Deep learning based simulators for the phosphorus removal process control in wastewater treatment via deep reinforcement learning algorithms

Esmaeel Mohammadi [a,b,*], Mikkel Stokholm-Bjerregaard [a], Aviaja Anna Hansen [a], Per Halkjær Nielsen [b], Daniel Ortiz-Arroyo [c], Petar Durdevic [c]

[a] Krüger A/S, Indkildevej 6C, Aalborg, 9210, North Jutland, Denmark
[b] Department of Chemistry and Bioscience, Aalborg University, Fredrik Bajers Vej 7H, Aalborg, 9220, North Jutland, Denmark
[c] AAU Energy, Aalborg University, Niels Bohrs vej 8, Esbjerg, 6700, South Jutland, Denmark

ARTICLE INFO

ABSTRACT

Phosphorus removal is vital in wastewater treatment to reduce reliance on limited resources. Deep reinforcement learning (DRL) can be used to optimize the processes in wastewater treatment plants by learning control policies through trial and error. However, applying DRL to chemical and biological processes is challenging due to the need for accurate simulators. This study trained six models to identify the phosphorus removal process and used them to create a simulator for the DRL environment. While achieving high accuracy (>97%) in one-step ahead prediction of the test dataset, these models struggled as simulators over longer horizons, showing uncertainty and incorrect predictions when using their own outputs for multi-step simulations. Compounding errors in the models' predictions were identified as one of the causes of this problem. This approach for improving process control involves creating simulation environments for DRL algorithms, using data from supervisory control and data acquisition (SCADA) systems with a sufficient historical horizon without complex system modeling or parameter estimation.

1. Introduction

Phosphorus (P) is essential for human nutrition and plant growth. Without it, the primary cells of plants, animals, and life would not exist (Nobaharan et al., 2021; Melgaço et al., 2021). The amount of available P is limited due to the decreasing number of phosphate rock resources (Porter and FitzSimons, 2009). The phosphorus nutrient is often found in wastewater treatment plants (WWTP) outlets in surface waters; therefore, high P concentrations lead to eutrophication that can affect the environment and human health (Welch and Lindell, 1980). Efficient phosphorus removal from wastewater prevents eutrophication and its consequences (Gu et al., 2021). It is possible to remove phosphorus from wastewater by incorporating phosphate into Total Suspended Solids (TSS) and subsequently removing it from them. A P-containing bio-solid (microorganisms, for example) or chemical precipitate can be formed (Tchobanoglous et al., 2003). Removal and recovery of P can be promoted by analyzing its dynamics through the wastewater treatment process, which helps engineers and operators comprehensively understand what is happening and identify potential future problems (Hansen et al., 2022). The dynamic control system measures and controls various parameters of the wastewater treatment process, such as pH, dissolved oxygen, temperature, and flow rate.

With real-time monitoring of these parameters, operators can optimize treatment efficiency, reduce energy consumption, and minimize waste production and byproduct generation (Revollar et al., 2017). The enhanced Biological phosphorus Removal (EBPR) process, which takes advantage of phosphorus accumulating organisms (PAO), is a more environmental-friendly and cost-effective way of phosphorus treatment rather than chemical removal methods which use the addition of metal salts for P precipitation (Acevedo et al., 2014). Combining chemical and biological processes can achieve a desirable level of P removal with lower expenses. Therefore, most plants nowadays use a hybrid P removal system (Bunce et al., 2018).

Control of the phosphorus removal systems in wastewater treatment plants is challenging in current literature because, on the one hand, variables like dissolved oxygen (DO) are not the best indicators of the aerobic zone. On the other hand, there are uncertainties in incoming phosphate concentration and the influence of process conditions (pH, temperature, etc.) on the coagulation process (Chong et al., 2013; Seviour et al., 2003). Although there is not a wide range of studies on P removal control strategies in the literature due to the mentioned issues, some researchers applied control methods like fuzzy control (Xu and Vilanova, 2015), Model Predictive Control (MPC) (Ostace et al., 2013),

---

and the Supervisory and Override Control Approach (SOPCA) (Sheik et al., 2022) in the framework of benchmark simulation models (BSM) developed by International Water Association (IWA) (Alex et al., 2008; Henze et al., 2006). Benchmark simulation models are standard generalized tools for the simulation of biological processes in WWTPs. Still, they are also not a good representative of the process dynamics, especially P removal, which is highly stochastic and unpredictable. Novel control strategies such as Deep Reinforcement Learning (DRL) algorithms have been recently introduced to overcome the complexity, uncertainties, and challenges of process control for different biological systems (Chen et al., 2021).

Reinforcement Learning (RL) is a machine learning technique that involves training an agent to make decisions based on rewards received from an environment. The agent learns to maximize rewards by taking actions that lead to positive outcomes and avoiding actions that lead to adverse outcomes. Reinforcement learning has been successfully applied in various applications, including game playing (Silver et al., 2017; Vinyals et al., 2019), robotics (Hua et al., 2021), and control systems (Sutton and Barto, 2018; Moriyama et al., 2018). Deep reinforcement learning is a variation of reinforcement learning that uses deep neural networks to represent the agent's policy and value functions (Sutton and Barto, 2018). DRL can learn from raw sensory data, such as images, audio, or text, and handle high-dimensional inputs and outputs (Mnih et al., 2015). Traditional optimal control methods, such as model predictive control (MPC), face limitations when applied to large-scale stochastic multiple-input multiple-output (MIMO) problems due to their online computational requirements and assumptions about uncertainty information (Maravelias and Sung, 2009). On the other hand, DRL can pre-compute optimal solutions offline, reducing online computation time, and can be trained in a process simulator to acquire a general knowledge of the process (Nian et al., 2020). Although the DRL agent's performance may not surpass that of a corresponding MPC designed based on the same simulator model, DRL's learned optimal policy implicitly includes information about optimal set points and inputs, akin to the concept of economic MPC (Nian et al., 2020). As a result, DRL shows promise in the process control industry and has been used considerably in recent control research (Bao et al., 2021; Lillicrap et al., 2015; Moriyama et al., 2018; Raju et al., 2015; Spielberg et al., 2017). In the context of wastewater treatment, predicting key indicators such as phosphorus levels is crucial for environmental management. The dynamic nature of wastewater systems, influenced by numerous unpredictable factors, poses a significant challenge. DRL's adaptability and learning capabilities present a novel approach to accurately predicting these indicators, enabling more effective and sustainable wastewater management strategies.

In Nian et al. (2020), some of the shortcomings of reinforcement learning have been discussed for process control applications. In summary, it can be data inefficiency, scalability, stability, convergence, constraints, and accurate simulator. The lack of a precise simulator for most industrial process control applications has been the main issue for using RL methods (Nian et al., 2020). This problem also extends to chemical and biological processes, which led researchers to try implementing different strategies for creating a simulation environment to train and test RL algorithms in it (Spielberg et al., 2017; Wang et al., 2018; Nian et al., 2019). The application of DRL methods in wastewater treatment plants, especially for phosphorus removal processes, is very limited. In Pang et al. (2019), a Q-learning algorithm was used to optimize aerobic and anaerobic hydraulic retention time (HRT) for the biological phosphorus removal process. They used the ASM2d model to generate state transition matrices to train the Q-learning algorithm. The developed model was verified using data from a lab-scale sequencing batch reactor (SBR) with aerobic and anaerobic processes. A Multi-Agent Deep Deterministic Policy Gradient (MADDPG) was used in Chen et al. (2021) for the purpose of dissolved oxygen and chemical dosage control in a WWTP. They used the MANTIS model, an integrated form of three other mechanistic models (Chen et al., 2021) to simulate the target WWTP.

Modeling the wastewater treatment process to study the system behavior has been going on for decades, from earlier mechanistic models (Henze et al., 2000) to advanced data-driven approaches (Hansen et al., 2022; Newhart et al., 2019). Simple wastewater treatment models such as the Activated Sludge Model (ASM) and Anaerobic Digestion Model (ADM) have been used for years to study the dynamics of WWTPs (Gujer, 2006; Batstone et al., 2002). These models are no longer feasible and accurate enough to describe different processes in wastewater treatment plants (Burton et al., 2014). Activated Sludge Models No. 2 are mainly used for biological phosphorus removal modeling (Henze et al., 2000). Anaerobic Digestion Model No. 1 (ADM1) has also been used for studying struvite precipitation (Ikumi et al., 2011) and biodegradability of organics in anaerobic digestion (Ikumi et al., 2014).

With the advance of Artificial Intelligence (AI), modeling of WWTPs based on machine learning and deep learning methods has become very popular (Ye et al., 2020). AI methods can predict operational parameters, evaluate energy usage, fault diagnosis, automation, and intelligent control in WWTPs (Malviya and Jaspal, 2021; Zhao et al., 2020). Data-driven modeling of WWTPs has emerged as an alternative to mechanistic models, as the former does not necessitate a thorough comprehension of the plant's design and operation. Furthermore, data-driven models can be developed relatively swiftly and with fewer input data, as per the findings of Newhart et al. (2019). Artificial Neural Networks (ANN), including the Multi-layer Perceptron (MLP) network, have been demonstrated as a robust and precise technique for forecasting operational parameters in WWTPs (Mannina et al., 2019; Nelles, 2020; Wunsch et al., 2018; Pisa et al., 2019c). Data from WWTPs can be treated as time series. The auto-regressive integrated moving average (ARIMA) model is a time series prediction method that uses past data to predict future target values (Berthouex and Box, 1996). ARIMA has been used for studies like sedimentation modeling (Park and Koo, 2015) and water quality prediction (Ömer Faruk, 2010). A Recurrent Neural Network (RNN), an extended feed-forward Neural Network (FFNN), can be used for time series prediction. RNN can capture and pass information through its elements with the help of memory and takes advantage of past information for decision-making (Cheng and Zhao, 2019; Zhu et al., 2020). RNNs have been used in controlling activated sludge process (Foscoliano et al., 2016), forecasting the water flow of the WWTP (Zhang et al., 2018), predicting the amount of ammonium and total nitrogen (Pisa et al., 2019b), as a software sensor for prediction of BOD, COD, and TSS indexes (Chang and Li, 2021), and as a fuzzy controller for the dissolved oxygen and nitrate concentration in WWTP (Gaitang et al., 2016).

As a type of recurrent neural network, Long Short-term Memory (*LSTM*) networks can learn order dependence for sequence prediction (Sak et al., 2014). The two technical problems of conventional RNNs overcome by *LSTM*s are vanishing and exploding gradients related to how the network is trained (Graves et al., 2009). Considering the good results of implementing *LSTM*s in areas like speech recognition (Peng et al., 2021) and natural language processing (Shuang et al., 2020), it has been used in wastewater treatment studies recently. In the wastewater treatment literature, *LSTM*s have been implemented for different kinds of applications such as prediction of effluent quality (Pisa et al., 2019a), forecasting wastewater flow rate (Kang et al., 2020), estimation of ammonium, total nitrogen, and total phosphorus removal efficiency (Yaqub et al., 2020), predicting influent BOD, effluent BOD, temperature, and power efficiency (Cheng et al., 2020), and as a control strategy in WWTPs (Pisa et al., 2020). In Hansen et al. (2022), an *LSTM* model was implemented to predict phosphorus dynamics in wastewater treatment plants. They used Bayesian optimization for hyperparameter tuning of the model, which could predict phosphorus concentrations up to 24 h in the future. Additionally, recent advancements in time series

forecasting methodologies have significant implications for environmental engineering. Bayesian methods provide probabilistic modeling and linear regression that excels in handling data inconsistencies like time lags and abnormal signals (Zhang et al., 2024; Zhao et al., 2023b). Meanwhile, the Bidirectional Long Short-term Memory (*Bi-LSTM*) networks were investigated for intricate pattern detection (Zhao et al., 2023a). These methodologies mirror the complexities in phosphorus removal from wastewater, underscoring the relevance of advanced AI techniques, such as Deep Reinforcement Learning, for precise and efficient environmental process modeling.

As a result of Vaswani et al.'s *Transformer*, encoder–decoder models based on attention mechanism were generated (Vaswani et al., 2017). *Transformer*s have been used to forecast time series in environmental engineering areas such as defect detection inside the sewage system (Dang et al., 2022) and prediction of effluent water quality in WWTPs (Huang et al., 2021). Later, *Informer* (Zhou et al., 2021) was introduced in 2021 to overcome some problems of the vanilla *Transformer*, such as high memory usage and limitations of the encoder–decoder architecture for time-series forecasting. *Informer* could outperform existing time-series forecasting methods like ARIMA, *LSTM*, and *Transformer*s for the benchmark datasets of energy and weather (Zhou et al., 2021). *Autoformer* is a recent version of *Transformer*s introduced in Wu et al. (2021) to perform long-time series forecasting. Currently, no study is based on the *Informer* and *Autoformer* models for wastewater treatment applications. Still, considering that the input sequence length for wastewater parameters is uncertain, whether these approaches will perform well in WWTP modeling is a question. The abovementioned deep learning models can be used to simulate the DRL agent's environment. The model inputs the current state of the environment and outputs the next state and the reward signal for the agent.

The lack of an accurate simulator for implementing deep reinforcement learning algorithms in WWTPs motivated us to study state-of-the-art deep learning models to address this problem. For this purpose, we first extracted a dataset containing information on the P removal process from the target WWTP (Agtrup, Denmark) and formulated it as a time-series prediction problem. Then, we trained six models to use them as simulation environments to implement DRL algorithms. These models had different architectures for time series prediction, such as linear (*LTSF Linear* (Zeng et al., 2022)), recurrent neural networks (*LSTM* (Hochreiter and Schmidhuber, 1997)), attention-based (*Transformer* (Vaswani et al., 2017), *Informer* (Zhou et al., 2021)), and auto-correlation (*Autoformer* (Wu et al., 2021)). We intended to use the various models to explore different approaches to identify the most effective time series forecasting model for implementing deep reinforcement learning algorithms. Through this exploration, we aimed to understand the challenges inherent in modeling such biological processes and identify the best path forward for designing an accurate simulator. The differences in the models' structure influenced the simulation environment results, where we found the strengths and weaknesses of each prediction mechanism. The key contributions of this study are outlined as follows:

- The creation of data-driven simulators for wastewater treatment processes designed to operate effectively without requiring in-depth system knowledge.
- A thorough evaluation of the simulation accuracy across various time series forecasting models, ranging from linear models to advanced, recurrent neural networks and attention-based structures.
- An in-depth sensitivity analysis of these models, focusing on how various wastewater treatment process variables impact the accuracy of phosphorus concentration predictions.
- Addressing the challenges in developing accurate simulators for wastewater treatment processes, particularly the issue of compounding errors.
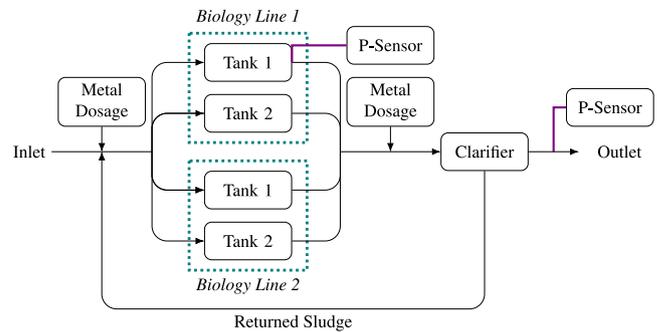
Fig. 1. Schematic of the phosphorus removal process in the plant with the flow lines: The iron salt is added to the inflow to the biological tanks, where P is removed, and a sensor in Tank 1 measures phosphate. There is a dosage of polyaluminium chloride before the secondary settler to remove the remaining P, and the final phosphate concentration is measured at the outlet.

The paper is structured as follows: Section 2 delves into the methodology, detailing the plant, the data, the models, and their implementation as simulators for training DRL algorithms on wastewater treatment processes. Section 3 presents the results of our experiments, showcasing the accuracy of the various models in one-step ahead prediction and multi-step simulations. Section 4 discusses the implications of our findings, drawing comparisons with existing methods and exploring the challenges of developing accurate simulators. Finally, Section 5 concludes the paper with a summary of our contributions and suggestions for future research.

2. Material and methods

2.1. Case study

This study focuses on Kolding central WWTP in Agtrup Denmark, which has a 125,000 population equivalents (PE) capacity and a current load of approximately 65.5% (Anon, 2021). The plant removes phosphorus with a combination of chemical and biological removal methods. In the chemical phosphorus removal, metal salts such as aluminum sulfate (alum), ferrous sulfate, or ferric chloride are added to the wastewater in a rapid mix tank, followed by flocculation to form a precipitate with soluble phosphorus (Burton et al., 2014). On the other hand, biological phosphorus removal relies on naturally occurring microorganisms called Phosphorus Accumulating Organisms (PAO). PAOs release stored phosphorus under anaerobic conditions and remove soluble phosphorus under aerobic conditions (Zhang et al., 2022). Both processes can be challenging to control, as inlet phosphate concentrations can vary unpredictably due to industrial contributions. Conversion of polyphosphate to orthophosphate before coagulant addition will affect coagulation efficiency in chemical removal. Additionally, cultivating a population of organisms that can survive alternating anaerobic and aerobic cycles requires long acclimation periods from a few days to several weeks or even months (Tuszynska et al., 2019). Moreover, high recycle nitrate concentrations can inhibit anaerobic zone processes for biological removal (Chong et al., 2013; Seviour et al., 2003).

2.1.1. Operation

The current operation of the plant for phosphorus removal is shown in Fig. 1. The system consists of a wastewater treatment plant where metal salts are added at two locations: Iron salts after the primary settler and Polyaluminium Chloride before the secondary settler. The biological phosphorus removal part is placed between the two settlers and consists of 2 parallel lines, each including two reaction tanks. Only tank 1 in biology line 1 has the phosphate sensor. The plant is currently being monitored and controlled by the Hubgrade™ Performance Plant system, designed by Krüger/Veolia. The sampling frequency is specified
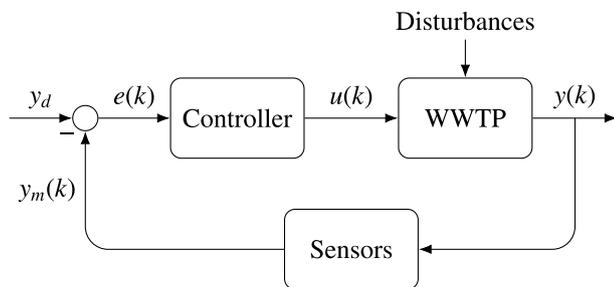
Fig. 2. The control block diagram of the system.

to measure the plant's outputs and the controller's updates, which is currently every two minutes.

The control strategy shown in Fig. 2 involves a feedback control algorithm that calculates the manipulated variable (metal dosage) based on the measured outputs and the setpoint. The specific algorithm used and any limitations or assumptions of the strategy are described in detail in Hansen et al. (2022). In general, the control strategy can be described by the following equation:

$$u(k) = K(y_d - y_m(k)) \tag{1}$$

Where $u(k)$ is the manipulated variable at time k, K is the controller gain, y_d is the setpoint or reference value for the phosphate concentration in the outlet, and $y_m(k)$ is the measured concentration at time k. The system responds to changes in the setpoint or disturbances, with limitations on its performance. The iron salts are dosed at specified locations and timing using a dosing mechanism based on the commands from the explained controller.

2.1.2. Dataset preprocessing

Nineteen months of data (D) were collected from the SCADA system at the Agtrup plant between June 2021 and January 2023. D consisted of 23 two-dimensional vectors, each representing a pair of variables at the treatment plant. Specifically, D can be represented as a matrix of size $n \times m$, where n is the number of measurements taken over the 19 months, and m is the number of variables measured. Each row of the matrix corresponds to a single measurement, and each column corresponds to a particular variable. The value of each variable is represented by the first component of the corresponding vector, while the second component represents the quality of the measurement. The quality variable takes binary values, with 0 and 1 indicating good and bad quality, respectively. More details about the dataset can be found in Hansen et al. (2022).

After preprocessing the data to detect issues such as insufficient quality data, negative values, and missing values, the dataset with 23 variables has been processed with feature selection or engineering methods. Insufficient quality, negative, and missing data were replaced by the previously available values, which did not have any of those issues. The Pearson correlation method of the *pandas* library (The pandas development team, 2020) was used to investigate the most critical variables of the plant affecting the phosphate concentration. This correlation method is a statistical measure that quantifies the linear relationship or degree of association between two continuous variables (Faizi and Alvi, 2023). The selected variables were Nitrate concentration [mg/L], Ammonia concentration [mg/L], and Ammonia plus Nitrate concentration [mg/L], which are added to the metal dosage flow [m^3/hr] and Phosphate concentration [mg/L] to form the final dataset. Additionally, Gradient Boosting Regression with decision trees was used to determine the most optimum way of using time features (hour, day of the week, and month) in the dataset to improve the prediction accuracy.

2.2. Deep learning models

The following will explain the types of different deep learning models used in the simulation environment for the phosphorus removal process in wastewater treatment plants. All models were built with the *PyTorch* library (Paszke et al., 2019).

2.2.1. Long Short-Term Memory (LSTM)

The *LSTM* was introduced by Hochreiter and Schmidhuber (1997) as a successful technique for addressing the vanishing gradient problem in recurrent neural networks. Unlike conventional RNNs, which use a recurrent node, the LSTM structure replaces the recurrent node with a memory cell. The memory cell contains a self-connected recurrent edge of fixed weight 1, which allows the gradient to pass through many time steps without disappearing or exploding. The memory cell is also equipped with multiple multiplicative gates, such as the input gate, forget gate and output gate. These gates determine whether to impact the internal state of the neuron, flush the internal state to zero, or allow the internal state to influence the cell's output, respectively.

LSTM models have become popular for time series forecasting and system identification due to their ability to capture long-term dependencies in the data and handle variable-length input sequences (Hansen et al., 2022; Wang, 2017). The procedure of forecasting time series with *LSTM* model is shown in Fig. 3. The power of *LSTM* to fit nonlinear long periodic data patterns like wastewater treatment data has resulted in its increased application in this field (Yunpeng et al., 2017). Despite the power of *LSTM* in time series forecasting, it sometimes has trouble capturing long-term dependencies, for example, in language modeling (Khandelwal et al., 2018), and the model cannot be parallelized.

2.2.2. Transformer

An important part of the *Transformer* model is the attention mechanism, originally intended to be a sequence-to-sequence RNN improvement; it was used to enhance encoder–decoder RNNs for machine translation (Bahdanau et al., 2014). Instead of compressing the input, attention suggests that the decoder revisit the input sequence at every step rather than compressing it. The decoder might be able to focus on particular parts of the input sequence at particular decoding steps rather than always seeing the exact representation of the input. As each step of the decoding process was performed, the attention mechanism enabled the decoder to dynamically attend to a different part of the input (Bahdanau et al., 2014). The *Transformer* architecture for machine translation was proposed by Vaswani et al. (2017), which dispenses with recurrent connections and incorporates cleverly arranged attention mechanisms instead of recurrent connections. As a result of its outstanding performance, the *Transformer* began to appear in most of the advanced natural language processing systems by 2018. The attention mechanism of the *Transformer* can be described by the following equation (Vaswani et al., 2017):

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

Here, Q, K, and V represent the queries, keys, and values matrices. Furthermore, d_k represents the dimension of queries and keys. The softmax function converts the vector of numbers to the vector of probabilities.

As shown in Fig. 4, the most critical layer of the *Transformer* is self-attention, which means compression of attention toward itself. Compared to previous recurrent and convolutional architectures, self-attention has the following advantages: parallel operations (compared to RNN) and no need for deep networks to find long sentences (Shaw et al., 2018). As self-attention can lead to loss of ordering information and the *Transformer*, the encoder has no recurrence like RNNs; the model takes advantage of positional encoding to preserve information about the order of tokens (Shaw et al., 2018; Vaswani et al., 2017).
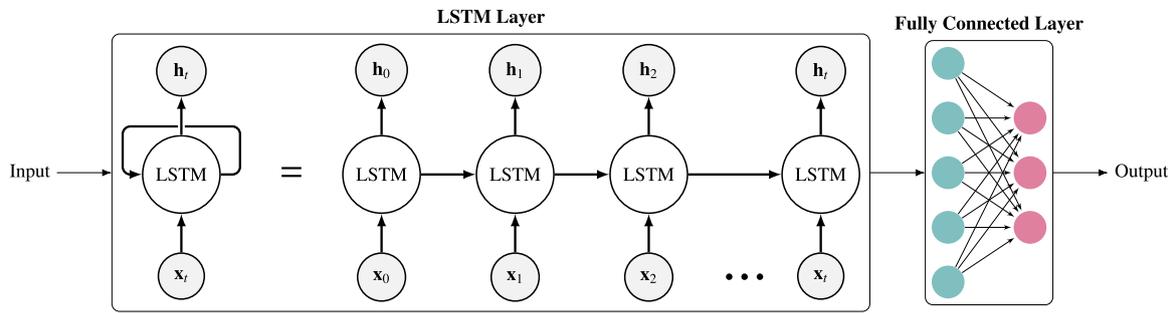
Fig. 3. The structure of the **LSTM** (Hochreiter and Schmidhuber, 1997) model for time series forecasting tasks, where $(\mathbf{x}_0, \ldots, \mathbf{x}_t)$, and $(\mathbf{h}_0, \ldots, \mathbf{h}_t)$ represent the input and the hidden state (output) of each LSTM cell.
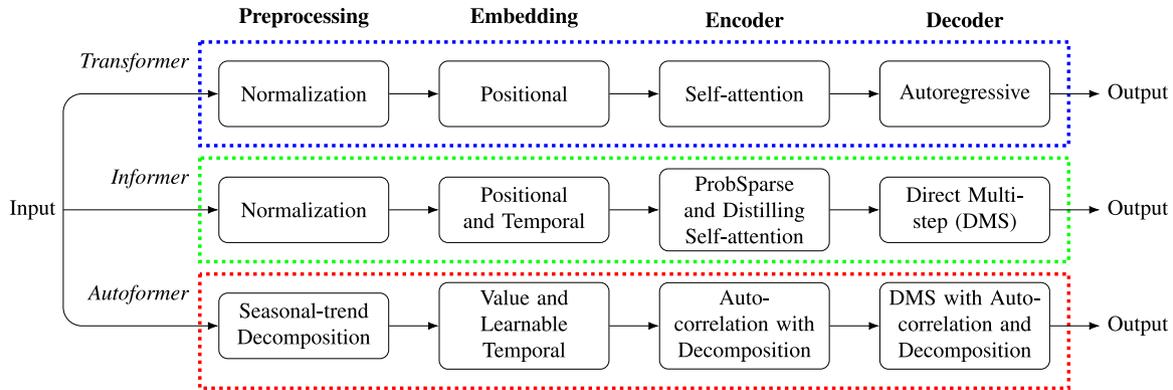
Fig. 4. The structures of encoder–decoder attention-based models (**Transformer** (Vaswani et al., 2017), **Informer** (Zhou et al., 2021)), and auto-correlation (**Autoformer** (Wu et al., 2021)) for time series forecasting tasks.

Compared to the *LSTM*, the *Transformer* is potentially more successful in capturing recurrent patterns with long-term dependencies. The model can access any part of history, no matter the distance (Lai et al., 2018). However, using the *Transformer* for extremely long sequences requires an immense computation power because the growth of space complexity in self-attention is quadratic (Huang et al., 2018), which can lead to problems in forecasting data with substantial long-term dependencies (Lai et al., 2018). Also, the *Transformer* uses autoregression to decode the output, resulting in error accumulation in long-term predictions (Zeng et al., 2022).

2.2.3. Informer

Applying the *Transformer* model to long-term time series forecasting (LTSF) has critical issues such as quadratic time complexity, high memory usage, and the inherent limitation of the encoder–decoder architecture (Zhou et al., 2021). The *Informer* model was introduced in Zhou et al. (2021) to overcome the limitations of the *Transformer*-based model for LTSF applications. To do so, *Informer* introduced some new features:

- Implementing the ProbSparse self-attention mechanism is proposed to reduce canonical self-attention. It achieves the complexity and memory usage of $O(L \log L)$, where L is the number of layers in the network.
- Attention scores are dominated by self-attention distilling operations.
- Improvements in prediction accuracy in LTSF problems, which contain *Transformer*-like models for capturing individual time-series dependency.
- Reduced space complexity to $O((2 - \epsilon)L \log L)$, where the parameter ϵ is a small positive constant that controls the method's accuracy.
- A generative style decoder was introduced to obtain long sequences with only one forward step.

The pipeline of the time series forecasting task with the *Informer* model can be seen in Fig. 4. The *Informer* employs a generative decoder with direct multi-step (DMS) forecasting, which can produce more accurate predictions as it does not use the last prediction results as input for the next step (Zhou et al., 2021; Zeng et al., 2022).

2.2.4. Autoformer

Using residuals and encoder decoders, *Autoformer* reconstructs the *Transformer* into a decomposition forecasting architecture using residuals and encoder decoders (Wu et al., 2021). The pipeline of the *Autoformer* model in Fig. 4 illustrates the differences between its architecture and attention-based models. The *Autoformer* extracts the trend-cyclical component of the time series by decomposing it into seasonal-trend blocks behind the neural blocks. Additionally, the *Autoformer* replaces attention with the Auto-Correlation mechanism, which calculates the relationship between the current value of the variable and its past values. Auto-correlation explores period-based dependencies by counting the series autocorrelations and aggregating similar subseries by time delay.

2.2.5. LTSF linear

Autoregressive or iterated multi-step (IMS) prediction of the long-term time series data has been proven to have some problems, like the significant accumulation of errors (Zeng et al., 2022). In Zeng et al. (2022), authors challenge the effectiveness of *Transformer*-based models for long-term time series prediction by direct multi-step (DMS) forecasting methods, which have better results when the prediction horizon is large. They argue that applying self-attention to time series data can lead to loss of ordering information, and positional encoding of the data cannot help preserve the temporal information. They claim that the time series data must have significant trends and seasonality to make a long-term prediction. Calling them embarrassingly simple models, they introduce a set of LTSF Linear baselines
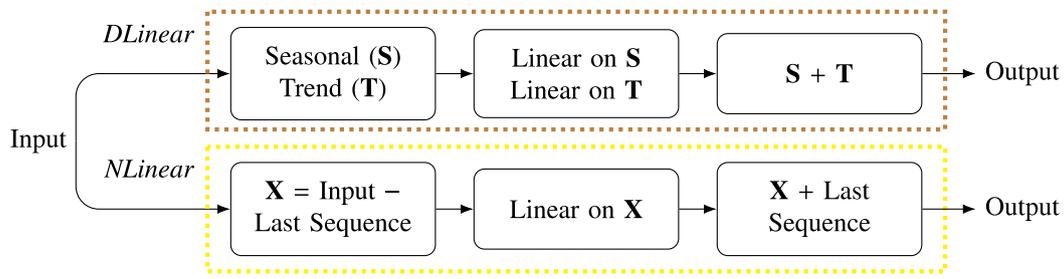
Fig. 5. The pipeline of **LTSF Linear** (Zeng et al., 2022) models (*DLinear* and *NLinear*) to forecast time series.

to compare with *Transformer*-based methods. LTSF Linear outperforms *Transformer*-based models up to 20%–50% in long-term time series forecasting, and they prove that in contrast to the past claims, the performance of *Transformer*s will not be improved by increasing the look-back window size (Zeng et al., 2022).

LTSF Linear is the simplest direct multi-step model with a linear temporal layer. The basic linear model uses a weighted sum operation of historical time series to predict the future. Considering the weighted sum operation, we will have the following:

$$\hat{\mathbf{X}}_i = \mathbf{W} \times \mathbf{X_i} \tag{3}$$

In the above formula, $\hat{\mathbf{X}}_i$ is the prediction, and $\mathbf{X_i}$ is the input for each variable (i). The linear layer is shown by $\mathbf{W}$ where $\mathbf{W} \in \mathbb{R}^{T \times L}$.

DLinear is one type of LTSF Linear model with a specific data preprocessing method. They used the decomposition layer introduced in previous *Autoformer* (Wu et al., 2021) and FEDformer (Zhou et al., 2022), then combined it with linear layers. It extracts two different components from the raw data, one is a trend component, and the other one is a reminder or a seasonal component. It gets the final prediction by summing up the results from two linear layers applied to each component (Zeng et al., 2022). *NLinear* is another linear model specifically used when a distribution shift is found in the dataset. Firstly, *NLinear* subtracts the input by the last value of the sequence as a simple normalization and then sends the input through a linear layer. At the end of the procedure, the subtracted part is added to the output when making the final prediction (Zeng et al., 2022). Fig. 5 shows the pipelines and differences of these two linear models while performing forecasting tasks.

2.2.6. Training of the models

Each one of the models was trained individually with the *PyTorch* library (Paszke et al., 2019) on the dataset from the actual plant as described in Section 2.1.2 as input. The dataset was divided between training and testing parts with a ratio of 85%/15%, respectively, and 15% of the training dataset was used for validation. The input to the model at each training step was a sequence of time steps with length *l*, including all the features of the pre-processed WWTP dataset. The output for each step was the values of all the features for one time step after the last step in the input.

At each time step, $\mathbf{x} = (x_1, \dots, x_n)$ denotes the values of the system's features. The sequence length (*l*) was set at 240, representing 240 min of the historical data as input. To ensure all models are trained in the best way possible, *Optuna* (Akiba et al., 2019) library is used for hyper-parameter optimization. *Optuna*'s Tree-structured Parzen Estimator (TPE) uses a sampler derived from Bayesian (Mockus, 2012) optimization. Using TPE, *Optuna* finds points closer to previous good results rather than at random (Akiba et al., 2019). We first defined a range for each hyper-parameter utilized in the training of models and then used *Optuna* to find the best combination of them, which had the lowest mean squared error for the prediction of the test dataset. After this step, the models were trained with the optimized hyper-parameters by using *Adam* as the optimization method and mean squared errors (*mse*) as the loss function. The script from Zeng et al. (2022), which

is available on GitHub was used with some changes to train *DLinear* and *NLinear* models. The best checkpoint for each model, which had the lowest *mse* for the validation dataset, was used to perform a one-step prediction on the test dataset and saved for further use in the simulation environment. Fig. 6 shows a schematic representation of the steps needed to train, test, and save each one of the models described in Section 2.2.

2.2.7. Sensitivity analysis

Time series forecasting often involves multivariate input data where the relationships between features and predictions can be very important. Sensitivity analysis seeks to unravel these relationships by quantifying the impact of variations in input features on the model's output (Saltelli et al., 2010; Al et al., 2019). It is an incredible tool for model interpretability, feature selection, and understanding the robustness of predictions in real-world scenarios. The Sobol method is a variance-based sensitivity analysis technique that decomposes the total variance in model output into contributions from individual input features and their interactions (Saltelli et al., 2010). The Sobol indices provide clear insights into the relative importance of features. The Sobol method employs a series of equations to calculate sensitivity indices, specifically the first-order (main effects) and total indices. Let us define our model output as Y and a set of input features as $[X_1, X_2, \dots, X_n]$. The first-order Sobol indices (S_i) measure the contribution of each individual input feature to the output variance (Saltelli et al., 2010):

$$S_i = \frac{\text{Var}(E(Y|X_i))}{\text{Var}(Y)} \tag{4}$$

Where: S_i, $\text{Var}(E(Y|X_i))$, $\text{Var}(Y)$ represent the first-order Sobol index for feature X_i, the conditional variance of Y given X_i and the total variance of Y, respectively. The total Sobol indices (S_{T_i}) account for both individual and interaction effects (Saltelli et al., 2010):

$$S_{T_i} = 1 - \frac{\text{Var}(E(Y|X_{\sim i}))}{\text{Var}(Y)} \tag{5}$$

In the above equation, S_{T_i} and $\text{Var}(E(Y|X_{\sim i}))$ are the total Sobol index for feature X_i and the conditional variance of Y due to all variables except X_i, respectively. We used the *SALib* library (Iwanaga et al., 2022) in *Python* to perform Sobol sensitivity analysis for all models, where the number of model evaluations (N) was set to be 1024, and the sensitivity of the P amount in models' output to each feature in the preprocessed dataset was calculated.

2.3. Formulation of the simulator

Phosphorus removal in WWTPs can be formulated as a nonlinear dynamical system, as P concentration in wastewater is a dynamic variable that changes over time due to inflow variations, biochemical reactions, and chemical additions (Kazadi Mbamba et al., 2019). Let us consider a nonlinear dynamical system with a vector of manipulated variables (**u**), a vector of measured outputs (**y**), a vector of unmeasured disturbances (**w**), and a vector of state variables (**x**). The following set
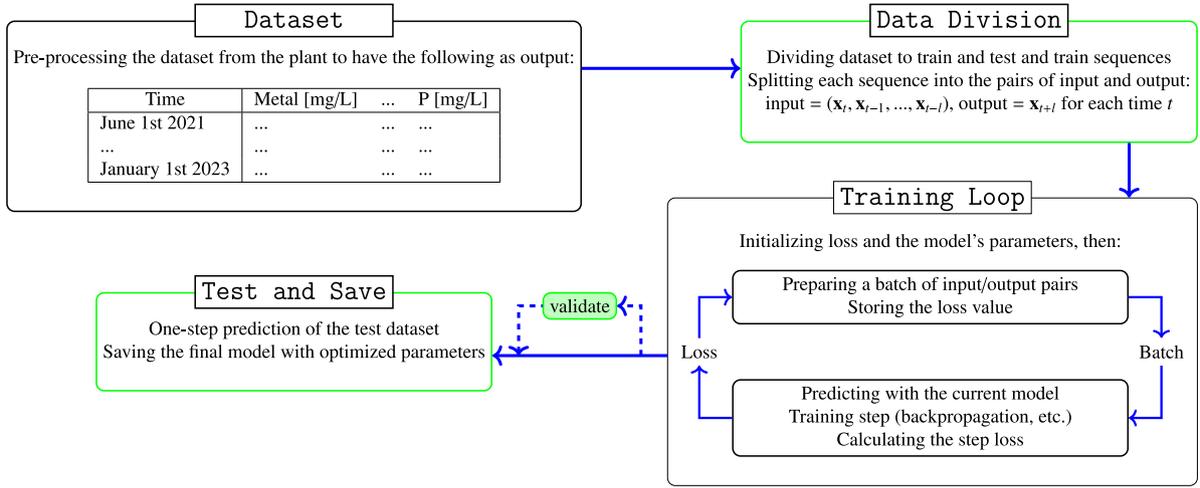
Fig. 6. The procedure of training, testing, and saving a model for the prediction of future time steps of the plant's data.

of equations can represent this system (Brunton and Kutz, 2019; Åström and Wittenmark, 2013; Wang, 2017):

State equation:

$$x(k + 1) = \mathbf{f}(k, x(k), u(k), w(k)) \tag{6}$$

Where $\mathbf{x}(k)$, $\mathbf{u}(k)$, and $\mathbf{w}(k)$ represent the state variables, the manipulated variables, and the unmeasured input disturbances at time k respectively.

Output equation:

$$y(k) = \mathbf{g}(k, x(k), u(k), v(k)) \tag{7}$$

Where $\mathbf{y}(k)$, and $\mathbf{v}(k)$ represent the measured output and the measured disturbances at time k. In the above equations, $\mathbf{f}$ represents the system's dynamics, while $\mathbf{g}$ represents the output equation. The function f describes how the state variable changes over time, given the values of the manipulated variable, measured output, and unmeasured disturbance. The function $\mathbf{g}$, on the other hand, describes how the state variable influences the measured output. The representation of the unmeasured disturbances in the system dynamics model needs disturbance models for both input and output. It requires accurate disturbance data, which is not possible for all systems. Here, we assume that the unmeasured disturbances ($w(k)$) are included in the state variables data ($\mathbf{x}(k)$) and result in an estimation of the true states for every step.

Now, let us consider a dynamical system with input history length l and manipulated variables $\mathbf{u}(k)$ and output at time $k + 1$ given by Cocianu et al. (2022), Zarzycki and Ławryńczuk (2021):

$$x(k + 1) = \mathbf{f}(x(k), x(k - 1), \dots, x(k - l + 1),$$
$$u(k), u(k - 1), \dots, u(k - l + 1)) \tag{8}$$

In the above equation, f is a nonlinear function that describes how the system evolves over time based on the input history $\mathbf{x}(k), \mathbf{x}(k - 1), \dots, \mathbf{x}(k - l + 1)$, and the manipulated variables $\mathbf{u}(k), \mathbf{u}(k - 1), \dots, \mathbf{u}(k - l + 1)$. In the current study, we use six deep learning models described in Section 2.2 as the f function stated in Eq. (8).

Here, we will explain the *LSTM* as an example of how a deep learning model can describe a specific discrete-time dynamical system. The basic *LSTM* model consists of three layers: the input layer, the *LSTM* layer, and the fully connected layer (Hochreiter and Schmidhuber, 1997). In order to achieve the value of $\mathbf{x}(k + 1)$ in Eq. (8), the state variables ($\mathbf{x}(k - l), \dots, \mathbf{x}(k)$), and the manipulated variables ($\mathbf{u}(k - l), \dots, \mathbf{u}(k)$) first will be passed through the *LSTM* layer. In this layer, the cell input activation vector ($\tilde{\mathbf{c}}$), the cell state vector ($\mathbf{c}$), and the hidden state vector ($\mathbf{h}$) will be computed at each time point $t \in [k - l, k]$ by the following equations (Hochreiter and Schmidhuber, 1997):

$$\tilde{\mathbf{c}}(t) = \sigma_c(\mathbf{W}_c(\mathbf{x}(t), \mathbf{u}(t)) + \mathbf{U}_c \mathbf{h}(t - 1) + \mathbf{b}_c) \tag{9}$$

$$\mathbf{c}(t) = \mathbf{f}(t) \cdot \mathbf{c}(t - 1) + \mathbf{i}(t) \cdot \tilde{\mathbf{c}}(t) \tag{10}$$

$$\mathbf{h}(t) = \mathbf{o}(t) \cdot \sigma_h(\mathbf{c}(t)) \tag{11}$$

Where $\mathbf{W}_c$, $\mathbf{U}_c$, and $\mathbf{b}_c$ are the weight matrix, recursive weights matrix, and the bias components of the *LSTM* cell, respectively, additionally, $\mathbf{f}(t)$, $\mathbf{i}(t)$, and $\mathbf{o}(t)$ represent the forget, input, and output gates activation vectors. Also, σ_c and σ_h are the sigmoid and the hyperbolic tangent functions. After this step, the output of the *LSTM* layer, which is its hidden state ($\mathbf{h}$), will be passed through a fully connected layer. According to Hochreiter and Schmidhuber (1997), Zarzycki and Ławryńczuk (2021), Cocianu et al. (2022), the output from the *LSTM* model for a dynamic system can be computed as follows:

$$x(k + 1) = \mathbf{W}_x \mathbf{h}(k) + \mathbf{b}_x \tag{12}$$

$\mathbf{W}_x$ and $\mathbf{b}_x$ are the weight vector and the bias components of the fully connected layer at the output of the *LSTM* model. $\mathbf{h}(k)$ is the hidden state or the output vector of the *LSTM* layer at time k. The final output of the *LSTM* network is the system's state variables $\mathbf{x}$ at time $k + 1$.

The explained method can be used to identify and simulate dynamic systems. This identification approach can also be utilized as a train and test environment for process control with deep reinforcement learning research in different industries. To do so, the first step will be formulating the industrial process as a discrete-time dynamical system. Then, the system function or the model should be specified and fit the historical data of the process. Once the trained model is ready, we can create the specific environment with the saved model and used dataset.

Generally, a reinforcement learning environment returns the agent four variables at each step. These four variables are the current or predicted state of the system, reward, a Boolean value specifying whether the episode has been finished or not called done, and info that consists of different information such as the current step of the simulation, total reward, and history of observations. Total reward, or cumulative reward, refers to the sum of rewards obtained by an agent over a sequence of interactions with an environment. It quantifies the performance of the agent in achieving its goals. In continuous tasks, the cumulative reward is often discounted by a gamma (γ) factor at each time step, which is calculated over T steps as (Lillicrap et al., 2015):

$$R_t = \sum_{i=t}^{T} \gamma^{(i-t)} \mathbf{r}(\mathbf{s}_i, a_i) \tag{13}$$

In the above equation, R_t is the cumulative reward, $\mathbf{r}$ is the reward function where $\mathbf{s}_i$ and a_i are the system's state and the action taken in

Table 1

Parameters used in the explanation of simulation environment and testing algorithm.

Parameter	Definition
episode length	the number of episodes (time steps) for testing the simulation
p	the date-time index of the initial input to the simulation environment
l	sequence length of the historical input data used in the model training
n	number of the state variables (features) in the dataset
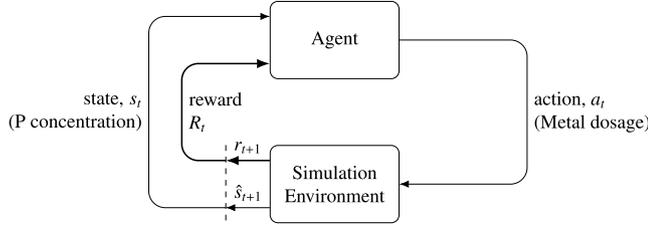
Fig. 7. The process of a testing simulation environment for one model in a specific time point.

Algorithm 1 DRL simulation environment testing

Inputs:
　dataset, model, p, M (episode length), and l
Outputs:
　The Simulated States over M episodes
Initialize:
　$t \leftarrow 0$
　s_0 from the point *p* of the dataset with size $l \times n$
　a_0 is the control variable from the last time step in s_0
for episode = 1, M **do**
　$\hat{s}_{t+1} \leftarrow model(s_t, a_t)$
　append $\hat{s}_{t+1}$ to s_t and update
　append $\hat{s}_{t+1}$ to *Simulated States*
　update a_{t+1} from the dataset
　$t \leftarrow t + 1$
end for

---

the step *i*. The agent will work based on the returned variables from the environment and generate the action (*a*), which, in our case, is the flow of metal dosage in the process. The action from the agent will replace the current metal addition in the state vector. The predicted state will be added to the state's history, and the process will be repeated. The whole procedure is shown in Fig. 7, which will be further explained in the following.

We created a test script to study the accuracy of the simulation environment with the six different models. The script takes one model as the predictor and starts prediction with a series of data from the actual dataset. We give each test step the episode length for which the environment must run. The algorithm for testing the developed environment is shown in algorithm 1. First, the trained model and dataset from the real plant are loaded, as explained in Section 2.1.2. Then, some parameters are defined as explained in Table 1 to test the environment.

In deep reinforcement learning, the state of the system at each time step *t* is defined as $\mathbf{s_t}$, and when using a model as $\hat{s}_t$ because it is a prediction. We can rewrite Eq. (6) as below for a DRL simulation environment:

$$\hat{\mathbf{s}}_{t+1} = f(\mathbf{s_t}, a_t) \tag{14}$$

Where $\hat{s}_{t+1}$ is the predicted state of the system at time $t+1$, and a_t is the action taken from the agent at time *t*. The input to the environment ($\mathbf{s_0}$) is initialized from the time point **p** with the size of $l \times n$. This input is sent to the environment as the system's current state, where the next state will be calculated according to Eq. (14). Then, the predicted state of the system ($\hat{s}_{t+1}$) is delivered to the test agent to produce an action. The agent produces an action (a_{t+1}), the real amount of the action variable collected from the plant. Finally, the action (a_{t+1}) is added to the input data ($\mathbf{s_{t+1}}$) for the next state and replaces the predicted amount of action variable. This process is repeated for the number of **episodes** specified, and the values of action, predicted objective variable and real objective variable from the dataset are saved for each time step.

By doing so, we are able to compare the real and simulated phosphorus amount at each step and observe the accuracy of the simulation environment in case of having a real-time controller. The real phosphorus amount comes from the original dataset, and the simulated one is extracted from the state of the environment.

2.4. Software and hardware

All of the tests for the simulation environment are implemented in programming language *Python* by using the *Gym* (Brockman et al., 2016) and *PyTorch* (Paszke et al., 2019) libraries. The AI Cloud service

from Aalborg University is used for GPU-based computations. The used compute nodes are each equipped with 2×24-core *Intel Xeon* CPUs, 1.5 TB of system RAM, and 16 *NVIDIA Tesla V100* GPUs with 32 GB of RAM each, all connected via *NVIDIA NVLink*.

3. Results

This section will present the results from the models' training and testing of the simulation environment for all of them. The results are discussed in Section 4.

3.1. Models

After training the models in Section 2.2, the one-step prediction results for the test dataset were compared. Each algorithm was tested several times by combining its main parameters, and the best model was determined by obtaining the lowest errors according to the calculated metrics. The considered metrics at each time step *t*, where $\mathbf{x}_{t,n}$ and $\hat{\mathbf{x}}_{t,n}$ are the actual and predicted values of the variable *n* in the dataset with the total number of variables *N*, and total number of samples *T*, were as following:

- **Mean Squared Error (MSE)**: Represents the average squared difference between actual and predicted values. A lower MSE suggests closer predictions to the actual values.

$$\text{MSE} = \frac{1}{T \times N} \sum_{t=1}^{T} \sum_{n=1}^{N} (\hat{\mathbf{x}}_{t,n} - \mathbf{x}_{t,n})^2 \tag{15}$$

- **Root Mean Squared Error (RMSE)**: The square root of MSE, providing error in the original data units. A lower RMSE indicates that the model's average predictions are closer to the actual values.

$$\text{RMSE} = \sqrt{\frac{1}{T \times N} \sum_{t=1}^{T} \sum_{n=1}^{N} (\hat{\mathbf{x}}_{t,n} - \mathbf{x}_{t,n})^2} \tag{16}$$

- **Mean Absolute Error (MAE)**: Measures the average absolute difference between actual and predicted values. Unlike MSE, it

Table 2

The metrics of the trained models for one-step prediction of the test dataset in the current work and previous studies, where the most optimum results are highlighted via the bold font.

Models	Variables	Metrics				Ref.
		mae	mse	rmse	r^2	
LSTM	5 2.1.2	0.0181	0.0012	0.0346	0.9471	Current Study
Transformer	5 2.1.2	0.0178	0.0013	0.0364	0.9644	
Informer	5 2.1.2	0.0184	0.0012	0.0346	0.9598	
Autoformer	5 2.1.2	0.0241	0.0015	0.0389	0.9096	
DLinear	5 2.1.2	**0.0081**	**0.0007**	**0.0255**	**0.9835**	
NLinear	5 2.1.2	0.0125	0.0009	0.0308	0.9800	
LSTM	1 (NH_4)	0.2730	0.3790	0.6160	–	Salles et al. (2022)
Transformer	1 (NH_4)	0.0300	0.0030	0.0510	–	Salles et al. (2022)
LSTM	1 (P)	–	0.0770	–	0.4960	Xu et al. (2023)
LSTM	1 (TP)	–	0.0180	–	–	Yaqub et al. (2020)
LSTM	1 (TN)	–	0.0150	–	–	Yaqub et al. (2020)
LSTM	1 (TP)	0.01176	0.00039	–	0.87456	Ly et al. (2022)

Table 3

Optimized hyper-parameters and the computation time of one iteration over the training data for all models.

Models	Hyper-parameters					
	Learning rate	Dropout	Batch-size	Number of layers	Dimension of layers	Cost time (s)
LSTM	1e−6	0.1	32	2	249	1053
Transformer	1e−7	0.1	64	2 (Enc.) - 1 (Dec.)	512	1123
Informer	1e−7	0.1	64	2 (Enc.) - 2 (Dec.)	512	2225
Autoformer	1e−7	0.1	64	2 (Enc.) - 1 (Dec.)	512	2171
DLinear	1e−6	0.1	16	–	–	335
NLinear	1e−6	0.1	16	–	–	242

does not penalize large errors as harshly. A lower MAE indicates better model accuracy.

$$\text{MAE} = \frac{1}{T \times N} \sum_{t=1}^{T} \sum_{n=1}^{N} |\hat{\mathbf{x}}_{t,n} - \mathbf{x}_{t,n}| \qquad (17)$$

- **Coefficient of determination** (r^2 **score**): Indicates the proportion of variance in the dependent variable that is predictable from the independent variable(s). A score closer to 1 means better prediction accuracy, while a score closer to 0 suggests poor model performance. Considering $\bar{\mathbf{x}}_n$ the mean value of the actual sequence over all samples, the r^2 score for the variable n can be computed as follows:

$$r^2 = 1 - \frac{\sum_{t=1}^{T} (\hat{\mathbf{x}}_{t,n} - \mathbf{x}_{t,n})^2}{\sum_{t=1}^{T} (\bar{\mathbf{x}}_n - \mathbf{x}_{t,n})^2} \qquad (18)$$

These metrics offer insights into the model's prediction accuracy in multivariate time series forecasting, with lower error values (MSE, RMSE, MAE) and higher r^2 scores, indicating better model performance. Table 2 shows the metrics for one-step prediction of the test dataset with the best-trained models, which had the lowest errors, alongside the results from similar previous studies. The majority of the studies used *LSTM* model to predict key variables in WWTPs like NH_4 (Salles et al., 2022), total nitrogen (TN) and total phosphorus (Yaqub et al., 2020), and effluent total phosphorus (TP) (Xu et al., 2023; Ly et al., 2022). One of the studies used *Transformer* model to predict the critical variables in WWTPs (Salles et al., 2022), while no similar applications of the other models were found in the literature.

Table 3 shows optimized hyper-parameters for all models using Bayesian optimization, which were chosen according to the most optimum metrics reported in Table 2. Additionally, Fig. 8 shows the test dataset's one-step prediction results for all models. The MSE reported in Fig. 8 is the mean squared error of the model's prediction for the test dataset. Moreover, the results of the Sobol sensitivity analysis for all models as described in 2.2.7 can be seen in Fig. 9, which shows the sensitivity of the phosphate amount in the model's prediction to each one of the input features. Additionally, the computation time of training for all models using the same hardware mentioned in Section 2.4 is shown in Table 3.

3.2. Simulation environment

The developed simulation environment was tested for all the models explained in Section 2.2. In order to get a comprehensive overview of the simulator results with the models, four sequences from different seasons were chosen. The four sequences were selected to explore the behavior of the models during different periods of the year when the data varied in terms of trends, disturbances, and sensor failures. The results of testing the simulation environment for different sequences are shown in Fig. 10. It is worth noting that the selection of these sequences was based on the characteristics of the data and did not relate to any seasonal studies. Mean squared error and Dynamic Time Warping (DTW) are used as metrics to analyze the behavior of models as simulators. The mean squared error over all of the state dimensions at each step of the simulation is computed as follows:

$$\text{MSE}_{single\_step} = \frac{1}{d_s} \sum_{d=1}^{d_s} (\hat{\mathbf{s}}_{t,d} - \mathbf{s}_{t,d})^2 \qquad (19)$$

In the above equation, $\hat{\mathbf{s}}_{t,d}$ and $\mathbf{s}_{t,d}$ are the predicted and actual state of the system at time t, while d_s indicates the last index of the state dimensions, which for our problem is the number of features N. According to the Eqs. (15) and (19), the average mean squared error over h consecutive steps for a simulation environment can be calculated by following:

$$\text{MSE}_{multi\_step} = \frac{1}{h \times d_s} \sum_{t=1}^{h} \sum_{d=1}^{d_s} (\hat{\mathbf{s}}_{t,d} - \mathbf{s}_{t,d})^2 \qquad (20)$$

DTW is a robust algorithmic approach mainly used for identifying similarities between two temporal sequences, say $X = [x_1, x_2, \ldots, x_N]$ and $Y = [y_1, y_2, \ldots, y_M]$, which may differ in speed or alignment. Contrary to standard methods, which rely on direct point-to-point comparisons, DTW optimally aligns sequences by iteratively "warping" the time axis (Müller, 2007). To elucidate, given two sequences of lengths N and M respectively, DTW computes a $N \times M$ cost matrix D, where each element $D(i, j)$ represents the distance between x_i and y_j. A common distance measure used is the Euclidean distance:

$$d(x_i, y_j) = (x_i - y_j)^2 \qquad (21)$$
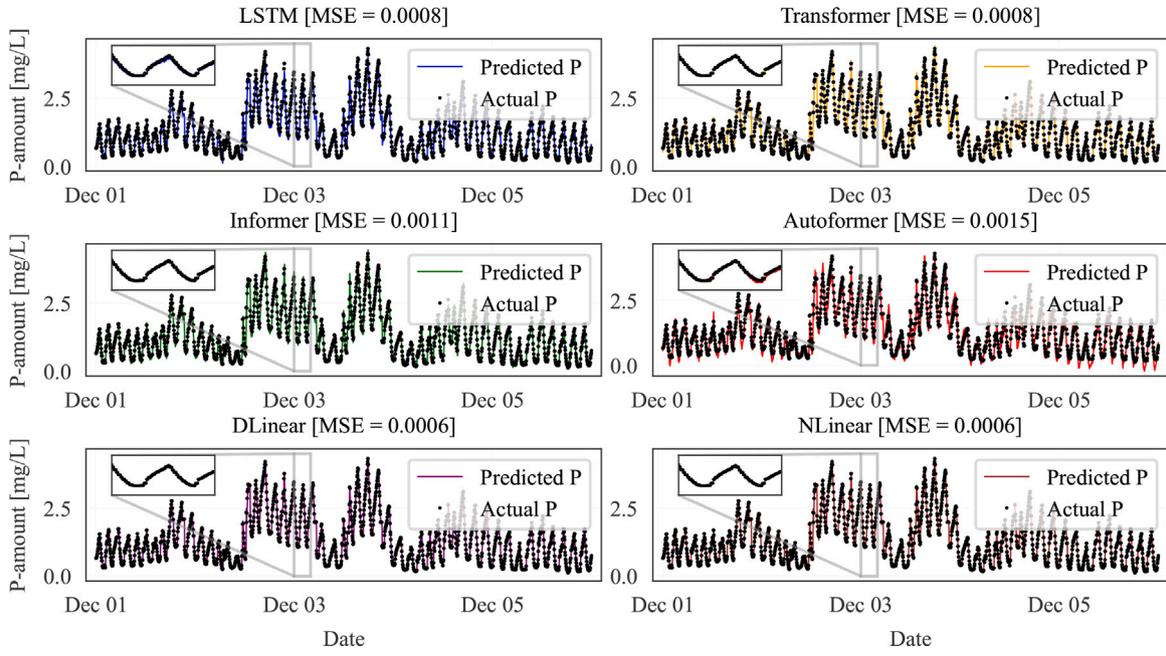
Fig. 8. Prediction results of all models for a part (5 days) of the test dataset.
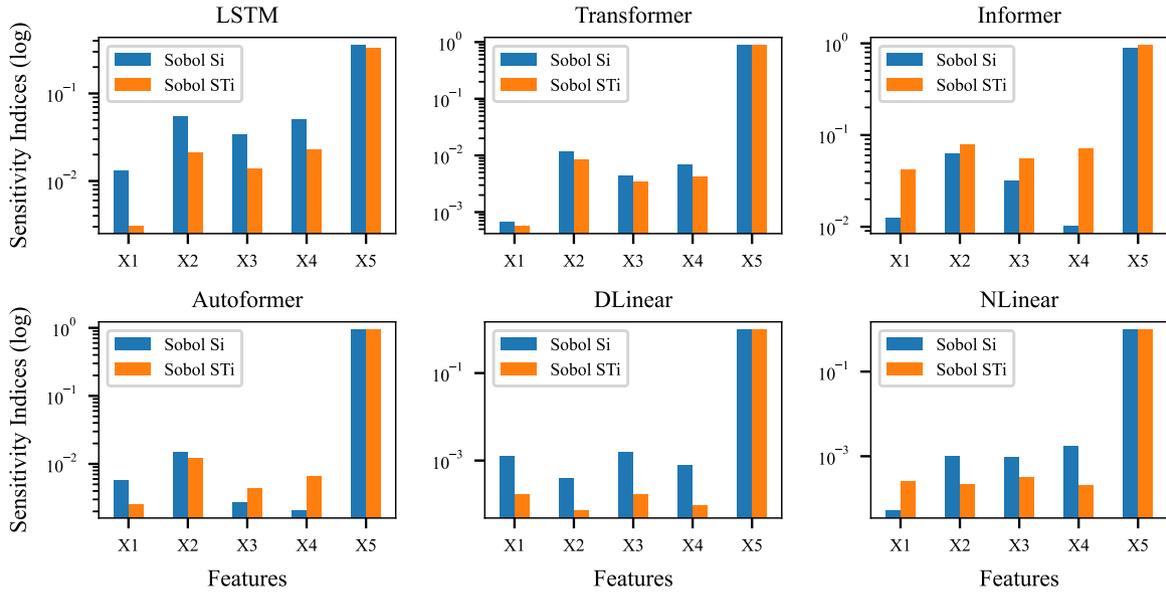
Fig. 9. Sobol sensitivity indices (logarithmic) of the P amount in the prediction to each one of the input features for all models, where $X1$, $X2$, $X3$, $X4$, and $X5$, represent the metal dosage, NH_4, NO_3, $NH_4 + NO_3$, and PO_4, respectively.

The objective is to find a warping path W that minimizes the cumulative distance:

$$D(i,j) = d(x_i, y_j) + \min\{D(i-1,j), D(i-1,j-1), D(i,j-1)\} \quad (22)$$

with boundary conditions $D(1,j) = \sum_{k=1}^{j} d(x_1, y_k)$ and $D(i,1) = \sum_{k=1}^{i} d(x_k, y_1)$.

The optimal path is the sequence of matrix indices that minimizes the total distance from $(1,1)$ to (N,M). This warping path represents how one sequence can be optimally "stretched" or "compressed" to align with the other. DTW's flexibility in matching varying temporal structures has made it indispensable in domains like speech recognition, bioinformatics, and gesture recognition, where the intrinsic dynamics and patterns within sequences are of greater importance than their exact temporal alignment.

The metrics of the testing simulation environment for the four sequences mentioned in Section 3.2 are shown in Table 4, where the number of simulation steps was considered to be 180 min. In Table 4, the lowest values of MSE and DTW for each sequence are highlighted in bold and underlined for clarity.

4. Discussion

Based on the results of the one-step prediction of the test dataset, Table 2 and Fig. 8, the best models are able to predict the next state of the system more than 97% accurately. The metrics shown in Table 2 suggest that the *DLinear* model outperforms the *LSTM* and formers while being close to *NLinear*. The *Autoformer* model exhibits the lowest accuracy among the models investigated in this work. This can relate to the highly stochastic nature of the wastewater treatment processes,

Table 4
The values of mean squared error and dynamic time warping for all models in different sequences.

Models	Autumn		Winter		Spring		Summer		Average	
	MSE	DTW	MSE	DTW	MSE	DTW	MSE	DTW	MSE	DTW
LSTM	63.812	29.219	124.323	11.302	2.866	2.874	23.831	4.207	53.708	11.901
Transformer	7.582	30.161	5.304	7.907	17.72	2.613	13.228	3.901	10.958	11.145
Informer	379.524	33.469	86.131	11.245	20.917	3.559	9.443	4.457	124.004	13.182
Autoformer	283.529	25.964	477.33	3.729	708.111	1.575	739.358	4.267	552.082	8.884
DLinear	11.092	26.899	16.237	4.137	5.658	3.029	11.358	3.724	11.086	9.447
NLinear	16.772	27.24	0.161	4.419	3.685	2.181	3.067	2.975	5.921	9.204
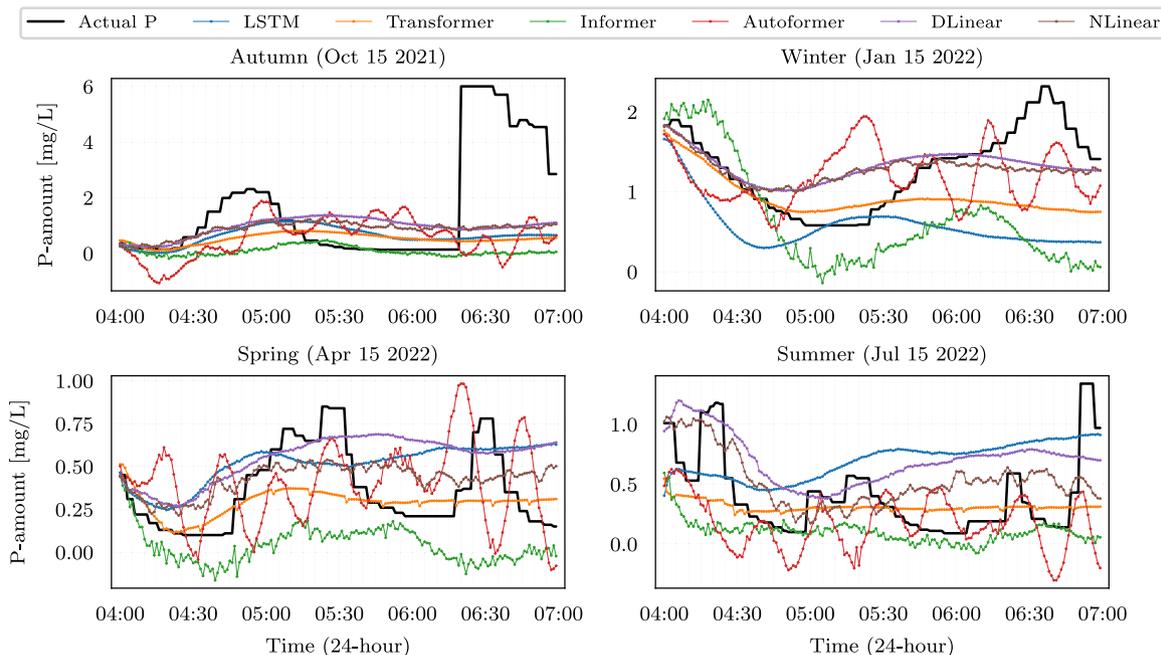
Fig. 10. Simulation environment results of all models for the four data points in different sequences.

making the extraction of the seasonal-trend components challenging. The previous studies were mainly focused on developing the *LSTM* or *Transformer* models to predict a single variable from the plant (Salles et al., 2022; Xu et al., 2023; Yaqub et al., 2020; Ly et al., 2022). In contrast, the models in this study were used to predict multiple variables as the system's future state, resulting in the models' increased dimension and output. However, the *LSTM* and *Transformer* models developed in the current study also outperform the similar architectures developed to predict wastewater treatment plant variables in past studies, as can be seen in Table 2.

The sensitivity analysis of the predictive accuracy of phosphorus amounts in the models' prediction, as shown in Fig. 9, states two notable observations: 1. The accuracy of the P amount in the predictions mainly depends on the historical trajectory of its values, and 2. Linear models exhibit limitations in capturing causal relationships among the input features at each time step.

Developing accurate simulation environments for phosphorus removal systems is a big challenge in DRL research due to the complexity and non-linearity of such systems in wastewater treatment plants. The results from this study show that even the state-of-the-art models in the field of time series forecasting might face some problems regarding biological process simulation. Fig. 10 shows the difficulties of deep learning models in predicting the phosphorus removal system's behavior. As one may notice, no specific model significantly outperforms others, and each one has more accurate or poorer predictions for particular periods.

The metrics in Table 4 show that the *NLinear* and *Autoformer* models perform the best concerning MSE and DTW, respectively. The average MSE values for all models vary significantly, indicating that linear

models provided predictions closer to the actual values at each step. In contrast, earlier models experienced fluctuations in their predictions, leading to increased MSE values. However, the average DTW values in Table 4 for various models are closely aligned, suggesting they exhibit similar capabilities in capturing the system's dynamics. Furthermore, the variation in metrics across each sequence highlights the models' responsiveness to input data, reflecting the distinct dynamics associated with date, time, and disturbances in wastewater treatment plants. Disturbances in the wastewater treatment plant might arise from numerous factors like rainfall, equipment malfunctions, or sudden changes in the inlet flow. Although these disturbances can be examined and incorporated into the system's model, they were not a primary focus of this research.

The possible main reason for inaccuracy and uncertainty in the models' predicted states is an issue called *compounding error* in reinforcement learning (Xiao et al., 2019). The compounding error refers to the increase of unacceptable one-step prediction errors over longer horizons (Xiao et al., 2019; Asadi et al., 2019; Lambert et al., 2022). According to Lambert et al. (2022), the state of a dynamical system after h episodes can be predicted by the following equation:

$$\hat{s}_{t+h} = f(...f(f(s_t, a_t), a_{t+1})..., a_{t+h}) \tag{23}$$

The prediction error for each episode can be defined as $e_t = \hat{s}_t - s_t$, increasing multiplicatively because each step's input consists of previously predicted states. This issue can be formulated like the following over a prediction horizon (episode length) of h (Lambert et al., 2022):

$$\hat{s}_{t+h} = f(...f(f(s_t, a_t) + e_t, a_{t+1}) + e_{t+1}..., a_{t+h}) + e_{t+h} \tag{24}$$
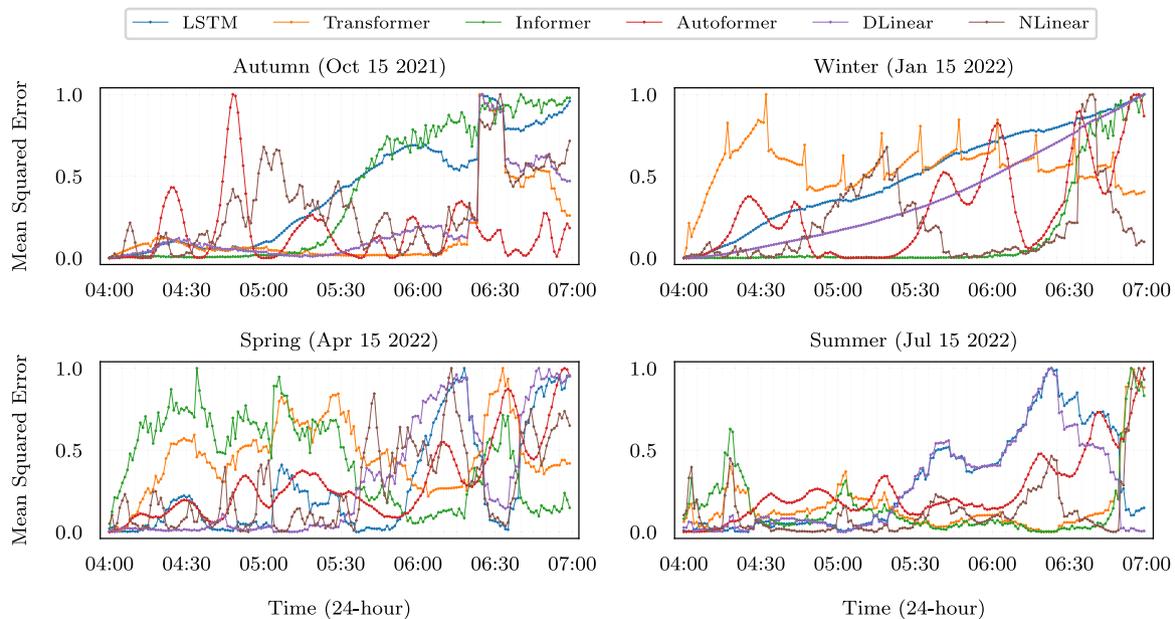
Fig. 11. Step-wise mean squared error of all models for the four data points in different sequences.

Then, the mean squared error of the prediction for each time step can be computed by Eq. (19). Fig. 11 shows the effect of compounding error in each simulation step for all models in different seasons. The figure illustrates fluctuations in the mean squared error across different models, with a notable increase in its value observed at later stages compared to earlier stages. The normalized MSE values presented in the plots range from 0, representing the most accurate predictions, to 1, indicating completely wrong predictions. These results suggest that the models may produce unrealistic predictions during extended episodes, and caution should be exercised in interpreting long-term simulations.

According to the previous descriptions, we can conclude that the compounding error issue can lead the system model toward unrealistic and uncertain predictions. This issue appears in different forms for each model, which we will discuss in the following sections.

4.1. LSTM

The *LSTM* model can capture the system's dynamics in the early steps, but as simulation goes on, it moves towards linear and constant values. Considering the issue of compounding error, the *LSTM* model could avoid extremely fluctuating predictions due to its ability to extract the information from the previous data as possible and capture long-term dependencies. In the summer sequence of Fig. 10, the *LSTM* has very close predictions to linear models, which results in the inability to capture the system's dynamics. Furthermore, Table 4 indicates that while the *LSTM* model may not exhibit the lowest average MSE, it captures the dynamics comparably to other models when analyzed via DTW.

4.2. Formers

The former models (*Transformer*, *Informer*, and *Autoformer*) are different in behavior compared to the *LSTM*. They act better in following the system dynamics but poorly predict the values. In the *Transformer*, autoregression in the decoder results in error accumulation in long-term predictions (Zeng et al., 2022). As explained in the previous section, the compounding error is the main challenge in simulating the phosphorus removal system's function. The formers also suffer from the compounding error issue, which is more different than *LSTM*. These models extract much information from the previous data at each step, leading them to extract many more wrong values. They learn to pay

attention to actual data information, which results in better predictions. Still, it will be more troublesome when this attention is applied to the predicted and non-precise data. They suffer from *too much* attention. This particular issue is notably evident in the spring sequence shown in Fig. 10 and Table 4, specifically regarding the *Autoformer* model. In this instance, while the model produces notably dynamic predictions that often exceed actual values, the DTW indicates its capability to maintain the system's dynamics, though with increased fluctuations. Additionally, as observed in Fig. 11, while former models show a rise in compounding errors over the course of the simulation, this trend is not consistently significant at every stage. This issue is linked to the black-box nature of these models, leading to unforeseeable accuracy in predictions at various simulation phases. Furthermore, a comparison between Figs. 10 and 11 reveals that despite lower mean squared errors in later simulation stages, these former models struggle to represent the system's dynamics accurately. This highlights the challenge of creating a metric that effectively assesses the dynamic representation capabilities of the simulators.

4.3. LTSF linear

DLinear and *NLinear* have more similar behavior to the *LSTM*, as they can keep the system dynamics at the beginning of the simulation. Still, they lean towards a linear constant prediction as it goes forward. On the one hand, the more simple and linear models cannot extract much information from historical data, resulting in the inability to capture system dynamics. On the other hand, their simple structure prevents them from forecasting far from the actual values, leading to a less fluctuating simulation. According to Fig. 10, this behavior can be seen in all sequences but is more obvious in the autumn and winter parts. The stochasticity of the P removal process, which results in different trends in phosphorus concentration throughout the year, is very problematic to linear models' prediction as they learn the linear relationship between input features and the target variable. Yet, Table 4 demonstrates the capability of these models in aligning their predictions with actual values, leading to low mean squared errors without resulting in high DTW values.

The linear models suffer more from the compounding error issue as they cannot get help from learned memory like *LSTM* or self-attention mechanisms like the formers to overcome the accumulation of errors. In the end, as shown in Fig. 9, it becomes evident that linear models

have problems capturing the causal relationships among the features. Consequently, they rely only on the input phosphorus levels to generate output P values, ignoring the use of causal relationships for corrective prediction enhancements.

4.4. Common issues

As the simulation behavior differs, the models share common issues in the results. The first one is that none of them captures the sudden changes in the process, which is more evident at the end of the autumn sequence (from 06:20 to 06:35) and in the summer sequence (from 06:50 to 07:00) in Fig. 10. The mentioned issue results in higher DTW values for all models, as shown in Table 4 in the Autumn compared to the other sequences. These changes can be related to sensor failures and inaccurate data from the actual plant, which is unknown, and there is no possibility of including them in the models. Even though the models were trained on the dataset with such inaccurate data, they were successful in not following the wrong dynamics.

One potential issue arises in the summer sequence, where the initial prediction differs from the actual observation. This disparity highlights the possibility of models generating slightly wrong predictions at the onset of the simulation, particularly when the entire input sequence is derived from the actual dataset. Consequently, models that begin with less precise predictions may exhibit inferior performance in subsequent episodes compared to those that initially generated accurate predictions.

While the models may generate similar values that differ slightly in dynamics, there are instances where the predicted values can significantly diverge. Such a scenario is apparent in the winter sequence of Fig. 10 and Table 4, indicating that the behavior of models relative to one another can vary depending on the input sequence.

Finally, selecting the optimal models based on Table 4 is a significant challenge. As can be seen from the table, the leading models for the simulation environment can differ according to the metric under consideration. The MSE monitors how accurately the models predict actual values at each sequence, whereas DTW assesses how well they simulate the dynamics of the system. Models that perform well in MSE might not consistently outperform in predicting the system's dynamics. Some models may attempt to forecast a linear average of actual values within a sequence to align closely with real data to explain this inconsistency.

5. Conclusions

The lack of accurate simulators presents a significant challenge in implementing Deep Reinforcement Learning for chemical and biological processes. In this study, the authors trained six different state-of-the-art time series forecasting models to create a simulator for the DRL environment. The findings of this study emphasized the following:

- The models demonstrated over 97% accuracy in one-step-ahead predictions on the test dataset.
- Sensitivity analysis of phosphorus amounts indicated that the models, especially linear ones, rely heavily on historical P amounts for making predictions.
- The performance of these models was compromised over longer time horizons in multi-step simulations due to uncertainty and incorrect prediction behavior.
- The *LSTM* and *Linear* models tended to produce linear predictions over time, failing to represent the system's dynamics accurately.
- The *Former* models exhibited more variability and fluctuating behavior as the simulation progressed.
- The use of these models in multi-step simulations highlighted limitations such as compounding errors and the inability to capture industrial processes' complex dynamics accurately.

Future research can focus on strategies such as developing hybrid models, preparing an input of less noisy data, and using different model training methods to improve the simulation environment results. Moreover, several deep reinforcement learning algorithms, including policy gradient methods, offer potential as promising methods for training control policies in imperfect simulation environments where it is challenging to train algorithms directly within a real-world environment. Ultimately, the proposed approach of creating simulation environments for DRL algorithms, utilizing SCADA data with a sufficient historical horizon to capture all system dynamics, is a promising way to improve process control in industrial applications.

CRediT authorship contribution statement

Esmaeel Mohammadi: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Software, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mikkel Stokholm-Bjerregaard:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing. **Aviaja Anna Hansen:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – review & editing. **Per Halkjær Nielsen:** Conceptualization, Supervision, Methodology. **Daniel Ortiz-Arroyo:** Formal analysis, Methodology, Validation, Writing – review & editing. **Petar Durdevic:** Conceptualization, Formal analysis, Methodology, Supervision, Validation, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Esmaeel Mohammadi reports financial support and administrative support were provided by Horizon Europe. The RecaP project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956454.

Data availability

The data can be available in the future after getting consent from the plant.

Acknowledgments

References

Åström, K., Wittenmark, B., 2013. Computer-Controlled Systems: Theory and Design, third ed. In: Dover Books on Electrical Engineering, Dover Publications.

Acevedo, B., Borrás, L., Oehmen, A., Barat, R., 2014. Modelling the metabolic shift of polyphosphate-accumulating organisms. Water Res. 65, 235–244.

Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M., 2019. Optuna: A next-generation hyperparameter optimization framework. arXiv:1907.10902.

Al, R., Behera, C.R., Zubov, A., Gernaey, K.V., Sin, G., 2019. Meta-modeling based efficient global sensitivity analysis for wastewater treatment plants – An application to the BSM2 model. Comput. Chem. Eng. 127, 233–246. http://dx.doi.org/10.1016/j.compchemeng.2019.05.015, URL https://www.sciencedirect.com/science/article/pii/S0098135419301462.

Alex, J., Benedetti, L., Copp, J., Gernaey, K., Jeppsson, U., Nopens, I., Pons, M., Rieger, L., Rosen, C., Steyer, J., et al., 2008. Benchmark Simulation Model No. 1 (bsm1). Report by the IWA Taskgroup on Benchmarking of Control Strategies for Wwtps 1, Department of Industrial Electrical Engineering and Automation, Lund.

Anon, 2021. Blue kolding 2021. https://bluekolding.dk/. (Accessed: 03 January 2023).

Asadi, K., Misra, D., Kim, S., Littman, M.L., 2019. Combating the compounding-error problem with a multi-step model. http://dx.doi.org/10.48550/ARXIV.1905.13320, arXiv.

Bahdanau, D., Cho, K., Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Bao, Y., Zhu, Y., Qian, F., 2021. A deep reinforcement learning approach to improve the learning performance in process control. Ind. Eng. Chem. Res. 60 (15), 5504–5515. http://dx.doi.org/10.1021/acs.iecr.0c05678.

Batstone, D.J., Keller, J., Angelidaki, I., Kalyuzhnyi, S.V., Pavlostathis, S.G., Rozzi, A., Sanders, W.T.M., Siegrist, H., Vavilin, V.A., 2002. Anaerobic Digestion Model No.1 (ADM1). In: Scientific and Technical Report Series, IWA Publishing.

Berthouex, P., Box, G.E., 1996. Time series models for forecasting wastewater treatment plant performance. Water Res. 30 (8), 1865–1875. http://dx.doi.org/10.1016/0043-1354(96)00063-2.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W., 2016. Openai gym. arXiv preprint arXiv:1606.01540.

Brunton, S.L., Kutz, J.N., 2019. Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control. Cambridge University Press, http://dx.doi.org/10.1017/9781108380690.

Bunce, J.T., Ndam, E., Ofiteru, I.D., Moore, A., Graham, D.W., 2018. A review of phosphorus removal technologies and their applicability to small-scale domestic wastewater treatment systems. Front. Environ. Sci. 6, 8.

Burton, F.L., Stensel, H.D., Tchobanoglous, G., 2014. Wastewater Engineering : Treatment and Resource Recovery, fifth ed. McGraw-Hill Higher Education, New York.

Chang, P., Li, Z., 2021. Over-complete deep recurrent neutral network based on wastewater treatment process soft sensor application. Appl. Soft Comput. 105, 107227. http://dx.doi.org/10.1016/j.asoc.2021.107227.

Chen, K., Wang, H., Valverde-Pérez, B., Zhai, S., Vezzaro, L., Wang, A., 2021. Optimal control towards sustainable wastewater treatment plants based on multi-agent reinforcement learning. Chemosphere 279, 130498. http://dx.doi.org/10.1016/j.chemosphere.2021.130498.

Cheng, T., Harrou, F., Kadri, F., Sun, Y., Leiknes, T., 2020. Forecasting of wastewater treatment plant key features using deep learning-based models: A case study. IEEE Access 8, 184475–184485. http://dx.doi.org/10.1109/ACCESS.2020.3030820.

Cheng, F., Zhao, J., 2019. A novel process monitoring approach based on feature points distance dynamic autoencoder. In: Kiss, A.A., Zondervan, E., Lakerveld, R., Özkan, L. (Eds.), 29th European Symposium on Computer Aided Process Engineering. In: Computer Aided Chemical Engineering. vol. 46, Elsevier, pp. 757–762. http://dx.doi.org/10.1016/B978-0-12-818634-3.50127-2.

Chong, M.N., Ho, A.N., Gardner, T., Sharma, A.K., Hood, B., 2013. Assessing decentralised wastewater treatment technologies: Correlating technology selection to system robustness, energy consumption and GHG emission. J. Water Clim. Change 4 (4), 338–347.

Cocianu, C.L., Uscatu, C.R., Avramescu, M., 2022. Improvement of LSTM-based forecasting with NARX model through use of an evolutionary algorithm. Electronics 11 (18), http://dx.doi.org/10.3390/electronics11182935.

Dang, L.M., Wang, H., Li, Y., Nguyen, T.N., Moon, H., 2022. DefectTR: End-to-end defect detection for sewage networks using a transformer. Constr. Build. Mater. 325, 126584. http://dx.doi.org/10.1016/j.conbuildmat.2022.126584.

Faizi, N., Alvi, Y., 2023. Chapter 6 - correlation. In: Faizi, N., Alvi, Y. (Eds.), Biostatistics Manual for Health Research. Academic Press, pp. 109–126. http://dx.doi.org/10.1016/B978-0-443-18550-2.00002-5, URL https://www.sciencedirect.com/science/article/pii/B9780443185502000025.

Foscoliano, C., Del Vigo, S., Mulas, M., Tronci, S., 2016. Predictive control of an activated sludge process for long term operation. Chem. Eng. J. 304, 1031–1044.

Gaitang, H., Junfei, Q., Honggui, H., 2016. Wastewater treatment control method based on recurrent fuzzy neural network. CIESC J. 67 (3), 954. http://dx.doi.org/10.11949/j.issn.0438-1157.20151898.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., Schmidhuber, J., 2009. A novel connectionist system for unconstrained handwriting recognition. IEEE Trans. Pattern Anal. Mach. Intell. 31 (5), 855–868. http://dx.doi.org/10.1109/TPAMI.2008.137.

Gu, S., Fu, B., Ahn, J.-W., Fang, B., 2021. Mechanism for phosphorus removal from wastewater with fly ash of municipal solid waste incineration, Seoul, Korea. J. Clean. Prod. 280, 124430. http://dx.doi.org/10.1016/j.jclepro.2020.124430.

Gujer, W., 2006. Activated sludge modelling: Past, present and future. Water Sci. Technol. 53 (3), 111–119. http://dx.doi.org/10.2166/wst.2006.082, arXiv:https://iwaponline.com/wst/article-pdf/53/3/111/432421/111.pdf.

Hansen, L.D., Stokholm-Bjerregaard, M., Durdevic, P., 2022. Modeling phosphorous dynamics in a wastewater treatment process using Bayesian optimized LSTM. Comput. Chem. Eng. 160, 107738. http://dx.doi.org/10.1016/j.compchemeng.2022.107738.

Henze, M., Gujer, W., Mino, T., van Loosedrecht, M., 2006. Activated Sludge Models ASM1, ASM2, ASM2d and ASM3. IWA Publishing, http://dx.doi.org/10.2166/9781780402369.

Henze, M., Gujer, W., Mino, T., van Loosdrecht, M., 2000. Activated Sludge Models ASM1, ASM2, ASM2d and ASM3, reprint ed. IWA Publishing.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9, 1735–1780.

Hua, J., Zeng, L., Li, G., Ju, Z., 2021. Learning for a robot: Deep reinforcement learning, imitation learning, transfer learning. Sensors 21 (4), 1278.

Huang, C.-Z.A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., Dai, A.M., Hoffman, M.D., Dinculescu, M., Eck, D., 2018. Music transformer. arXiv preprint arXiv:1809.04281.

Huang, Z.-l., Zhang, L.-y., Zhang, Y., Qian, S.-w., Wang, C.-j., 2021. Transformer based multi-output regression learning for wastewater treatment. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence. ICTAI, pp. 698–703. http://dx.doi.org/10.1109/ICTAI52525.2021.00110.

Ikumi, D., Brouckaert, C., Ekama, G., 2011. Modelling of struvite precipitation in anaerobic digestion. In: Proc. IWA Watermatex2011, San Sebastian, Spain, vol. 20, e22.

Ikumi, D., Harding, T., Ekama, G., 2014. Biodegradability of wastewater and activated sludge organics in anaerobic digestion. Water Res. 56, 267–279. http://dx.doi.org/10.1016/j.watres.2014.02.008.

Iwanaga, T., Usher, W., Herman, J., 2022. Toward SALib 2.0: Advancing the accessibility and interpretability of global sensitivity analyses. Socio-Environ. Syst. Model. 4, 18155. http://dx.doi.org/10.18174/sesmo.18155, URL https://sesmo.org/article/view/18155.

Kang, H., Yang, S., Huang, J., Oh, J., 2020. Time series prediction of wastewater flow rate by bidirectional LSTM deep learning. Int. J. Control Autom. Syst. 18 (12), 3023–3030.

Kazadi Mbamba, C., Lindblom, E., Flores-Alsina, X., Tait, S., Anderson, S., Saagi, R., Batstone, D., Gernaey, K., Jeppsson, U., 2019. Plant-wide model-based analysis of iron dosage strategies for chemical phosphorus removal in wastewater treatment systems. Water Res. 155, 12–25. http://dx.doi.org/10.1016/j.watres.2019.01.048, URL https://www.sciencedirect.com/science/article/pii/S0043135419301046.

Khandelwal, U., He, H., Qi, P., Jurafsky, D., 2018. Sharp nearby, fuzzy far away: How neural language models use context. arXiv preprint arXiv:1805.04623.

Lai, G., Chang, W.-C., Yang, Y., Liu, H., 2018. Modeling long-and short-term temporal patterns with deep neural networks. In: The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 95–104.

Lambert, N., Pister, K., Calandra, R., 2022. Investigating compounding prediction errors in learned dynamics models. http://dx.doi.org/10.48550/ARXIV.2203.09637, arXiv.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2015. Continuous control with deep reinforcement learning. http://dx.doi.org/10.48550/ARXIV.1509.02971.

Ly, Q.V., Truong, V.H., Ji, B., Nguyen, X.C., Cho, K.H., Ngo, H.H., Zhang, Z., 2022. Exploring potential machine learning application based on big data for prediction of wastewater quality from different full-scale wastewater treatment plants. Sci. Total Environ. 832, 154930. http://dx.doi.org/10.1016/j.scitotenv.2022.154930, URL https://www.sciencedirect.com/science/article/pii/S004896972202023X.

Malviya, A., Jaspal, D., 2021. Artificial intelligence as an upcoming technology in wastewater treatment: A comprehensive review. Environ. Technol. Rev. 10 (1), 177–187. http://dx.doi.org/10.1080/21622515.2021.1913242.

Mannina, G., Rebouças, T.F., Cosenza, A., Sànchez-Marrè, M., Gibert, K., 2019. Decision support systems (DSS) for wastewater treatment plants – A review of the state of the art. Bioresour. Technol. 290, 121814. http://dx.doi.org/10.1016/j.biortech.2019.121814.

Maravelias, C.T., Sung, C., 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. Comput. Chem. Eng. 33 (12), 1919–1930.

Melgaço, L., Robles-Aguilar, A., Meers, E., Mota, C., 2021. Phosphorus recovery from liquid digestate by chemical precipitation using low-cost ion sources. J. Chem. Technol. Biotechnol. 96 (10), 2891–2900. http://dx.doi.org/10.1002/jctb.6842, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/jctb.6842.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Mockus, J., 2012. Bayesian Approach to Global Optimization: Theory and Applications. In: Mathematics and its Applications, Springer Netherlands.

Moriyama, T., Magistris, G.D., Tatsubori, M., Pham, T.-H., Munawar, A., Tachibana, R., 2018. Reinforcement learning testbed for power-consumption optimization. In: Asian Simulation Conference. Springer, pp. 45–59.

Müller, M., 2007. Dynamic Time Warping. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 69–84. http://dx.doi.org/10.1007/978-3-540-74048-3_4, Ch. Dynamic Time Warping.

Nelles, O., 2020. Nonlinear System Identification: From Classical Approaches to Neural Networks, Fuzzy Models, and Gaussian Processes. Springer Nature.

Newhart, K.B., Holloway, R.W., Hering, A.S., Cath, T.Y., 2019. Data-driven performance analyses of wastewater treatment plants: A review. Water Res. 157, 498–513. http://dx.doi.org/10.1016/j.watres.2019.03.030.

Nian, R., Liu, J., Huang, B., 2020. A review on reinforcement learning: Introduction and applications in industrial process control. Comput. Chem. Eng. 139, 106886. http://dx.doi.org/10.1016/j.compchemeng.2020.106886.

Nian, R., Liu, J., Huang, B., Mutasa, T., 2019. Fault tolerant control system: A reinforcement learning approach. SICE 1010–1015.

Nobaharan, K., Bagheri Novair, S., Asgari Lajayer, B., van Hullebusch, E.D., 2021. Phosphorus removal from wastewater: The potential use of biochar and the key controlling factors. Water 13 (4), http://dx.doi.org/10.3390/w13040517.

Ömer Faruk, D., 2010. A hybrid neural network and ARIMA model for water quality time series prediction. Eng. Appl. Artif. Intell. 23 (4), 586–594. http://dx.doi.org/10.1016/j.engappai.2009.09.015.

Ostace, G.S., Baeza, J.A., Guerrero, J., Guisasola, A., Cristea, V.M., Agachi, P.Ş., Lafuente, J., 2013. Development and economic assessment of different WWTP control strategies for optimal simultaneous removal of carbon, nitrogen and phosphorus. Comput. Chem. Eng. 53, 164–177. http://dx.doi.org/10.1016/j.compchemeng.2013.03.007.

Pang, J.-W., Yang, S.-S., He, L., Chen, Y.-D., Cao, G.-L., Zhao, L., Wang, X.-Y., Ren, N.-Q., 2019. An influent responsive control strategy with machine learning: Q-learning based optimization method for a biological phosphorus removal system. Chemosphere 234, 893–901. http://dx.doi.org/10.1016/j.chemosphere.2019.06.103.

Park, S.-H., Koo, J., 2015. Application of transfer function ARIMA modeling for the sedimentation process on water treatment plant. Int. J. Control Autom. 8 (10), 135–144.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (Eds.), In: Advances in Neural Information Processing Systems, vol. 32, Curran Associates, Inc., pp. 8024–8035, URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Peng, Z., Dang, J., Unoki, M., Akagi, M., 2021. Multi-resolution modulation-filtered cochleagram feature for LSTM-based dimensional emotion recognition from speech. Neural Netw. 140, 261–273. http://dx.doi.org/10.1016/j.neunet.2021.03.027.

Pisa, I., Morell, A., Vicario, J.L., Vilanova, R., 2020. Denoising autoencoders and LSTM-based artificial neural networks data processing for its application to internal model control in industrial environments–the wastewater treatment plant control case. Sensors 20 (13), http://dx.doi.org/10.3390/s20133743.

Pisa, I., Santín, I., Morell, A., Vicario, J.L., Vilanova, R., 2019a. LSTM-based wastewater treatment plants operation strategies for effluent quality improvement. IEEE Access 7, 159773–159786. http://dx.doi.org/10.1109/ACCESS.2019.2950852.

Pisa, I., Santín, I., Vicario, J.L., Morell, A., Vilanova, R., 2019b. ANN-based soft sensor to predict effluent violations in wastewater treatment plants. Sensors 19 (6), 1280.

Pisa, I., Vilanova, R., Santín, I., Vicario, J.L., Morell, A., 2019c. Artificial neural networks application to support plant operation in the wastewater industry. In: Technological Innovation for Industry and Service Systems: 10th IFIP WG 5.5/SO-COLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2019, Costa de Caparica, Portugal, May 8–10, 2019, Proceedings 10. Springer, pp. 257–265.

Porter, R., FitzSimons, D., 2009. Phosphorus in the Environment: Its Chemistry and Biochemistry. In: Novartis Foundation Symposia, Wiley.

Raju, L., Milton, R., Suresh, S., Sankar, S., 2015. Reinforcement learning in adaptive control of power system generation. Procedia Comput. Sci. 46, 202–209.

Revollar, S., Vega, P., Vilanova, R., Francisco, M., 2017. Optimal control of wastewater treatment plants using economic-oriented model predictive dynamic strategies. Appl. Sci. 7 (8), http://dx.doi.org/10.3390/app7080813.

Sak, H., Senior, A., Beaufays, F., 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv:1402.1128.

Salles, R., Mendes, J., Araújo, R., Melo, C., Moura, P., 2022. Prediction of key variables in wastewater treatment plants using machine learning models. In: 2022 International Joint Conference on Neural Networks. IJCNN, pp. 1–9. http://dx.doi.org/10.1109/IJCNN55064.2022.9892661.

Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S., 2010. Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. Comput. Phys. Comm. 181 (2), 259–270. http://dx.doi.org/10.1016/j.cpc.2009.09.018, URL https://www.sciencedirect.com/science/article/pii/S0010465509003087.

Seviour, R.J., Mino, T., Onuki, M., 2003. The microbiology of biological phosphorus removal in activated sludge systems. FEMS Microbiol. Rev. 27 (1), 99–127.

Shaw, P., Uszkoreit, J., Vaswani, A., 2018. Self-attention with relative position representations. arXiv preprint arXiv:1803.02155.

Sheik, A.G., Machavolu, V.R.K., Seepana, M.M., Ambati, S.R., 2022. Integrated supervisory and override control strategies for effective biological phosphorus removal and reduced operational costs in wastewater treatment processes. Chemosphere 287, 132346. http://dx.doi.org/10.1016/j.chemosphere.2021.132346.

Shuang, K., Tan, Y., Cai, Z., Sun, Y., 2020. Natural language modeling with syntactic structure dependency. Inform. Sci. 523, 220–233. http://dx.doi.org/10.1016/j.ins.2020.03.022.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al., 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815.

Spielberg, S., Gopaluni, R., Loewen, P., 2017. Deep reinforcement learning approaches for process control. In: 2017 6th International Symposium on Advanced Control of Industrial Processes. AdCONIP, IEEE, pp. 201–206.

Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.

Tchobanoglous, G., Burton, F., Stensel, H., Metcalf & Eddy, Inc., Burton, F., 2003. Wastewater engineering: Treatment and reuse. McGraw-Hill higher education, McGraw-Hill Education.

The pandas development team, 2020. pandas-dev/pandas: Pandas. Zenodo, http://dx.doi.org/10.5281/zenodo.3509134.

Tuszynska, A., Kaszubowska, M., Kowal, P., Ciesielski, S., Makinia, J., 2019. The metabolic activity of denitrifying microorganisms accumulating polyphosphate in response to addition of fusel oil. Bioprocess Biosyst. Eng. 42, http://dx.doi.org/10.1007/s00449-018-2022-0.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30.

Vinyals, O., Babuschkin, I., Czarnecki, W.M., Mathieu, M., Dudzik, A., Chung, J., Choi, D.H., Powell, R., Ewalds, T., Georgiev, P., et al., 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575 (7782), 350–354.

Wang, Y., 2017. A new concept using LSTM neural networks for dynamic system identification. In: 2017 American Control Conference. ACC, pp. 5324–5329. http://dx.doi.org/10.23919/ACC.2017.7963782.

Wang, Y., Velswamy, K., Huang, B., 2018. A novel approach to feedback control with deep reinforcement learning. IFAC-PapersOnLine 51 (18), 31–36.

Welch, E., Lindell, T., 1980. Ecological Effects of Waste Water. Cambridge University Press.

Wu, H., Xu, J., Wang, J., Long, M., 2021. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. Adv. Neural Inf. Process. Syst. 34, 22419–22430.

Wunsch, A., Liesch, T., Broda, S., 2018. Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). J. Hydrol. 567, 743–758.

Xiao, C., Wu, Y., Ma, C., Schuurmans, D., Müller, M., 2019. Learning to combat compounding-error in model-based reinforcement learning. http://dx.doi.org/10.48550/ARXIV.1912.11206, arXiv.

Xu, H., Vilanova, R., 2015. Application of fuzzy control on wastewater treatment plant for P-removal. In: 2015 23rd Mediterranean Conference on Control and Automation. MED, IEEE, pp. 545–550.

Xu, Y., Wang, Z., Nairat, S., Zhou, J., He, Z., 2023. Artificial intelligence-assisted prediction of effluent phosphorus in a full-scale wastewater treatment plant with missing phosphorus input and removal data. ACS ES&T Water http://dx.doi.org/10.1021/acsestwater.2c00517.

Yaqub, M., Asif, H., Kim, S., Lee, W., 2020. Modeling of a full-scale sewage treatment plant to predict the nutrient removal efficiency using a long short-term memory (LSTM) neural network. J. Water Process Eng. 37, 101388. http://dx.doi.org/10.1016/j.jwpe.2020.101388, URL https://www.sciencedirect.com/science/article/pii/S221471442030266X.

Ye, Z., Yang, J., Zhong, N., Tu, X., Jia, J., Wang, J., 2020. Tackling environmental challenges in pollution controls using artificial intelligence: A review. Sci. Total Environ. 699, 134279. http://dx.doi.org/10.1016/j.scitotenv.2019.134279.

Yunpeng, L., Di, H., Junpeng, B., Yong, Q., 2017. Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network. In: 2017 14th Web Information Systems and Applications Conference. WISA, pp. 305–310. http://dx.doi.org/10.1109/WISA.2017.25.

Zarzycki, K., Ławryńczuk, M., 2021. LSTM and GRU neural networks as models of dynamical processes used in predictive control: A comparison of models developed for two chemical reactors. Sensors 21 (16), 5625. http://dx.doi.org/10.3390/s21165625.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2022. Are transformers effective for time series forecasting? arXiv preprint arXiv:2205.13504.

Zhang, X., Ding, Y., Zhao, H., Yi, L., Guo, T., Li, A., Zou, Y., 2024. Mixed skewness probability modeling and extreme value predicting for physical system input–output based on full Bayesian generalized maximum-likelihood estimation. IEEE Trans. Instrum. Meas. 73, 1–16. http://dx.doi.org/10.1109/TIM.2023.3343742.

Zhang, C., Guisasola, A., Baeza, J.A., 2022. A review on the integration of mainstream P-recovery strategies with enhanced biological phosphorus removal. Water Res. 212, 118102. http://dx.doi.org/10.1016/j.watres.2022.118102.

Zhang, D., Martinez, N., Lindholm, G., Ratnaweera, H., 2018. Manage sewer in-line storage control using hydraulic model and recurrent neural network. Water Resour. Manag. 32 (6), 2079–2098.

Zhao, L., Dai, T., Qiao, Z., Sun, P., Hao, J., Yang, Y., 2020. Application of artificial intelligence to wastewater treatment: A bibliometric analysis and systematic review of technology, economy, management, and wastewater reuse. Process Saf. Environ. Prot. 133, 169–182. http://dx.doi.org/10.1016/j.psep.2019.11.014.

Zhao, H.-W., Ding, Y.-L., Li, A.-Q., Chen, B., Wang, K.-P., 2023a. Digital modeling approach of distributional mapping from structural temperature field to temperature-induced strain field for bridges. J. Civ. Struct. Health Monit. 13 (1), 251–267. http://dx.doi.org/10.1007/s13349-022-00635-8.

Zhao, H., Ding, Y., Meng, L., Qin, Z., Yang, F., Li, A., 2023b. Bayesian multiple linear regression and new modeling paradigm for structural deflection robust to data time lag and abnormal signal. IEEE Sens. J. 23 (17), 19635–19647. http://dx.doi.org/10.1109/JSEN.2023.3294912.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., Jin, R., 2022. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. arXiv:2201. 12740.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W., 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. arXiv:2012. 07436.

Zhu, R., Tu, X., Xiangji Huang, J., 2020. Chapter seven - deep learning on information retrieval and its applications. In: Das, H., Pradhan, C., Dey, N. (Eds.), Deep Learning for Data Analytics. Academic Press, pp. 125–153. http://dx.doi.org/10.1016/B978-0-12-819764-6.00008-9.