Universität Bielefeld

Technische Fakultät

1st AI for Drinking Water Chlorination Challenge IJCAI 2025

Challenge Report

Authors:
Christoph Kowalski,
Jonas Schilin

Submission Date:
August 3rd, 2025

Contents

1 Introduction

In the context of the 1st AI for Drinking Water Chlorination Challenge at IJCAI 2025 (Artelt et al., 2025), we are investigating various methods to regulate the chlorine content in water distribution systems under different operational conditions. The challenge uses the CY-DBP water distribution system model (Pavlou et al., 2024), a realistic simulation based on a municipality in Cyprus, covering approximately 31.2 km of pipes and serving around 12,000 residents.
The simulation leverages EPANET and EPANET-MSX, two libraries which are able to model multi-species chemical reactions, including the formation of chlorine by-products such as trihalomethanes (THMs) and haloacetic acids (HAAs), under varying operational conditions and uncertainties.
The simulation simulates the normal water consumption of a municipality in Cyprus and its strain on the distribution system over a total period of one year in time steps of 5 minutes, taking into account daily, weekly and annual cycles, i.e. seasonal changes. In addition, various scenarios are simulated in which incidents occur, such as leaks in the pipes, resulting in sudden loss of water and pressure or sudden contamination events of the water in parts of the network. Monitoring of the system is additionally restricted, as the regulation system only has a total of 19 different sensors in the network. 17 of these sensors measure the chlorine concentration in the water and another two sensors measure the flow through the system. The system can be controlled via a total of five different chlorine injection pumps which the regulating system can use to add more chlorine to the network at the five separate points.
The primary goal of the challenge is to develop AI-based control mechanisms that are adaptive and resilient in real-world contamination scenarios.
This goal is expressed through a total of five different metrics by which the regulatory systems are evaluated: the total amount of chlorine injected, which represents the cost of control; compliance with restrictions on the operation of the chlorine injection pump, such as the maximum injection amount and the rate of change of the injection amount; local compliance with predetermined limits on chlorine concentration; the risk of infection from consumption of the water due to too -low or too-high amounts of chlorine; and finally, the spatial variations of the two aforementioned metrics to evaluate a particular notion of fairness.
In the course of this challenge, we tested a total of three fundamentally different approaches: a PPO agent (Schulman et al., 2017), an MPC (Mayne et al., 2000) and a rule-based system.
This report further outlines the approaches we considered, the challenges we encountered, the results we achieved, and the conclusions we have drawn from the work completed.

2 Related work

For the challenge, a total of three completely different approaches for the regulation systems were tested, each of which was optimized using different methods. All of the implemented approaches use a state predictive model based on an LSTM to predict boundary violations even before they occur.
The large number of different approaches is due to the interest of the authors in experimenting with different approaches from different disciplines. This, however, has the drawback of none of the approaches being fully optimized, due to the very limited time of the project.

2.1 State Predictive Models

One of the major challenges of chlorine injection control is the time delay between injection and arrival at the nodes in the network. Therefore, predicting the chlorine concentration in advance to inject chlorine even before the concentration is outside of the boundaries is crucial. To this

aim, a Recurrent Neural Network (Elman, 1990) can be trained as it is especially suited to cover the temporal dependencies present in a water distribution network. Due to the limited space in this report and the vast amount of literature on RNNs, this report does not dive deeper into the basic functioning of RNNs.

2.1.1 Long Short-Term Memory (LSTM) and Bidirectional LSTM (BiLSTM) Networks

Standard RNNs struggle with learning long-range dependencies due to vanishing or exploding gradients during training.
LSTM (Hochreiter and Schmidhuber, 1997) networks mitigate this problem by introducing gated memory cells that allow the model to selectively retain or forget information across time steps. This enables LSTMs to capture both short-term and long-term dependencies more effectively.
A BiLSTM (Graves et al., 2005) extends standard LSTMs by processing the input sequence in both forward and backward directions. Formally, for an input sequence $(x_1, x_2, \ldots, x_T)$, a BiLSTM computes two separate hidden sequences:

- A forward pass: $\overrightarrow{h}_t = \text{LSTM}_{\text{fwd}}(x_t, \overrightarrow{h}_{t-1})$
- A backward pass: $\overleftarrow{h}_t = \text{LSTM}_{\text{bwd}}(x_t, \overleftarrow{h}_{t+1})$

The final hidden representation at each time step is typically the concatenation $h_t = [\overrightarrow{h}_t; \overleftarrow{h}_t]$, allowing the model to incorporate both past and future context. This is especially beneficial in domains such as time series forecasting or state prediction in control systems, where bidirectional context improves prediction accuracy.

2.1.2 Modell distilation

For time-sensitive applications such as the control of a water distribution system, it is necessary to process incoming information quickly in order to be able to reach a decision in real time. The same applies to the optimization of RL or MPC approaches, where a large state prediction model costs a lot of time that is not actively invested in improvement. This is why one wants to avoid large neural networks as much as possible, both during training and at test time. At the same time, one does not want to sacrifice the power and robustness of large models. Therefore, a process called model distilation (Hinton et al., 2015) is used. This involves using two models, a large complex model that can develop complex structures in the feature space in order to process data optimally, called the teacher model, and a small lighter model that has fewer weights called the student model. The latter can learn directly from the activations in the layers of the large model and thus exploit the feature space of the large model.

2.2 First approach: Proximal Policy Optimization (PPO)

PPO is a policy-gradient method in reinforcement learning (RL) that balances performance improvement with policy stability. Like other RL approaches, PPO operates in the framework of a Markov Decision Process, defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where:

- $\mathcal{S}$ is the set of environment states,
- $\mathcal{A}$ is the set of actions,
- $P(s'|s, a)$ defines the state transition dynamics,
- $r(s, a)$ is the reward function,

- $\gamma \in [0, 1]$ is the discount factor.

The goal is to learn a stochastic policy $\pi_\theta(a|s)$, parameterized by θ, that maximizes the expected discounted return:

$$J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right] \tag{1}$$

PPO improves policy stability by modifying the vanilla policy gradient loss with a clipped surrogate objective:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \tag{2}$$

where $r_t(\theta) = \dfrac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio between new and old policies, $\hat{A}_t$ is an estimator of the advantage function, and ϵ is a small trust-region parameter (typically 0.1–0.3). This clipping prevents the policy from changing too abruptly in a single update, ensuring stable learning.
PPO was chosen for this project as it is comparably stable during training and well tested for a broad array of reinforcement learning tasks. Additionally, an implementation in stable-baselines-3 is available, and also an adaptation that uses an LSTM to better catch the temporal dependencies exists as an implementation compatible with stable-baselines-3.

2.3 Second approach: Model Predictive Control (MPC)

MPC is a control strategy that uses a predictive model of the system to solve an optimization problem over a finite time horizon. At each time step, the controller computes a sequence of control actions by minimizing a cost function, but only the first action is applied before re-optimizing at the next step.
Let x_t denote the system state at time t, and u_t the control input. The system dynamics are given by a model:

$$x_{t+1} = f(x_t, u_t) \tag{3}$$

MPC solves the following optimization problem at each step:

$$\begin{aligned} &\min_{u_{t:t+N-1}} \sum_{k=0}^{N-1} \ell(x_{t+k}, u_{t+k}) + \ell_f(x_{t+N}) \\ &\text{s.t.} \quad x_{t+k+1} = f(x_{t+k}, u_{t+k}), \quad x_t \text{ given}, \quad u_{t+k} \in \mathcal{U}, \quad x_{t+k} \in \mathcal{X} \end{aligned}$$

where ℓ is the stage cost, ℓ_f the terminal cost, and $\mathcal{U}$, $\mathcal{X}$ denote input and state constraints, respectively.
MPC has a long history in water-network regulation because it explicitly handles multi-input multi-output dynamics and hard bounds like chlorine concentration and pump rates (Elsherif et al., 2024; Wang et al., 2021). Building on this standing tradition, a MPC approach seems obvious.

2.4 Third approach: Rule-based approach

Even before the emergence of large-scale machine learning systems, complex water networks were already handled either by human operators or rule-based control policies. Especially when computational resources are limited, a simple rule-based approach might already give quite good results compared to very challenging to train reinforcement learning algorithms.

3 Methodology

3.1 State Predictive Model

As described above, predicting the next states is crucial to react in time to boundary violations, as the chlorine does not arrive at all nodes instantly after injection. Therefore, an LSTM was trained to predict the next timesteps, while looking at the timestamp, concentrations, flows, and actions of the last hour. In order to train the model, every simulation was run 10 times with a random injection pattern. The samples were divided into a train and test split, and mean squared error (MSE) was used to evaluate model performance. The test loss decreased to 0.00006 within the first 80 epochs and did not improve anymore from there. Additionally, plotting the prediction and matching it to the actual values confirmed the good performance of the model, as can be seen in fig. 1.
Given that full time-series data is available during training, we are not limited to strictly unidirectional input data. To better exploit the temporal dependencies in both directions, we trained a BiLSTM as an enhancement to the unidirectional LSTM used in earlier experiments.
For training the BiLSTM, the architecture and hyperparameters were kept consistent with the unidirectional setup. However, during training, each input sequence included not only the past but also future data points relative to the prediction target.
Now, this model is not really usable at test time, as there are no future observations at test time. However, we can take advantage of the more complex feature space of the BiLSTM by using it as a teacher model for an LSTM student model. Compared to the large LSTM, the student model is much more lightweight without sacrificing too much precision, visible in fig. 2. Similar to the bigger models, the student model also shows fast convergence, as visible in fig. 3.
As this project focuses on the different control policies, no further finetuning is applied to the state model predictor, but the student model is used for the control policies.

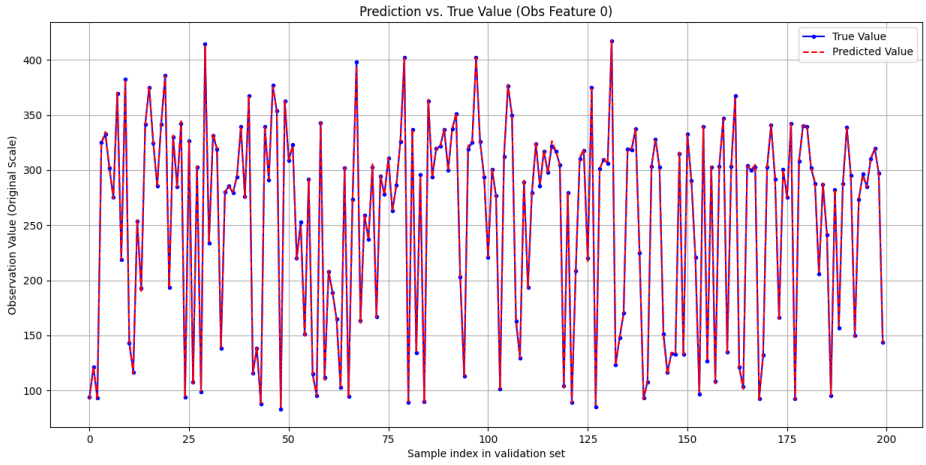

Figure 1: State Predictive model prediction vs. real value for random injection policy for flow.

3.2 Reinforcement learning

One of the most important aspects of reinforcement learning is a good reward function. If there are loopholes, the agent might exploit them to maximize the score without actually learning the intended policy. For example, in the domain of water distribution networks, a high reward for low fluctuations in injection might cause the agent to simply never inject chlorine, as this maximizes

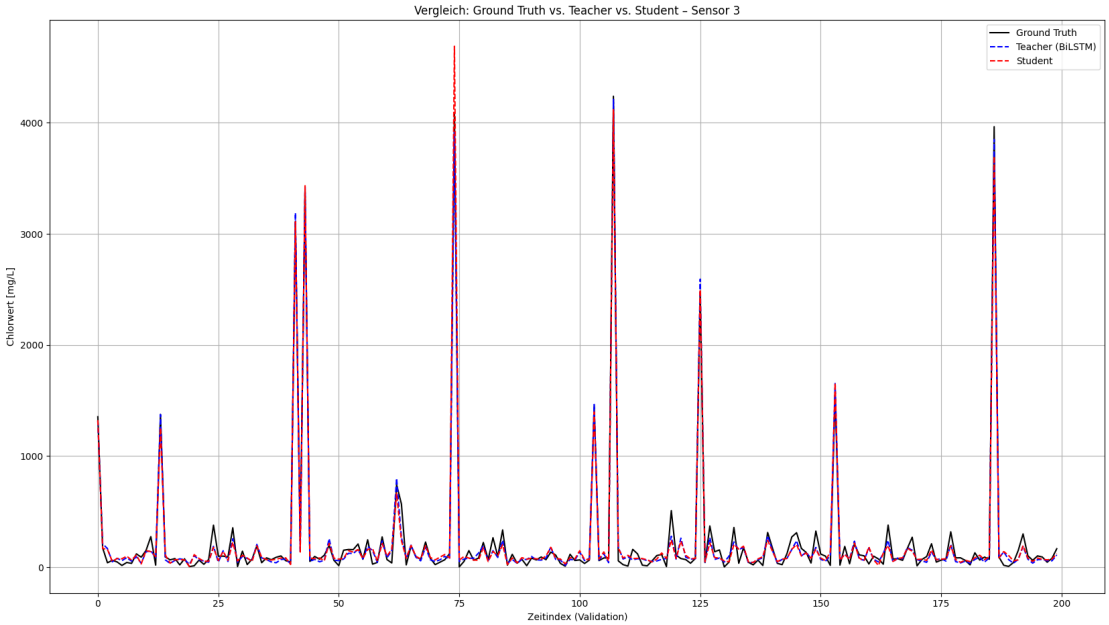

Figure 2: BiLSTM vs. Student Model vs. ground truth prediction for random injection policy for chlorine concentration at node 3.

the reward, although this is not the intended policy. As combining sub-goals in one reward function needs careful crafting of the reward function, our first reward function focuses on one goal only.
In the case of chlorination control in a drinking water network, ensuring that the chlorine concentration is within safe boundaries is the single most important objective. Therefore, a reward function, visible in fig. 4, was designed that encourages the agent to ensure a chlorine concentration within the safe boundaries. The key characteristic of the reward function is the different gradient outside and inside the safe boundaries. The reward function has a shallow gradient if the chlorine level is inside the safe boundaries but not at the optimal level of 0.3. Outside of the safe boundaries, the gradient is steeper to encourage the agent to optimize those chlorine levels before trying to reach the optimal value for nodes that are already within the safe boundaries. It is important to highlight that the agent gets an increased reward if the chlorine concentration gets closer to the safe boundaries, but is still outside of them. This is an important difference from a step function, as this would only give a positive incentive as soon as the safe boundaries are reached.
Further refinements of the reward function were planned; however, due to the bad performance of the RL approach, as will be discussed in the next section, and the limited scope of this project, no further refinements were made, but other approaches were developed.

3.3 MPC

The MPC approach uses a trained student LSTM model to simulate and evaluate future chlorine dynamics in the network. At each decision step, the controller generates a sequence of chlorine injection actions across all booster stations, with the goal of maintaining safe chlorine levels throughout the network while minimizing chemical usage.
The MPC agent performs the following steps at every control interval:

1. It collects the most recent 12 timesteps of observations to build an input window.

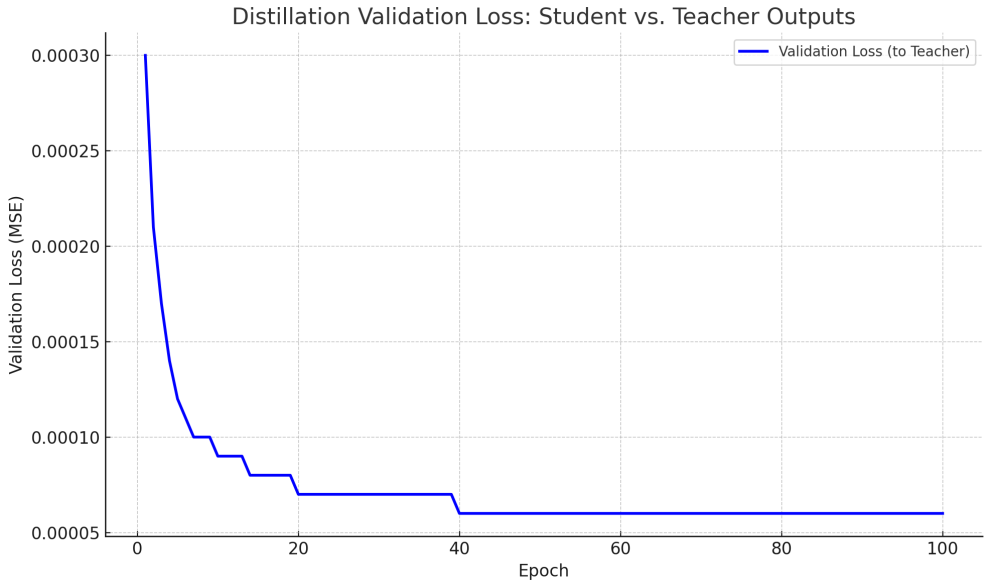

Figure 3: Validation loss (MSE) between student and teacher model outputs during distillation training. Lower is better imitation of teacher predictions.

2. It uses this window to roll out chlorine concentration predictions over a 15-step horizon (equivalent to 1.25 hours).

3. It formulates an optimization problem over the planned injection values at each of the pumps over the horizon.

The optimization minimizes a cost function that combines two terms:

- The mean squared error between the predicted chlorine concentration and a target value of 0.3 mg/L, with strong penalties for predicted values outside the defined safe range of $[0.2, 0.4]$ mg/L.
- A regularization term that penalizes the total chlorine injected, promoting more efficient chemical use.

This objective function is optimized using the L-BFGS-B algorithm from scipy.optimize, which supports box constraints on the injection range at each pump (i.e., 0 to 1 mg/L). After optimization, only the first action in the optimized sequence is executed, and the process is repeated at the next time step in a receding-horizon fashion.
Although the entire MPC pipeline was successfully implemented, including integration with the student LSTM model and the optimization procedure, the approach proved computationally infeasible within the constraints of this project. The need to simulate hundreds of LSTM predictions per step and optimize over a large continuous action space resulted in excessively long runtimes. As a result, we were unable to run full episodes or collect evaluation metrics within the available project time frame.
Nevertheless, the MPC framework is functionally complete and could be used in future work, potentially with performance improvements such as reducing the control horizon, simplifying the action space, or approximating the optimization via sampling or learning-based surrogates.

3.4 Rule-based approach

The rule-based approach uses simple rules to control the chlorine injection at the different booster stations. Firstly, stable injection patterns that can be adapted if the chlorine concentration is

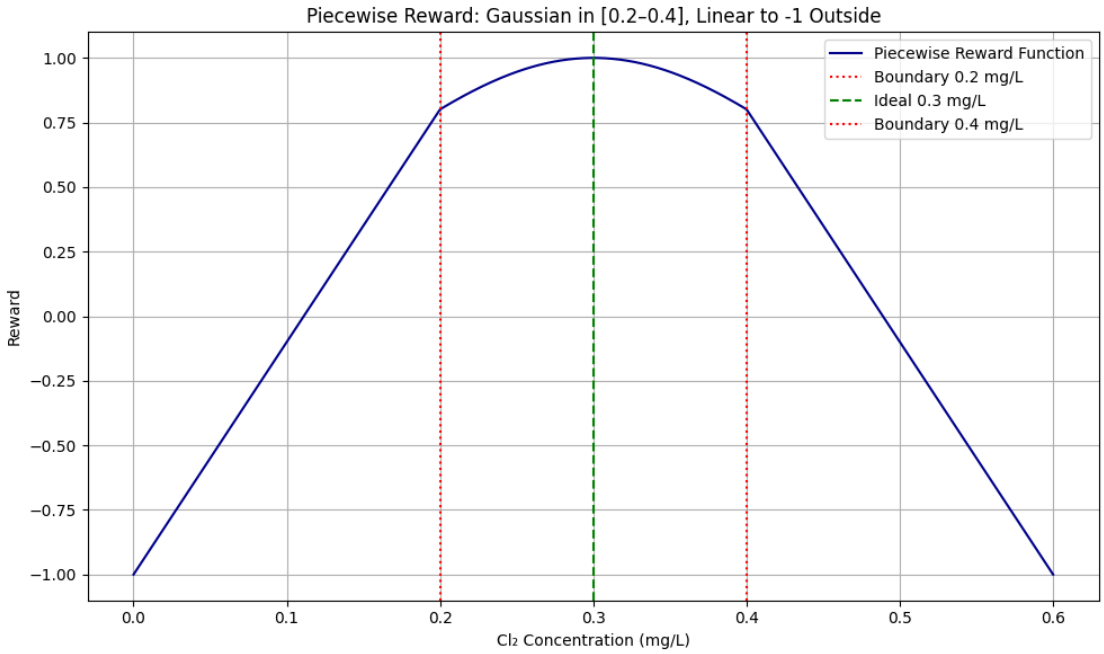

Figure 4: Adapted reward function to avoid boundary violations.

outside of the boundaries need to be established.
Secondly, a sensitivity analysis to determine which pumps have the greatest impact on which sensors needs to be conducted. This determines which injection rates should be adapted to increase or decrease the chlorine concentration at the specific nodes.
Thirdly, the state predictive model is integrated to allow reacting to predicted boundary violations before they occur. This is crucial as the chlorine needs time to travel through the network, and therefore, injecting more or less chlorine does not have immediate effects on the chlorine level at all nodes.
Fourthly, the flow information can be considered as the chlorine demand increases with an increasing water flow, and water consumption varies in the course of one day.

4 Experiments

The evaluation metric chosen is the boundary violation section of the evaluation script. As described above, the project wants to achieve several goals; however, keeping the chlorine concentration within safe boundaries is the most important. As our first experiments showed that the problem is not trivial, all other sub-goals are disregarded, as a policy that keeps chlorine levels within the safe boundaries is essential.

4.1 Reinforcement learning

The reinforcement learning agent was trained on the first six-day scenario and the first 18 days of the one-year simulation. Of the six-day scenario, only three days are available to the simulation, as the first three days are used to populate the network.
This restricted choice of simulations was necessary to balance computational time and task complexity and to establish a proof of concept on a small subset of simulations, as more general policies are also harder to train.
As described above, the agents' reward function focused on boundary violations only, and the estimate was a training time of at least 1-2 million steps. However, due to the very high run time

of the simulation, it was not possible to train an agent for that long.
Several different experiments have been conducted to test the effect of the predictor network and varying hyperparameters. First, a simple PPO algorithm was trained with our custom reward function to establish a baseline.
Second, the output of the predictor network, i.e., all the flow and chlorine concentrations over the next fifteen timesteps, was given to the agent combined with the current observation. This greatly increased the size of the input space from 19 to over 300 and slowed down training even more.
Third, the output of the predictor network was transformed to only indicate a trend over the predicted timesteps by fitting a linear function through each sensor's prediction. This preserved valuable information about which actions might be necessary, but decreased the input space down to less than 40 features.
Fourth, the key hyperparameters to achieve training stability were adapted in order to see whether small training successes could be achieved.
As visible in fig. 7, the model did not learn anything meaningful, as the training reward fluctuated and even decreased over time. For all the experiments conducted, there was no improvement in the training reward, and therefore, the RL approach was not able to solve the problem within the available time of the project and with the available computational resources. Additionally, it is not possible to evaluate the effect of the predictor network as no training improvements were observable for any architecture. It must be emphasized again that complex RL problems like this normally need millions of steps, which was not feasible in the current setting.
However, when applying the longest trained PPO with the predictive state model using the evaluation script on scenario 1, the bound_violations score was at 0.185, which is considerably better than the random policy, however, worse than a constant injection pattern, as visible in the next section. Therefore, an RL agent might be able to solve the problem with longer training times.
As the models became too big to train and test locally, training and evaluation were conducted on the GPU cluster, which sometimes caused considerable wait times, as working on the study partition only, limiting the presented results.

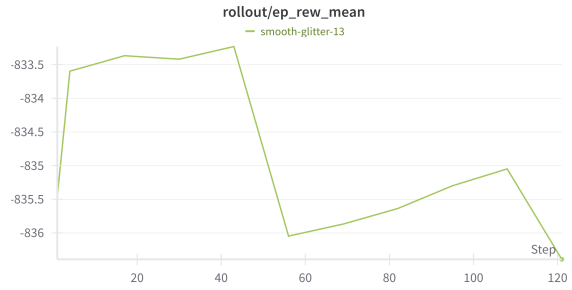

Figure 5: (a) Enhanced model training reward scenario 0

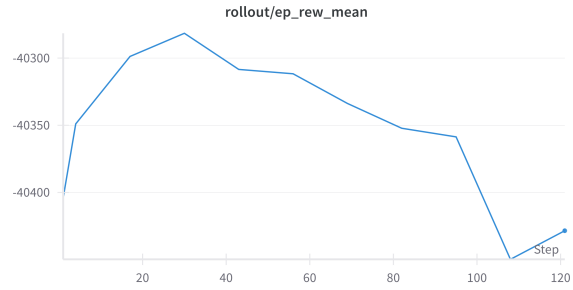

Figure 6: (b)Enhanced model training reward on first 18 days of one-year simulation

Figure 7: Comparison of training rewards between models with LSTM-based chlorine prediction on different simulations. Charts created by Weights & Biases (Biewald, 2020). Each step corresponds to one update over 4096 steps, totaling around 500k steps.

4.2 Rule-based Approach

Several experiments were conducted to test and optimize a hand-crafted rule-based policy.
Firstly, it is necessary to find a good, stable injection pattern. Therefore, we iterated over many possible injection values that all injection points share and that optimize the boundary violation

criteria. For several different scenarios, a constant injection of 12.5 mg/L was found to be the best. However, as the injection pumps are located at different points in the network, differing injection rates might be optimal for the individual pumps. Therefore, starting from those baseline values, a grid search is conducted that varies the injection values of the different pumps. If an injection value was adapted, all other pumps were evaluated again with differing values until no changes were made anymore. This resulted in a great increase in injection at the first booster station to 250 mg/L. This high value fits the network architecture as the first booster station is at the beginning of the network, and therefore the chlorine injected there travels through the whole network, while the other booster stations only supply chlorine to small parts of the network.
Secondly, after finding good stable injection values, it is necessary to adapt chlorine injection in case of certain nodes being outside of the safe boundaries. To assess which pumps need to increase or decrease chlorine injection, a sensitivity analysis was conducted with the goal of determining the two most influential pumps for each sensor. Therefore, the chlorine injection of all pumps was set to 0 mg/L, and one pump at a time injected 250 mg/L of chlorine, and the change in chlorine levels at all nodes was measured. A heatmap with the sensitive analysis can be found in fig. 8. Drawing from this analysis, a mapping can be created that maps every sensor node to the two most influential pumps whose chlorine injection can be adjusted if the chlorine concentration is outside of the boundaries.
Experiments were conducted with simple if-else rules, decreasing or increasing the injection of chlorine by a constant factor delta at the corresponding pumps if a node is outside of the safe boundaries. A search for the optimal delta, however, showed that 0.0 is the optimal value, making the rules unused. This might be due to the very late reaction to boundary violations, as the chlorine needs time to travel through the network, and a change in water flow might have bigger effects on chlorine concentration than the badly timed reaction to boundary violations. Therefore, the predictor network was used to estimate the chlorine level in the future and react in advance. However, a delta of 0.0 still achieves the best results, as visible in fig. 9, indicating that more careful rules on when and how to inject chlorine at the corresponding nodes are necessary. However, due to the very limited scope of the project, this needs to be discussed in future work on this project.

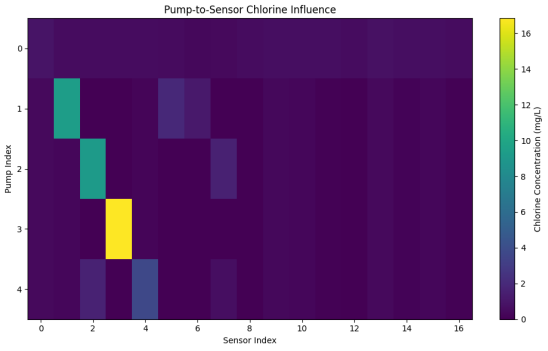

Figure 8: Heatmap of sensor sensitivity to pump chlorine injection

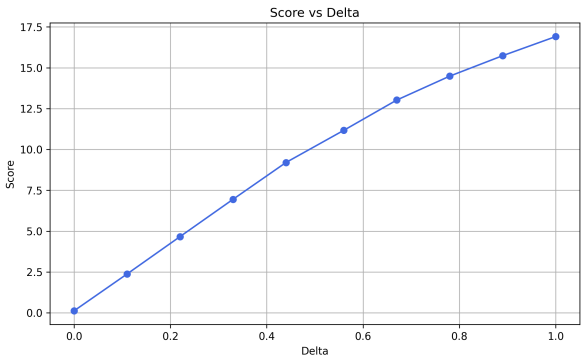

Figure 9: Boundary violation score in relation to the chosen delta when using the lookahead networks.

Figure 10: Evaluation of rule-based approach

The environment observation, however, not only gives the chlorine concentration at selected nodes but also the water flow information at two positions, most notably the flow close to the reservoir. This can be used as the chlorine demand is highly dependent on the amount of new water flowing into the network. Over a day, the water demand differs greatly, as during the day the water demand is higher and therefore the flow increases greatly, causing an increased demand for chlorine

injection. Correlating the chlorine injection of the first booster station with the measured water flow decreases the boundary violations from 0.135 to 0.129 for the first scenario and from 0.13 to 0.126 for the first 18 days of the long scenario. A random control policy achieves a boundary violation value of greater than 16 in both simulations.
While this project was able to increase the performance of the control policy by finding good injection rates for the pumps, it must be noted here that the adaptation of the chlorine injection when boundary violations are detected is quite complex, and therefore, further research must experiment with different adaptations of the injection pattern if such violations are detected.

5 Conclusion

This project experiments with several policies to control chlorine injection for a water distribution network. To that end, three fundamentally different approaches were implemented, and experiments were conducted. First, a PPO-based RL agent is trained, second, a MCP model is implemented, and third, a rule-based policy is crafted. All of these approaches are supplemented with an LSTM-based state predictive model to predict the chlorine concentration at different sensors in the future. This was necessary, as the chlorine needs time to travel through the network, and therefore it needs to be injected before the boundary violation occurs.
Different experiments and training of the RL agent showed that the available computational resources in combination with the slow runtime of the simulation were not sufficient to properly train the agent within the scope of this project. No variation of the RL agent was able to learn anything meaningful, becoming visible in a non-improving train and validation reward. Therefore, it is not possible to conduct any analysis about the impact of the state predictive model or the effect of hyperparameter tuning.
The MPC method has a similar problem: due to the minimum required foresight, the MPC relies on extensive calculations by the state predictive model, and therefore also the simulation environment, to make its decisions. During trials, it became clear that it could not be used effectively with our available resources. The reason for this is that the model often entered local minima at a very early stage without any recognizable improvement. But it was difficult to make iterative improvements to the procedure, as even a simulation of just a few simulation days takes several days of trial time. Nevertheless, the MPC approach seems to be very well suited to the problem, as the metrics from the competition rules can be translated directly into constraints for the MPC, whilst the problem is to drastically reduce the computing time.
Possible approaches would be the discretization of the action space, which then allows gridsearch-like optimizations over a much smaller action space. Dynamic programming approaches to save decisions for the same situations and thus save computing time or to operate search space pruning by directly abandoning non-optimal solutions. Other approaches that were considered were parallelizations such as individual MPCs for each pump. However, these concepts are all very sophisticated and unfortunately not realistic within the scope of the challenge.
The rule-based approach started by exploring the best stable injection patterns. Afterwards, a sensitivity analysis established which chlorine pumps have the greatest impact on which chlorine sensor, and simple rules were implemented on when to increase or decrease chlorine injection at the corresponding injection pumps. Searching for the optimal rate of injection change showed that simply injecting the base pattern performs better than the rules implemented. Therefore, the state predictive model was added to allow for injection changes even before bound violations happen. This, however, also did not show an increase in performance. Therefore, it must be noted that the adaptation of the injection pattern if a boundary violation is predicted or detected is quite challenging and therefore needs further research.
In general, this project implemented and experimented with many different approaches to evaluate the differences instead of focusing on only one approach and optimizing this for best performance.

Statement of Contributions

Conceptualization of the reward function for the RL agent and the training of the State Predictive Model were done together. While Christoph focused a bit more on the RL agent and the rule-based approach, Jonas focused on the MPC. The report was written together. In general, close communication during all phases of the project resulted in an equal participation of both team members, and therefore, a common grade would be appreciated.

References

Artelt, A., Hermes, L., Strotherm, J., Hammer, B., Vrachimis, S. G., Kyriakou, M. S., Eliades, D. G., Polycarpou, M. M., Paraskevopoulos, S., Vrochidis, S., Vrochidis, S., Taormina, R., Savic, D., & Koundouri, P. (2025). *1st AI for Drinking Water Chlorination Challenge* (tech. rep.). 34th International Joint Conference on Artificial Intelligence (IJCAI).

Biewald, L. (2020). *Experiment Tracking with Weights and Biases* (tech. rep.). Weights & Biases.

Elman, J. L. (1990). Finding Structure in Time. https://doi.org/doi.org/10.1016/0364-0213(90)90002-E

Elsherif, S. M., Taha, A. F., & Abokifa, A. A. (2024, September). Disinfectant Control in Drinking Water Networks: Integrating Advection-Dispersion-Reaction Models and Byproduct Constraints. https://doi.org/10.48550/arXiv.2409.08157

Graves, A., Fernández, S., & Schmidhuber, J. (2005). Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, W. Duch, J. Kacprzyk, E. Oja, & S. Zadrożny (Eds.), *Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005* (pp. 799–804, Vol. 3697). Springer Berlin Heidelberg. https://doi.org/10.1007/11550907_126

Hinton, G., Vinyals, O., & Dean, J. (2015, March). Distilling the Knowledge in a Neural Network. https://doi.org/10.48550/arXiv.1503.02531

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780.

Mayne, D., Rawlings, J., Rao, C., & Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*(6), 789–814. https://doi.org/10.1016/S0005-1098(99)00214-9

Pavlou, P., Kyriakou, M., Vrachimis, S. G., & Eliades, D. G. (2024). A Comprehensive Virtual Testbed for Modeling Disinfection Byproduct Formation in Water Distribution Networks. *Engineering Proceedings*, *69*(1), 33. https://doi.org/10.3390/engproc2024069033

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017, August). Proximal Policy Optimization Algorithms. https://doi.org/10.48550/arXiv.1707.06347

Wang, S., Taha, A. F., & Abokifa, A. A. (2021). How Effective is Model Predictive Control in Real-Time Water Quality Regulation? State-Space Modeling and Scalable Control. *Water Resources Research*, *57*(5), e2020WR027771. https://doi.org/10.1029/2020WR027771