradigm—Why Water Distribution Optimization Doesn't Work". en. In: *Journal of Water Resources Planning and Management* 127.4, pp. 203–205. ISSN: 0733-9496, 1943-5452. DOI: 10.1061/(ASCE)0733-9496(2001)127:4(203). URL: https://ascelibrary.org/doi/10.1061/%28ASCE%290733-9496%282001%29127%3A4%28203%29 (visited on 10/19/2023).

Yang, Xin-She (Jan. 2014). "Chapter 5 - Genetic Algorithms". In: *Nature-Inspired Optimization Algorithms*. Ed. by Xin-She Yang. Oxford: Elsevier, pp. 77–87. ISBN: 9780124167438. DOI: 10.1016/B978-0-12-416743-8.00005-1. URL: https://www.sciencedirect.com/science/article/pii/B9780124167438000051 (visited on 03/04/2024).

Zhang, Jingzhao, Tianxing He, Suvrit Sra, and Ali Jadbabaie (May 2019). *Why gradient clipping accelerates training: A theoretical justification for adaptivity*. en. URL: https://arxiv.org/abs/1905.11881v2 (visited on 01/08/2024).

Zhang, Zhongheng, Marcus W. Beck, David A. Winkler, Bin Huang, Wilbert Sibanda, and Hemant Goyal (June 2018). "Opening the black box of neural networks: methods for interpreting neural network models in clinical applications". In: *Annals of Translational Medicine* 6.11, p. 216. ISSN: 2305-5839. DOI: 10.21037/atm.2018.05.32. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6035992/ (visited on 02/11/2024).

A

Appendix

A.1. Hardware and training

The development of any novel approach in machine learning can require fine-tuning and iterations until it produces reasonable results. This is why it was considered unnecessary to use a supercomputer during training. When changes are constantly being made to the datasets and the models, there is little sense in submitting every change as a batch job in the cloud for it to be evaluated. It is much faster if one can train locally and quickly evaluate results, albeit the dataset size, here referring to the sample size but also the network's complexity, of this local environment has to be simplified.

This is what was achieved with Fossolo, as it is a simple network with around 40 junctions and 60 pipes. Training models on this network rarely took more than 15 minutes with 5 minutes being the average. This was all performed on a laptop with a powerful CPU (11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz). Machine learning experts would normally opt for a GPU instead of a CPU for model training, as they tend to be faster BUBER et al. (2018). Still, this is only the case for large datasets and models. For the ones explored in this paper, there was only a marginal speedup when training on a desktop computer with a powerful GPU (GTX 970).

What is more surprising from comparing the differences between the models produced with identical settings and PyTorch+Cuda setups but different hardware is that the models' performance differed significantly. The ones trained on the laptop CPU almost always outperformed the ones trained on the GPU. This difference was as high as 10% when comparing RMSE and can not be attributed to inconsistent floating point accuracy. This is another area of future research, with a possible hardware or software bug being the culprit. It could also be that some quality of unrolling causes models to be especially sensitive to small decimal differences, for example, ones caused by rounding.

After this was confirmed it was decided to keep using the GPU to benefit from the ability to be training models simultaneously and then record their results. In the instance of promising models trained on the GPU, their results were replicated on the CPU to ensure validity.

A.2. Predicted nodes' location

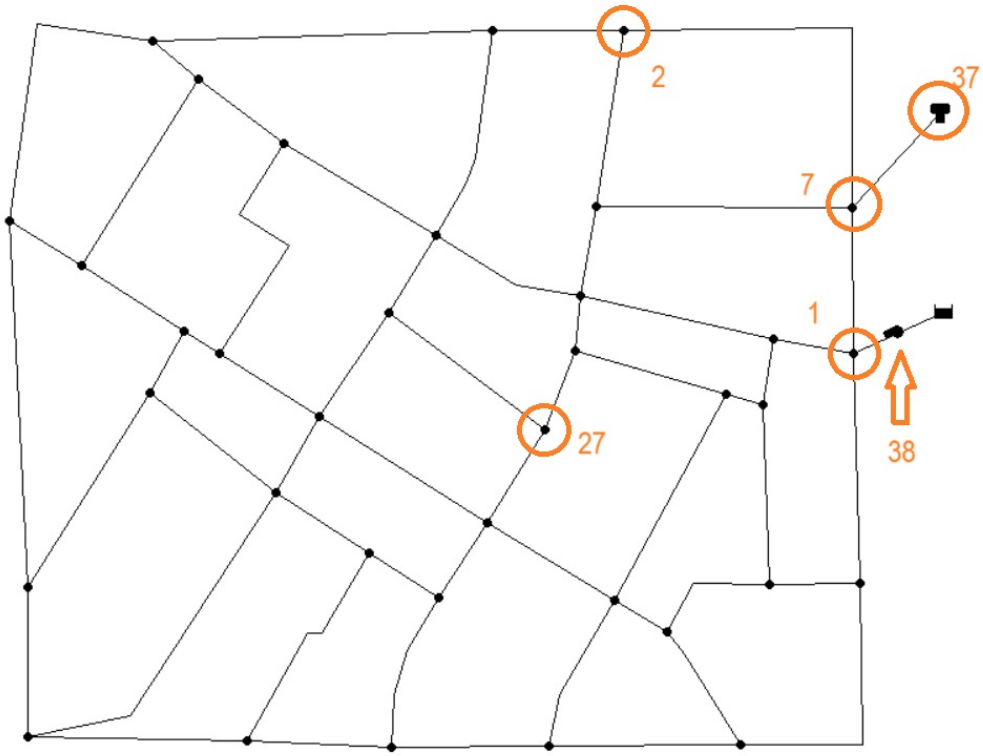


Figure A.1: Highlighted nodes whose predictions were graphed

A.3. Fossolo results

A.3.1. Continuity results

Multi-Layer Perceptron Starting with the Multi-Layer Perceptron (MLP), its results are in figure A.2. Two graphs indicating the general performance are included here. Examining the sub-figures unveils a large mismatch. The MLP appears as if determining an average value instead of fitting to the data. This was the case with all MLPs trained.

Finally, the first node displayed in the plots is the one next to the reservoir. At first, it appears that this prediction is erroneous but paying attention to the vertical axis values reveals that the prediction for this node is at most off by 0.1m of head.

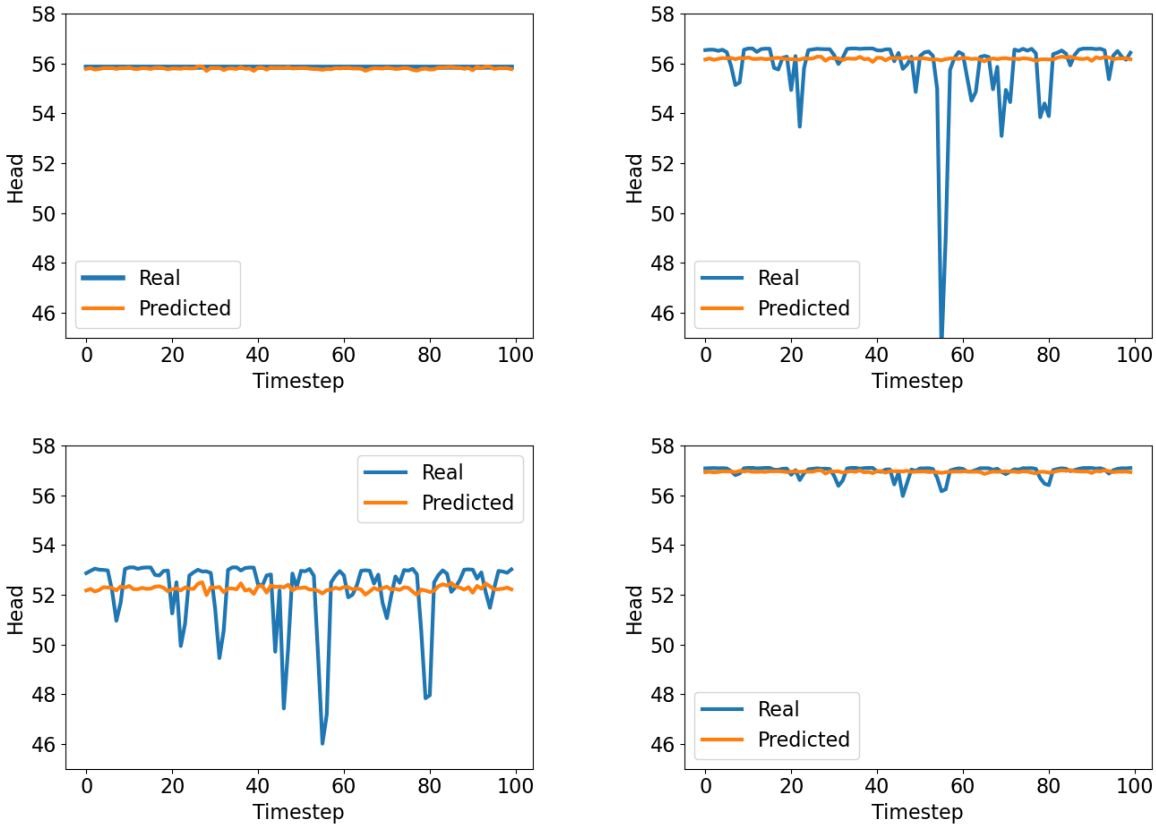


Figure A.2: MLP predicting continuous simulation (4 days)

The prediction of the node *Next to Reservoir - 1* is the only node for which we can consider the prediction by the MLP successful. The general prediction of the MLP for continuity is considered poor. Except for this node, predictions for the rest of the nodes are ineffective. An R2 score of 0.69 is deceiving in this case as, without being able to capture any trends, the MLP can at best provide an average of the system's state. This is not useful as it is precisely the variation due to demand that we are looking to capture.

BaselineUnrolling For the BaselineUnrolling (BU) model, prediction performance improves when judging by the metrics. There is a slight increase in the R2 and a significant decrease in the MSE. Besides improved metric performance, manually examining the values indicates that they are better aligned with the real values. *Next to Reservoir* predictions remain within a margin of around 0.1m. Then, for the rest of the nodes, the BU model is capable of capturing intra-day variability adequately, with the morning peak being captured quite clearly and the late-night peak to an extent. What is being referred to as a peak here is observed as a dip in the graphs, since when there is a rise in demand, this is reflected as a decrease in the average head of the system, as more water is withdrawn from it. Inter-day variations are not captured very well. Examining the orange lines closely makes it apparent that the model is not simply predicting the same trend for every day, but it has slight differences and adjustments depending on the input data. At the same time, they are not nearly enough to capture the substantial differences between days. The morning peak differs by as much as 5m of head in *Random Node - 7*, when comparing days 1 and 3, specifically, around hours 20 and 55, while effectively the same prediction is made for all 4 of the days.

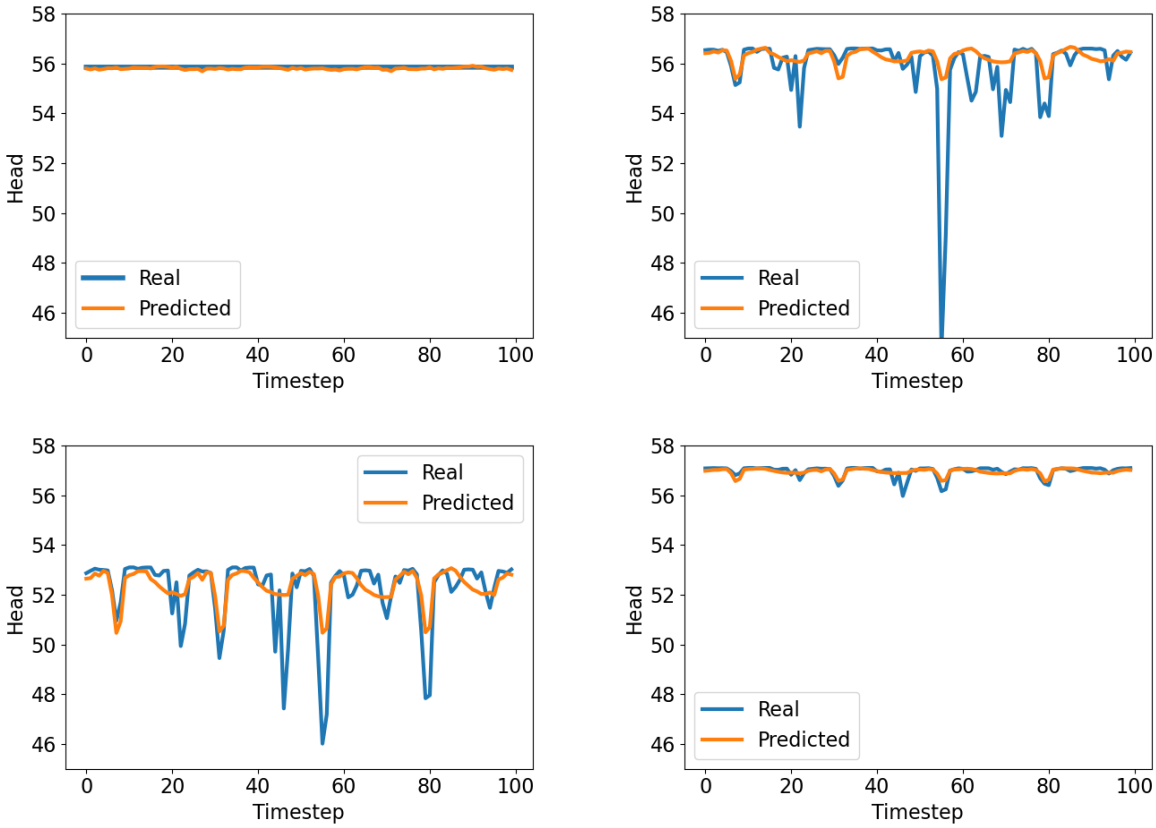


Figure A.3: BU predicting continuous simulation (4 days)

The BaselineUnrolling is much more suitable for surrogate modeling in comparison to the MLP. Unlike the MLP, BU is capturing trends even where there are none, such as in the case of the node next to the reservoir. The peaks and troughs of this trend thankfully do not deviate as much to tamper the quality of the prediction. Values remain within an absolute error of 0.1m. As for the rest of the system, knowing that it is possible to capture the variations of demand is a good indication to proceed. With an R2 score of 0.762, shown in table A.1, BU is a much better predictor than the MLP.

UnrollingModel The UnrollingModel's (UM) predictions are similar to the BU's. The only notable difference is that the UM takes more than double the time to train and requires almost triple times the number of parameters. Due to not containing any notable differences the graphs are omitted and included in the figure A.4. Similarly to the BU, the node next to the reservoir draws a trend that is nowhere to be found in the data for this node. Again, intra-day variability is captured to an extent, with inter-day barely.

In general, both of the models displayed for BU and UM are some of the best examples of overall network performance. It is theorized that increasing model complexity and dataset size should be capable of capturing more inter and intra-day variation, such as the late-night peak. This was not pursued further due to the tendency to produce unstable models, also detailed in section 6.2, along with worse overall prediction. Some attempts were made to limit instability, but for the most part, models that extended beyond a certain level of complexity led to exploding gradients and an arduous investigation of the causes, usually providing no better results. As mentioned in the same section, fixing this instability issue would be a key for future research looking to adapt and improve this process.

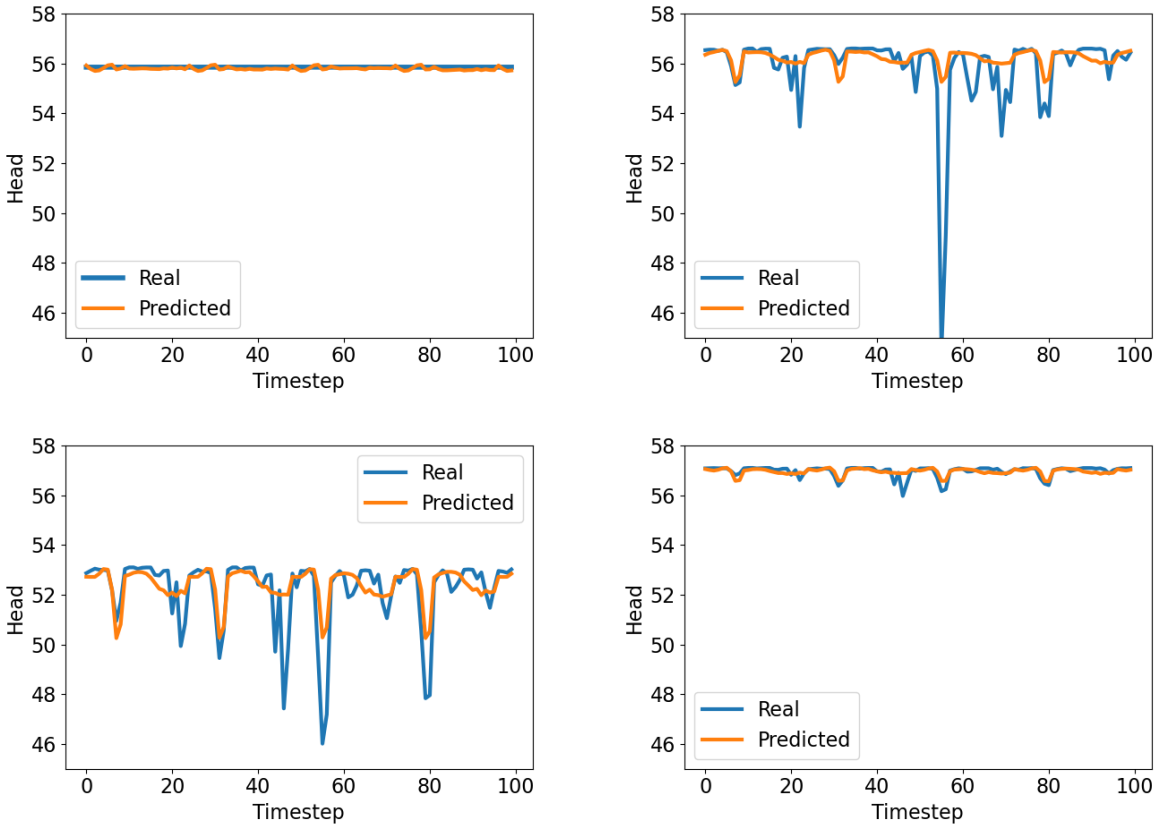


Figure A.4: UM predicting continuous simulation (4 days)

	MLP	BU	UM
R2	0.69	0.762	0.761
MSE	1.85	0.744	0.745
# of parameters	25892	12880	34417

Table A.1: Comparison metrics between models for continuous network

A.3.2. Tank results

A tank is added to the networks as specified in section 3.1.1. Hyper-parameters are then tuned and training ensues in the same manner.

Multi-Layer Perceptron Training the MLP with this modification produces similar results, showcased in figure A.5. The head values of the nodes included in the figure are either completely averaged out, like with node 7, or the pattern learned by the model appears random compared to the real values.

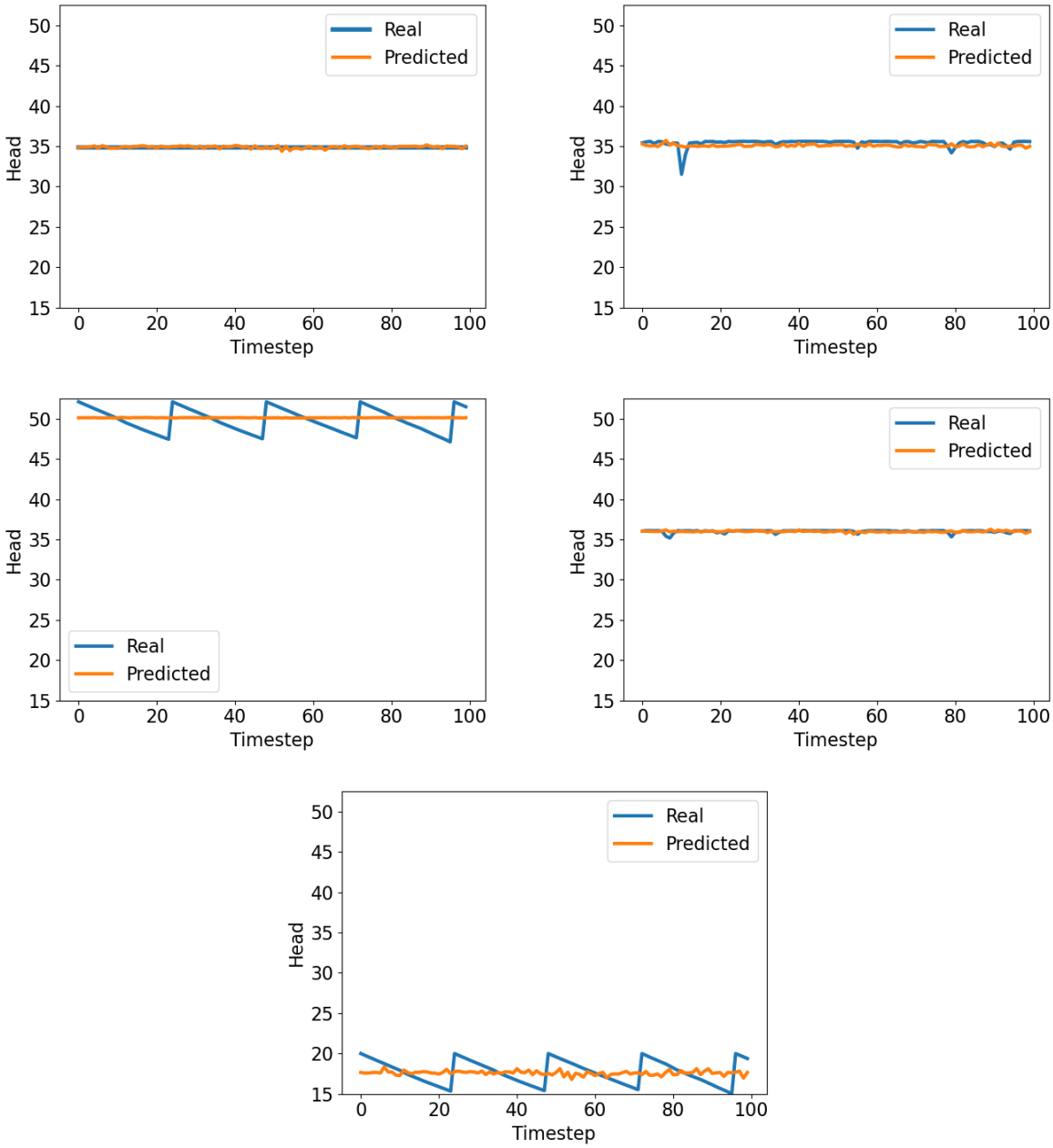


Figure A.5: MLP predictions including tank (4 days)

Examining closely nodes 1, 27, and 37 reveals that it is a quite similar pattern repeated for every node, with a different starting point. One can easily be deceived because of the change in the vertical axis range, which makes the predictions appear different between graphs even though they are following the same trends. This leads to the conclusion that the MLP constructed is incapable of capturing inter-nodal variance. It is essentially trying to fit the same pattern on all nodes, which is not possible due to their behavior being completely different, especially when considering tanks. Thus, the high R2 score is once again considered deceptive, and manual examination reveals that the model is unsuccessful.

BaselineUnrolling The BU model is quite successful at capturing the tank's behavior. The problem of high similarity among predictions remains along with a slightly higher mismatch when examining nodes 2 or 27. As for the node next to the tank and the tank itself, the heads are predicted almost perfectly, shown here in figure A.6.

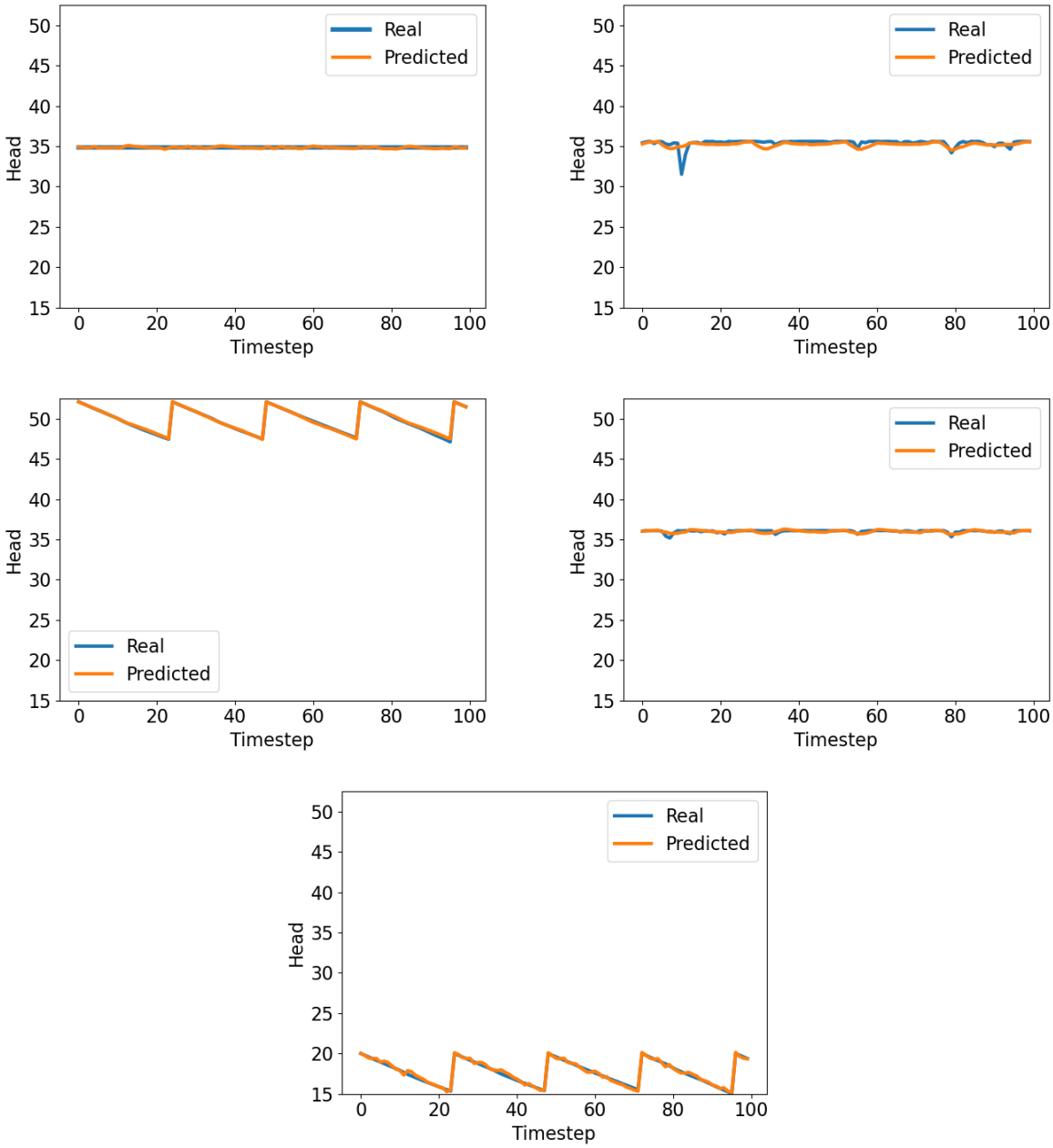


Figure A.6: BU predictions including tanks (4 days)

The degradation of the BU nodal predictions when the tank is included is unfortunate but not unexpected. The model is asked to capture a new element's behavior while retaining almost the same complexity as previously. The parameters are only increased as much as necessary to accommodate the tanks, meaning that the initial tank level is added to the input. No further model changes were made, like increasing the amount of layers in the model. When considering this, it is remarkable that the model was immediately able to capture the behavior of tanks so well. What is also interesting to note is the differences in the prediction structure of node 2 and node 27. Node 2, being close to the tank, has smoother predictions resembling the tank's, while node 27, being further away, has more variation in its pattern.

UnrollingModel Training the UM with the tanked version of Fossolo provides a similar outcome to BU for the node next to the tank and the tank itself. As for the rest of the nodes, predictions degrade

significantly, as can be seen in appendix figure A.7. Inter and intra-day variations are barely captured, with a satisfying average being predicted for all nodes.

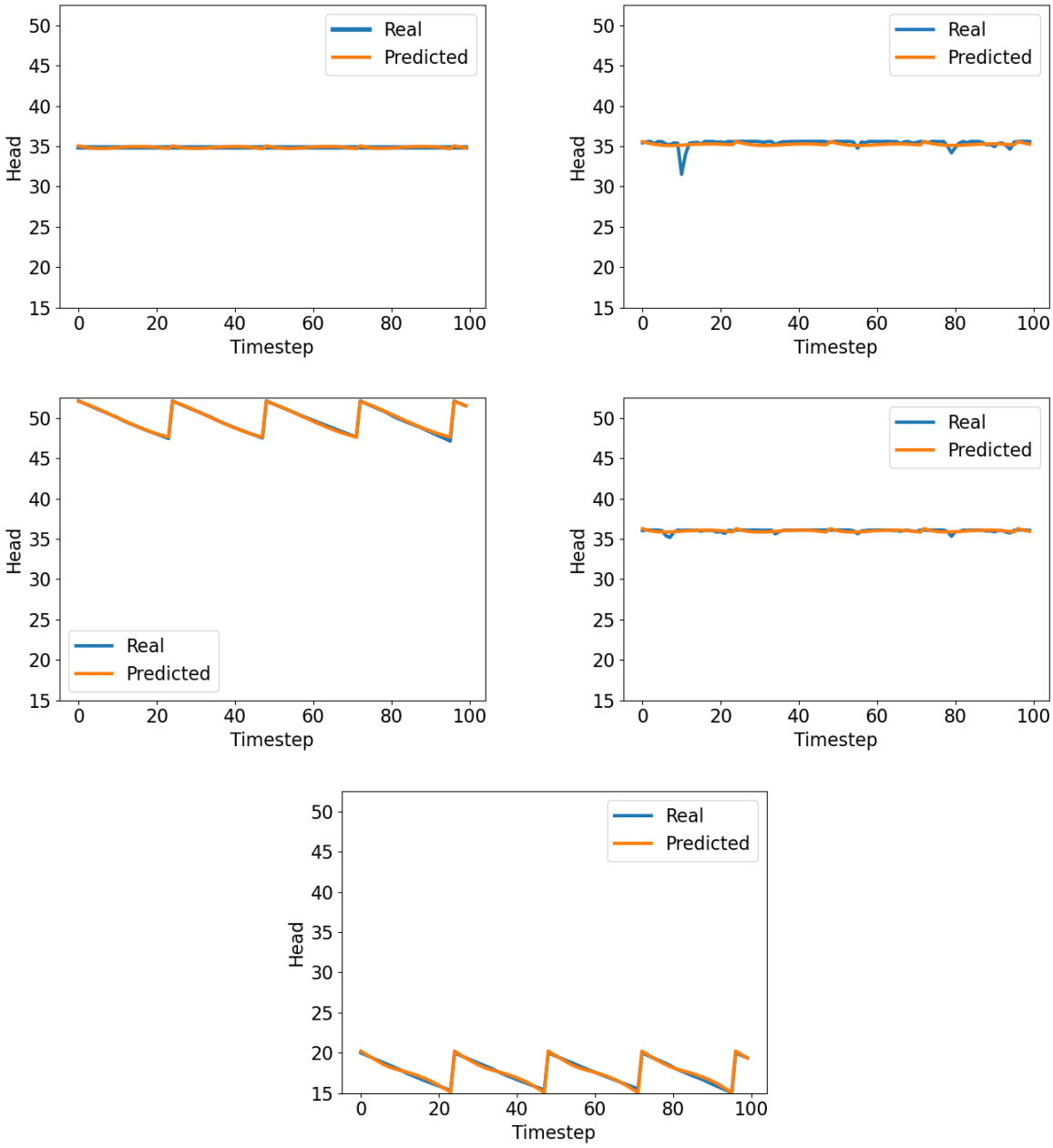


Figure A.7: UM predictions including tanks (4 days)

Understanding the reason why including tanks causes the UM's general performance to degrade would be an interesting subject for future research. The predictions of the UM are also smoother than the rest. Focusing on the tank, it looks like the UM is trying to fit some sort of sinusoidal pattern to it. This manages good performance but is unexpected and counter-intuitive behavior. One likely contributor to this is the fact that the initial tank level was not varied but kept constant between runs. As can be seen, even when there are variations in demand, reflected in the difference in nodal pressures, without a pump the tank is always simply draining in the same manner. This is likely confusing the model leading to it trying to transfer and fit this sinusoidal pattern to the rest of the nodes.

	MLP	BU	UM
R2	0.92	0.953	0.948
MSE	1.4	0.79	0.87
# of parameters	26021	13493	35878

Table A.2: Comparison metrics between models for tanked network

A.3.3. Pump and tank results

The performance of the MLP does not change when the pump is added. It simply also manages to predict an average for the pump. The MLP is hardly better than an average predictor.

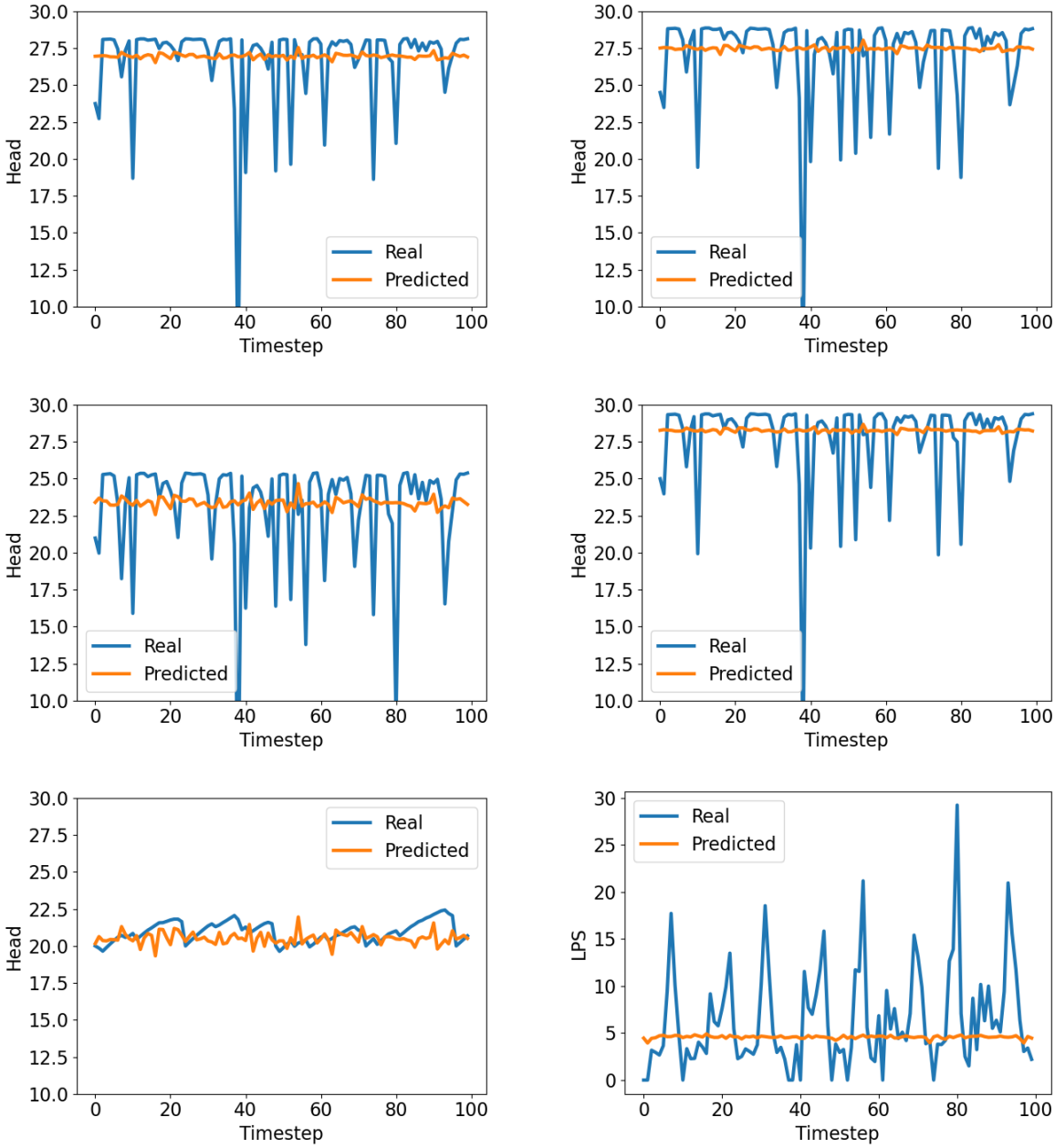


Figure A.8: MLP predicting network with a pump and a tank (4 days)

BaselineUnrolling The BU model manages to capture the pump flow and retain some of its performance when predicting tanks but stops being able to capture a generic node's head. Specifically, it is unable to capture most of the nodes' head drops due to the pump schedule and barely captures peaks in demand. The tank prediction also worsens as the tank pattern is much more erratic due to the pump operation.

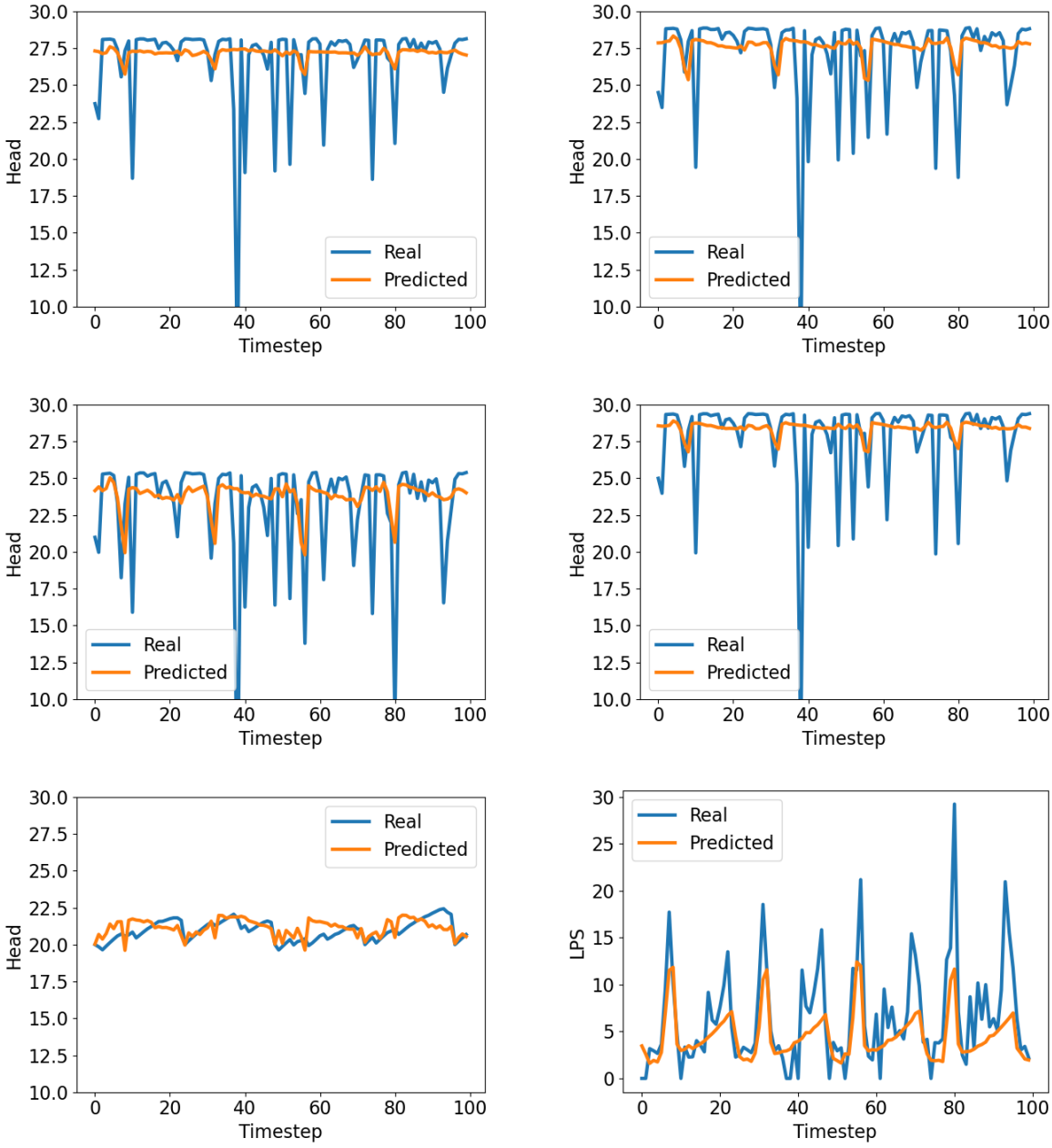


Figure A.9: BU predicting network with a pump and a tank (4 days)

Thus, it looks like another example where the added complexity was captured but at the cost of the previous nodes' performance. This means that, much like with the MLP, BU would be a sub-par candidate for surrogate modeling. Manual examination combined with a low R2 score (0.64) is enough to exclude it.

UnrollingModel Finally, the UM model is presented. This is a version constructed by inserting the pump schedule into the existing UM's input. This is a small modification performed with the hope that the network can incorporate this schedule into the rest of its structure, as described in the methodology section 3.1.1.

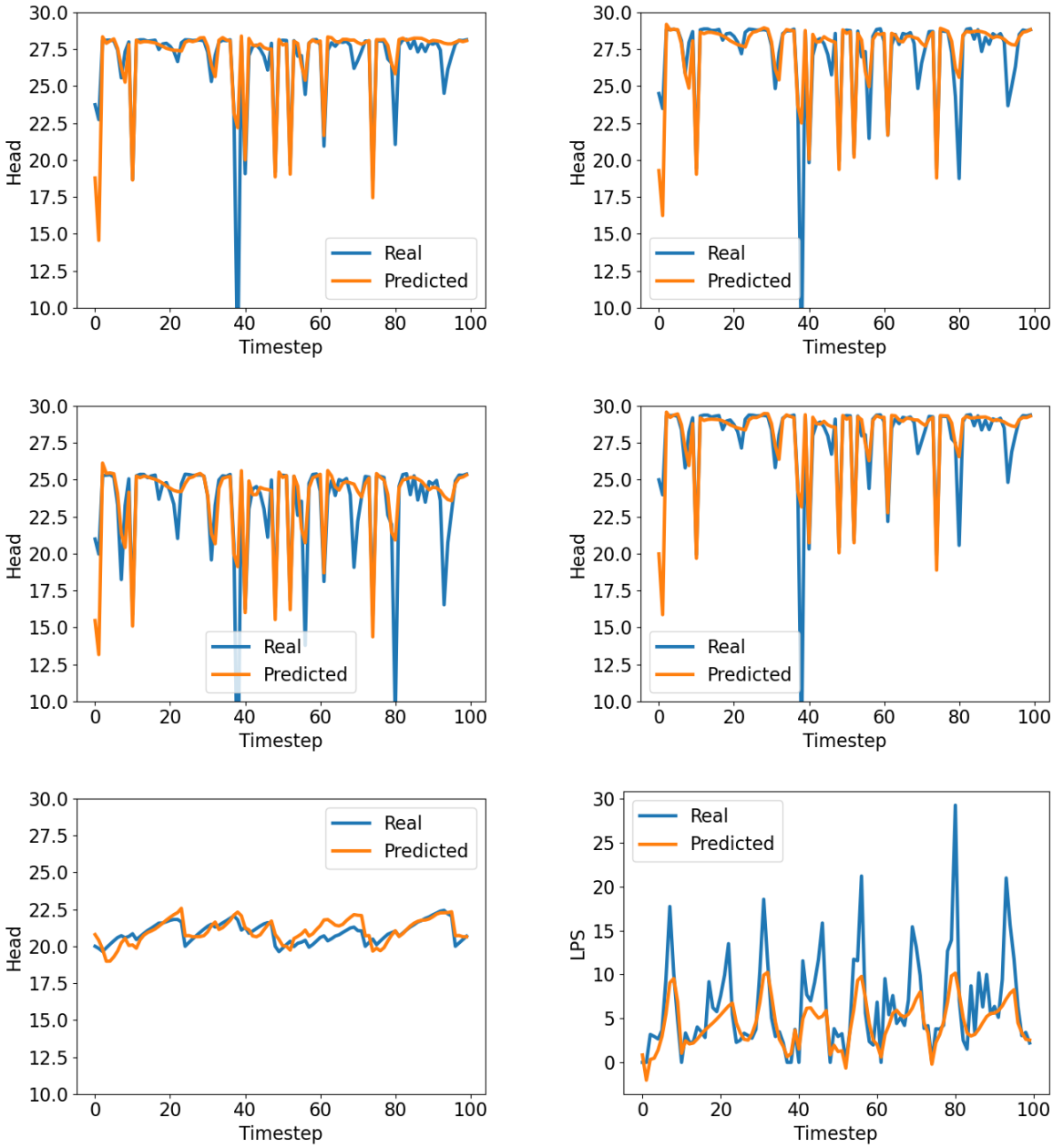


Figure A.10: UM predicting network with a pump and a tank (4 days)

This final model is the focus of this thesis. It is performing well considering that the previous UM model trained on the tanked version of Fossolo was unable to capture variability in nodal heads. This model captures not only the effects of the pump schedule but also of changing demand. The large drops in the head are a result of the pump being turned off but there are also small ones, like in node 2 around hour 20, that are captured. This behavior is almost miraculous but in line with typical network behavior. It can often be the case that a model is not simply the sum of its parts. Capturing complexity is not necessarily a linear process moving from simple to more complex predictive tasks Abbott (1994). Nevertheless, a

future investigation would be useful to understand the differences between the networks and why these lead to such different model behaviors.

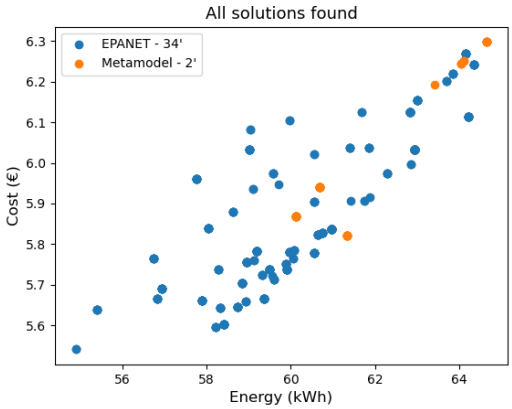


Figure A.11: Cost over energy for all solutions found during optimization

	MLP	BU	UM
R2	0.62	0.64	0.831
MSE	10.9	10.3	4.74
# of parameters	26278	13454	35278

Table A.3: Comparison metrics between models for Fossolo version with a pump and a tank

A.4. Net3 network and results

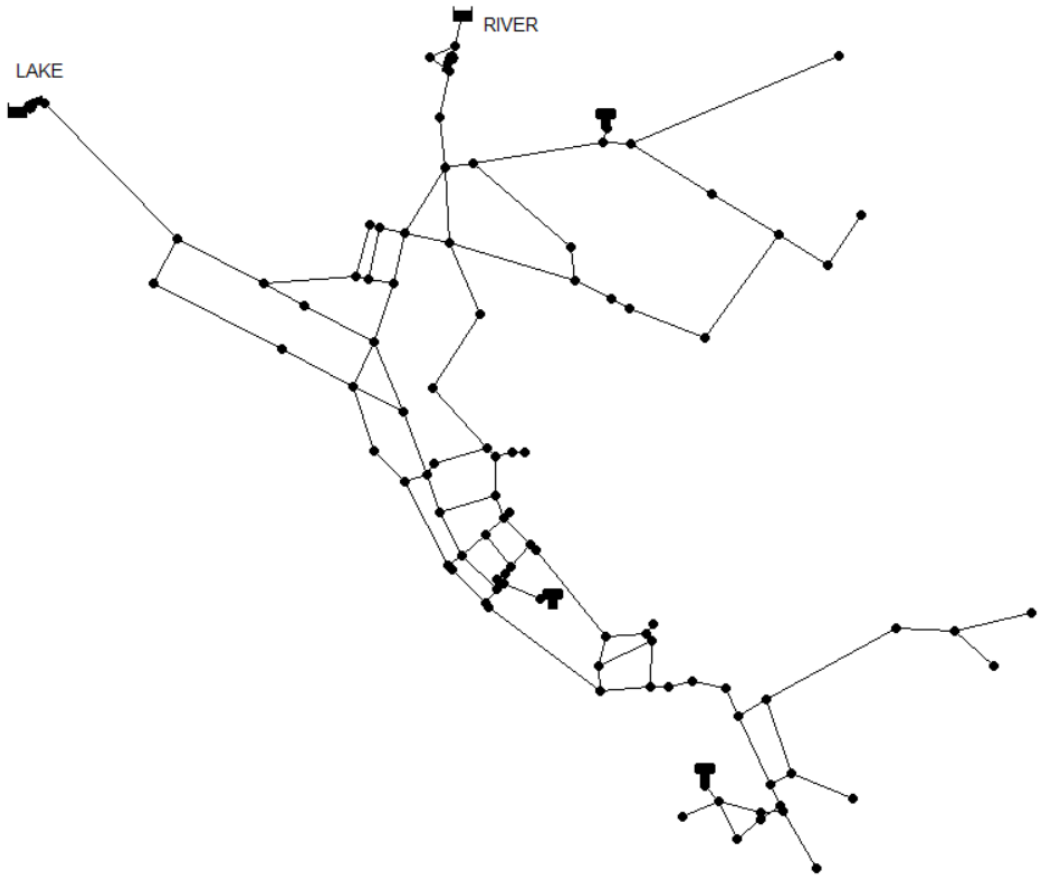


Figure A.12: An overview of the EPANET Net 3 network

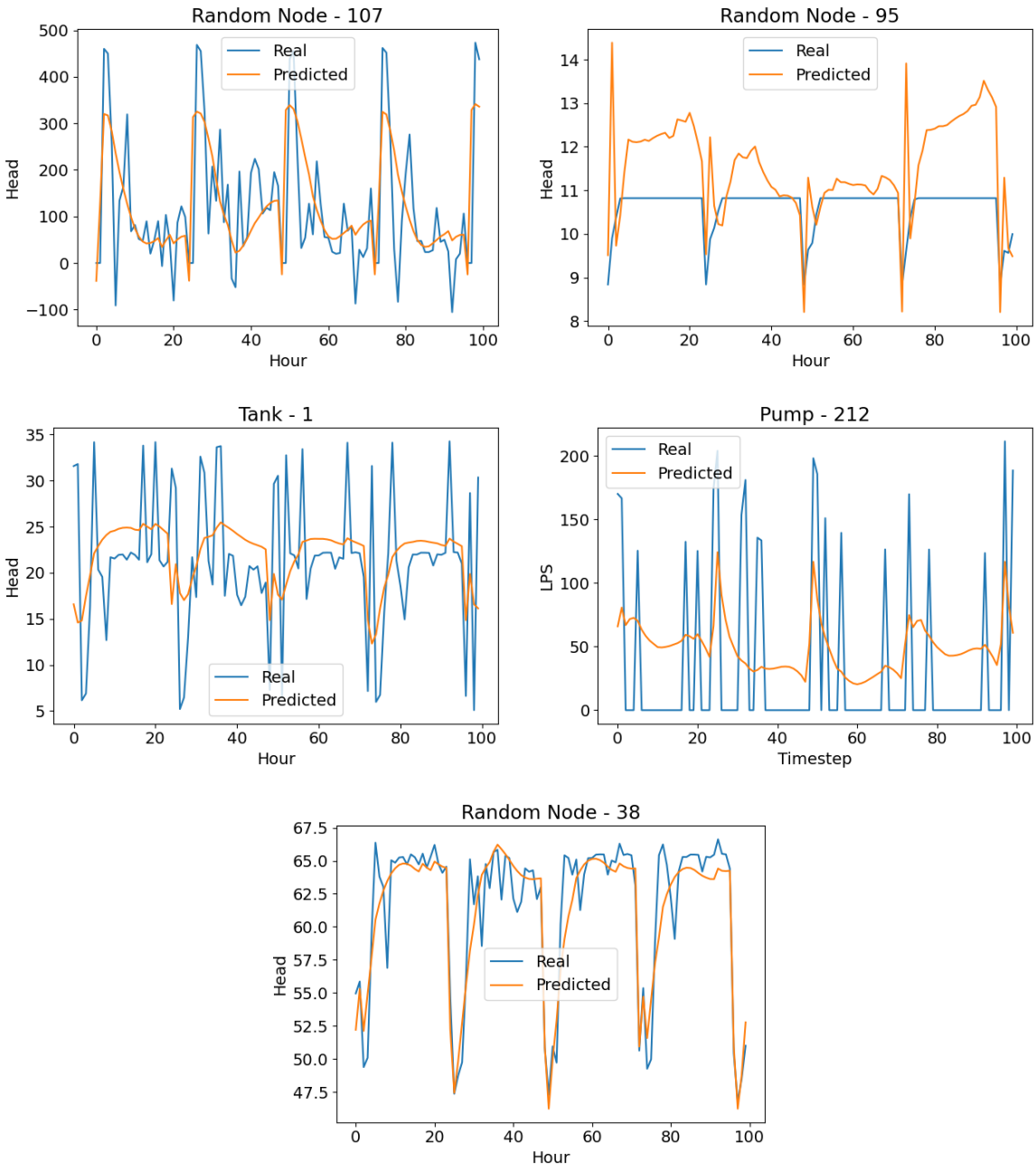


Figure A.13: All graphs produced after meta-modeling Net3 network

A.5. KY2 network and results

Figure A.14: An overview of the KY2 network

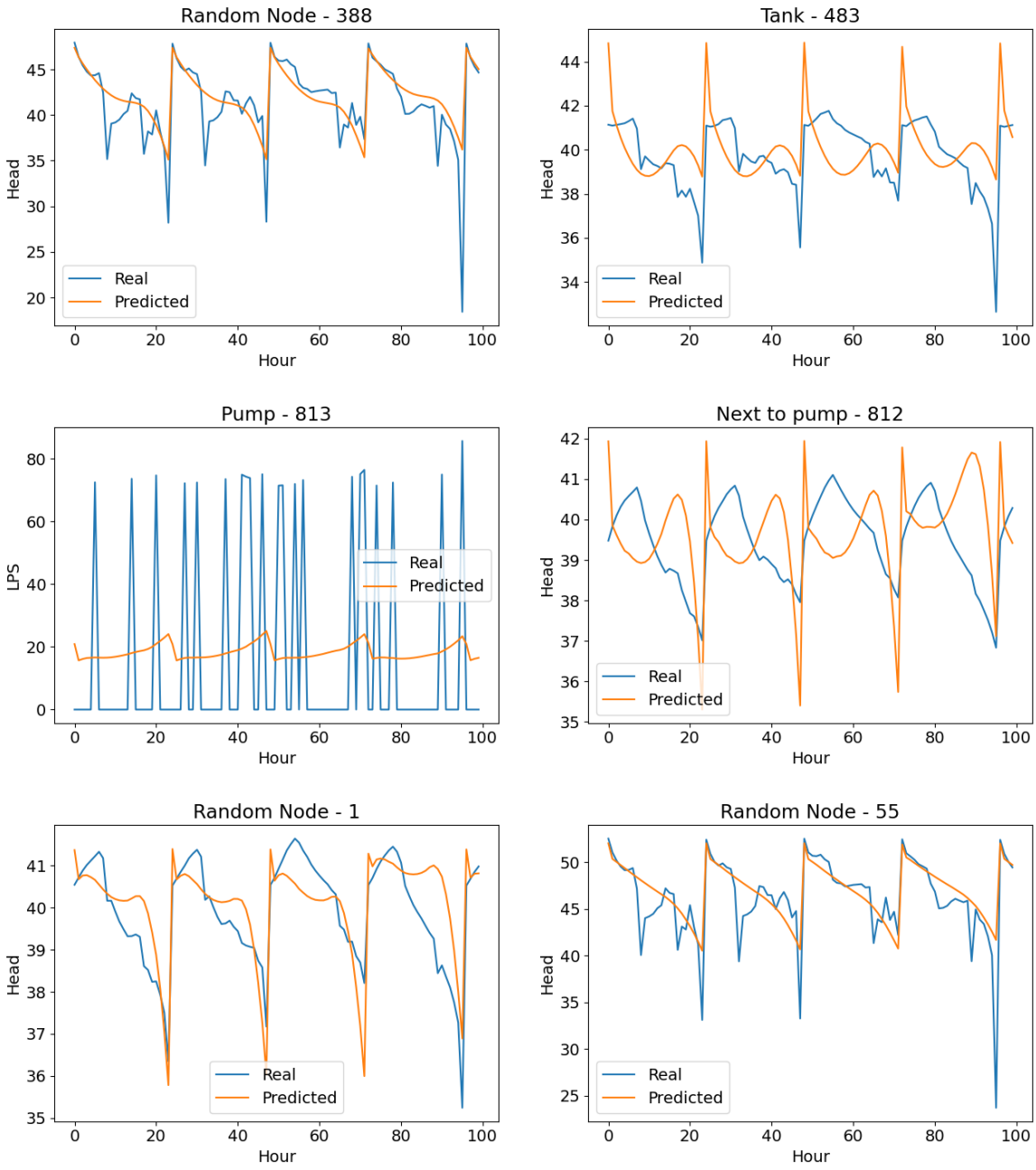


Figure A.15: All graphs produced after meta-modelling KY2 network (1)

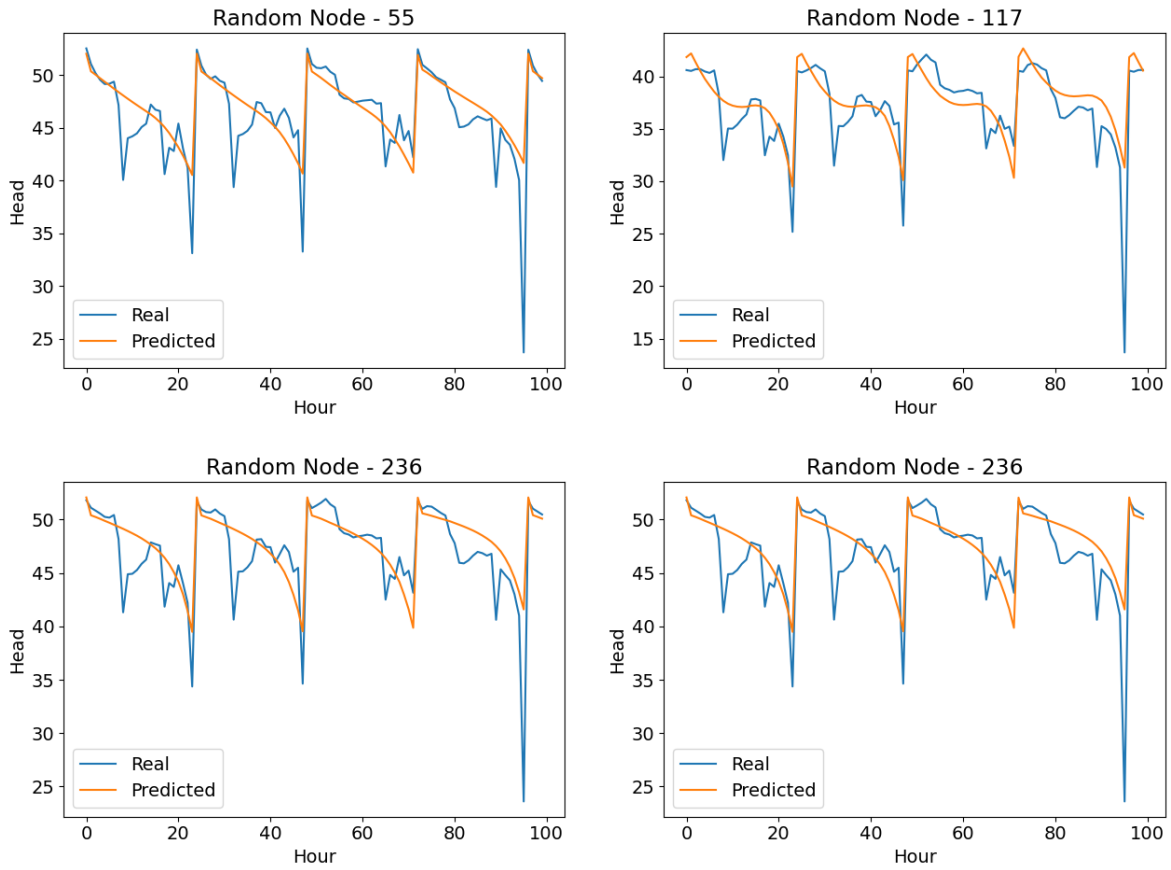


Figure A.16: All graphs produced after meta-modelling KY2 network (2)