# Multi-Step Simulation Improvement for Time Series Using Exogenous State Variables


Esmaeel Mohammadi[1,2,a], Daniel Ortiz-Arroyo[3,b], Mikkel Stokholm-Bjerregaard[1,c] and Petar Durdevic[3,d]

[1] *Krüger A/S, Indkildevej 6C, Aalborg, 9210, North Jutland, Denmark*

[2] *Department of Chemistry and Bioscience, Aalborg University, Fredrik Bajers Vej 7H, Aalborg, 9220, North Jutland, Denmark*

[3] *AAU Energy, Aalborg University, Niels Bohrs vej 8, Esbjerg, 6700, South Jutland, Denmark*



Keywords: Deep Reinforcement Learning, Dynamic Model, Simulator, LSTM, Exogenous, Phosphorus.



Abstract: Accurate simulation of wastewater treatment systems is essential for optimizing control strategies and ensuring efficient operation. This study focuses on enhancing the predictive accuracy of a Long Short-Term Memory (LSTM)-based simulator by incorporating exogenous state variables, such as temperature, flow, and process phases, that are independent of output and control variables. The experimental results demonstrate that including these variables significantly reduces prediction errors, measured by Mean Squared Errors (MSE) and Dynamic Time Warping (DTW) metrics. The improved model, particularly the version that uses actual values of exogenous state variables at each simulation step, showed robust performance across different seasons, reducing MSE by **55%** and DTW by **34%** compared to the model which didn't include exogenous state variables. This approach addresses the compounding error issue in multi-step simulations, leading to more reliable predictions and enhanced operational efficiency in wastewater treatment.




## Glossary of Terms and Acronyms

- **LSTM**: Long Short-Term Memory
- **MSE**: Mean Squared Error
- **DTW**: Dynamic Time Warping
- **WWTP**: Wastewater Treatment Plant
- **DAD**: DATA AS DEMONSTRATOR

## 1 INTRODUCTION

Accurate simulation of wastewater treatment systems' behavior is essential for optimizing control strategies and ensuring efficient operation. Based on mathematical and statistical approaches, traditional simulation models often face challenges in prediction accuracy due to the non-linear, stochastic, and non-stationary nature of system dynamics (Gujer et al., 1995; Hansen et al., 2024; Hansen et al., 2022). Factors such as varying influent characteristics and operational conditions add complexity, making accurate predictions challenging (Gujer et al., 1995; Mohammadi et al., 2024c; Hansen et al., 2022).

Recent advancements in deep learning, particularly Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997a), have shown promise in modeling complex time series data (Mohammadi et al., 2024c; Hansen et al., 2022). LSTMs can learn long-term dependencies, making them suitable for predicting the behavior of complex systems like wastewater treatment plants. However, one significant challenge in developing a multi-step simulation environment using deep learning models is the accumulation of errors, known as compounding errors (Mohammadi et al., 2024c; Mohammadi et al., 2024a).

The previous work by (Gao et al., 2023) incorporated exogenous variables within LSTM models for multivariate time series prediction, focusing on using Neural ODEs to enhance prediction smoothness and interpretability. However, their approach did not directly address the critical issue of compounding errors


[a] https://orcid.org/0000-0001-7109-7944
[b] https://orcid.org/0000-0002-1297-3702
[c] https://orcid.org/0000-0002-3714-5137
[d] https://orcid.org/0000-0003-2701-9257





Mohammadi, E., Ortiz-Arroyo, D., Stokholm-Bjerregaard, M. and Durdevic, P.
Multi-Step Simulation Improvement for Time Series Using Exogenous State Variables.
DOI: 10.5220/0012927300003822



In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 1*, pages 651-659
ISBN: 978-989-758-717-7; ISSN: 2184-2809

in multi-step forecasting, which is particularly relevant in dynamic systems like wastewater treatment.

This study differentiates itself by specifically targeting the reduction of compounding errors in multistep simulations by strategically including exogenous variables. Building on the iterative improvement method proposed in (Mohammadi et al., 2024a), our approach integrates exogenous variables directly into the model's predictive framework. Unlike previous studies, we explicitly evaluate and compare scenarios with and without exogenous variables, quantifying their impact on prediction accuracy and error propagation over time.

An improved *LSTM*-based simulator was developed to train deep reinforcement learning algorithms for wastewater treatment, focusing on the phosphorus removal process. The *LSTM* model was used as a benchmark to investigate the improvement of the simulation using the proposed method. The core contribution is incorporating the exogenous state variables into the *LSTM* model and using their actual values at each step of the multi-step simulation. This method aims to enhance prediction accuracy by preventing the accumulation of prediction errors over multiple steps. The experimental results demonstrated that the incorporation of the exogenous variables improves the simulation accuracy by **55%** and **34%** in terms of mean squared errors (MSE) and dynamic time warping (DTW), respectively.

### 1.1 Contributions

This paper introduces several key contributions to the field of time series prediction and multi-step simulation:

- **An Improved Multi-step Forecasting Model:** We incorporate exogenous variables like temperature, flow, and process phases, enhancing the accuracy of long-term predictions by reducing compounding errors in multi-step simulations.
- **Extensive Experimental Validation:** Our method, validated with real-world wastewater treatment data, shows significant improvements over state-of-the-art models in MSE and DTW metrics.
- **Improved Simulation for Reinforcement Learning:** The enhanced LSTM-based simulator improves the training of deep reinforcement learning algorithms, enabling more effective control strategies in wastewater treatment processes.

## 2 METHODS

This section provides an overview of our research data and methods. It details the wastewater treatment plant data, including exogenous state variables, and presents a mathematical proof of reduced multistep error with these variables. Finally, it outlines the *LSTM* model structure, training procedure, experimental design, and the hardware and software used.

### 2.1 The Plant and Dataset

This study focuses on the data from Kolding Central WWTP in Agtrup, Denmark. The time-series dataset for two years was collected through the Hubgrade$^{TM}$ Performance Plant system, designed by Krüger/Veolia (Krüger A/S, 2023). Data preprocessing played a crucial role in enhancing model performance. The raw wastewater treatment data was initially normalized using the Min-Max technique, scaling the features from 0 to 1. Feature selection was guided by principal component and correlation analysis. Variables of the system that demonstrated the highest correlation with the target variable, Phosphate concentration, were selected as inputs for the model, in conjunction with the target variable itself and the action variable, Metal dosage. Details about the plant, dataset, and preprocessing are provided in (Mohammadi et al., 2024b; Mohammadi et al., 2024c).

### 2.2 Incorporation of Exogenous State Variables

Different variables in the plant and their impact on the accuracy of the phosphorus removal process simulation were investigated. To do this, two dataset variations, named *IOPPo* and *IOPTQCfFiFoPo*, were created, each incorporating a unique combination of exogenous and state variables (detailed in Table 1). The primary distinction between these datasets is the inclusion or exclusion of the exogenous state variables. This approach analyzed how including the exogenous state variables affects the simulation's accuracy.

In control theory, the state-space equations consist of two main components: the state and output. The state equation is given by (Hespanha, 2018; Goodwin et al., 2001):

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t) + E\mathbf{w}(t) \tag{1}$$

In this equation, $\mathbf{x}(t)$ represents the state vector, which encapsulates the internal condition of the system at time $t$. The matrix $A$ is known as the state matrix and defines the dynamics of the state vector $\mathbf{x}(t)$

Table 1: Notations for the variables used in datasets. The type describes a control variable (C), exogenous variable (E), and objective variable (O) (Mohammadi et al., 2024b).

| Symbol | Type | Description | Unit |
|---|---|---|---|
| I | C | Flow of the Iron to the biology tanks | $L/h$ |
| O | C | Dissolved oxygen | $mg/L$ |
| P | C | Flow of the PAX to the settler | $L/h$ |
| T | E | Temperature of the biology tank | $°C$ |
| Q | E | Flow of the wastewater to the biology tank | $m^3/h$ |
| Cf | E | Maximum critical function percentage | % |
| Fi | E | Process phase at the inlet (tank 1 or 2). | - |
| Fo | E | Process phase at the outlet (tank 1 or 2). | - |
| Po | O | Phosphate concentration in the biology tank | $mg/L$ |

without any external input. The vector $\mathbf{u}(t)$ is the control input, representing the actions taken to control the system, and $B$ is the input matrix describing how the control inputs affect the state vector. The term $E\mathbf{w}(t)$ includes the exogenous state variables $\mathbf{w}(t)$, which are external disturbances or inputs that affect the system, with $E$ being the matrix that shows how these disturbances impact the state vector. The output equation is given by:

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\nu(t) \tag{2}$$

Here, $\mathbf{y}(t)$ is the output vector, representing the measurable outputs of the system. The matrix $C$ is the output matrix, which maps the state vector $\mathbf{x}(t)$ to the output $\mathbf{y}(t)$. The vector $\nu(t)$, which is the output noise, also influences the output through the feedforward matrix $D$.

Assume a system where the state, control, and exogenous state variables at each time step $t$ are represented by $\mathbf{x}(t) \in \mathbb{R}^{n_x}$, $\mathbf{w}(t) \in \mathbb{R}^{n_w}$ and $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, The state vector $\mathbf{x}(t)$ can be partitioned into two components:

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{x}_c(t) \\ \mathbf{x}_e(t) \end{bmatrix} \tag{3}$$

where $\mathbf{x}_c(t) \in \mathbb{R}^{n_c}$ are objective, or the controllable state variables affected by the control inputs $\mathbf{u}(t)$. Moreover, $\mathbf{x}_e(t) \in \mathbb{R}^{n_e}$ are the exogenous state variables not affected by $\mathbf{u}(t)$ but affecting the objective variables. The state-space equations considering exogenous state variables can be written as:

$$\begin{cases} \mathbf{x}_c(t+1) = A_c\mathbf{x}_c(t) + B_c\mathbf{u}(t) + E_c\mathbf{x}_e(t) \\ \mathbf{x}_e(t+1) = A_e\mathbf{x}_e(t) + \mathbf{w}(t) \end{cases} \tag{4}$$

In these equations, $A_c \in \mathbb{R}^{n_c \times n_c}$ is the state matrix for the objective variables, defining their dynamics. The matrix $B_c \in \mathbb{R}^{n_c \times n_u}$ is the input matrix describing how the control inputs affect the objective variables. The term $E_c \in \mathbb{R}^{n_c \times n_e}$ represents the influence of the exogenous state variables on the objective variables. For the exogenous state variables, $A_e \in \mathbb{R}^{n_e \times n_e}$ is the state matrix that defines their dynamics. Lastly, $\mathbf{w}(t) \in \mathbb{R}^{n_e}$ represents the external disturbances that impact the exogenous state variables.

The output equation considering the effect of exogenous state variables is:

$$\mathbf{y}(t) = C_c\mathbf{x}_c(t) + C_e\mathbf{x}_e(t) + D\mathbf{u}(t) \tag{5}$$

Where, $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ represents the output vector, which contains the objective variables. The matrix $C_c \in \mathbb{R}^{n_y \times n_c}$ maps the objective variables to the output, while $C_e \in \mathbb{R}^{n_y \times n_e}$ maps the exogenous state variables to the output. Additionally, $D \in \mathbb{R}^{n_y \times n_u}$ is the feedforward matrix that describes the direct influence of the control inputs on the output. Finally, the exogenous state variables are those components of the state vector that influence the objective variables but are not influenced by the control inputs. They capture the effects of external disturbances and are crucial for accurately modeling and controlling complex systems.

Two learned non-linear and sequence-to-sequence models ($f$) for the prediction of time series data $\mathcal{D}$ which output the state of the system at time $t+1$ are defined as follows:

- $f_A$ **(Exogenous State Variables in the Model's Prediction):**

$$\hat{\mathbf{s}}(t+1) = \hat{\mathbf{x}}_c(t+1) + \hat{\mathbf{x}}_e(t+1) = f_A(\hat{\mathbf{X}}_c, \hat{\mathbf{X}}_e, \mathbf{U}(t)) \tag{6}$$

- $f_B$ **(No Exogenous State Variables in the Model's Prediction):**

$$\hat{\mathbf{s}}(t+1) = \hat{\mathbf{x}}_c(t+1) = f_B(\hat{\mathbf{X}}_c, \mathbf{X}_e(t), \mathbf{U}(t)) \tag{7}$$

Where $\hat{\mathbf{s}}(t+1) \in \mathbb{R}^{n_{out}}$ is the prediction of the model with $n_{out}$ being the number of the variables in the prediction which is $n_c + n_e$ for the model $f_A$ and $n_c$ for the model $f_B$. Moreover, $\hat{\mathbf{x}}_c(t+1) \in \mathbb{R}^{n_c}$ and $\hat{\mathbf{x}}_e(t+1) \in \mathbb{R}^{n_e}$ are the prediction of the model for state and exogenous state variables. The terms $\mathbf{X}_c(t) \in \mathbb{R}^{l \times n_c}$, $\mathbf{X}_e(t) \in \mathbb{R}^{l \times n_e}$, and $\mathbf{U}(t) \in \mathbb{R}^{l \times n_u}$ define the historical record of the system's state, exogenous, and control variables at time $t$. When using these models as simulators over $h \in \mathbb{Z}^+$ horizon, if the prediction error at each time step is defined using the Euclidean distance, expressed as $e_t = \|\hat{\mathbf{s}}_t - \mathbf{s}_t\|$, then the system's state returned by the models at the time step $t+h$ can be described as (Mohammadi et al., 2024c):

$$\hat{\mathbf{s}}_{t+h} = f(\ldots f(f(\mathbf{s}_t, a_t) + e_t, a_{t+1}) + e_{t+1} \ldots, a_{t+h}) + e_{t+h} \tag{8}$$

The accuracy of each model as a simulator depends on the prediction errors from the beginning of the simulation until the end because the error at each step is accumulated and results in lower accuracy as the simulation continues. This compounding error issue is addressed in (Mohammadi et al., 2024a), where an improvement method was implemented to reduce the error at each simulation step, minimizing the effect of compounding error. The prediction error at the time step $e_{t+k}$ with $k \in \mathbb{Z}^+$ and $0 < k \leq h$, for $\hat{f}_A$ is:

$$e^{\hat{f}_A}{}_{t+k} = e_{t+k,c} + e_{t+k,e} \tag{9}$$

While the prediction error for $\hat{f}_B$ is defined as:

$$e^{\hat{f}_B}{}_{t+k} = e_{t+k,c} \tag{10}$$

The trained models can be improved to minimize multi-step simulation errors by employing an iterative method that incorporates the model's predictions back into the input at each training step (Mohammadi et al., 2024a). This approach, inspired by the work of (Venkatraman et al., 2015), is specifically adapted for recursive multi-step forecasting. During training, the model forecasts across the designated prediction horizon for each input-output pair from the dataset. At each step, the model's output is fed back into the input for the subsequent step's prediction, continuing this process until the horizon is reached. The loss is calculated for each one-step prediction and used for learning, enabling the model to correct errors at each step before proceeding, thereby preventing error propagation throughout the forecasting or simulation horizon. Assuming $\hat{f}_A$ and $\hat{f}_B$ to be the improved versions of learned $f_A$ and $f_B$ based on (Mohammadi et al., 2024a), it can be concluded that:

**Theorem 1.** *If $e^{\hat{f}'_A}{}_{t+k}$ and $e^{\hat{f}'_B}{}_{t+k}$ are the prediction errors of the simulators based on $f'_A$ and $f'_B$ at time step $t+k$, then $\forall f'_A \in \{f_A, \hat{f}_A\}, \forall f'_B \in \{f_B, \hat{f}_B\}$ we have $e^{\hat{f}'_B}{}_{t+k} \leq e^{\hat{f}'_A}{}_{t+k}$.*

*Proof.* exogenous state variables are unaffected by the system's state and control variables, so their values as a part of the model's input at each simulation step can be sampled from the offline data $\mathcal{D}$. With the same state and control variables at each step, the input at time step $t+k$ for model $\hat{f}'_A$ is:

$$\mathbf{S}(t) = \hat{\mathbf{X}}_c + \hat{\mathbf{X}}_e(t) + \mathbf{U}(t) \tag{11}$$

While the input to the model $\hat{f}'_B$ is:

$$\mathbf{S}(t) = \hat{\mathbf{X}}_c + \mathbf{X}_e(t) + \mathbf{U}(t) \tag{12}$$

The error term $e_{t+k,e}$ for the exogenous state variables in the model $\hat{f}'_a$ is calculated as:

$$e^{\hat{f}'_A}{}_{t+k,e} = \sum_{m=1}^{n_e} \|\hat{\mathbf{x}}_{e\{t+1,m\}} - \mathbf{x}_{e\{t+1,m\}}\| \geq 0 \tag{13}$$

While for $\hat{f}'_B$, as the exogenous state variables are sampled from the real values in $\mathcal{D}$, the error term $e_{t+k,e}$ for them in the model is equal to zero:

$$e^{\hat{f}'_B}{}_{t+k,e} = \sum_{m=1}^{n_e} \|\mathbf{x}_{e\{t+1,m\}} - \mathbf{x}_{e\{t+1,m\}}\| = 0 \tag{14}$$

It can be concluded from equations 13 and 14 that the error term for exogenous state variables which will affect the model's prediction for the next step in the model $\hat{f}'_B$ will always be lower or equal to the error term in the model $\hat{f}'_A$:

$$e^{\hat{f}'_B}{}_{t+k} \leq e^{\hat{f}'_A}{}_{t+k} \tag{15}$$

□

## 2.3 The LSTM Model

The *LSTM* architecture as shown in Figure 1 was designed with multiple layers to capture complex patterns in wastewater treatment data. Specifically, the network comprises two *LSTM* layers, each consisting of 256 units, as described in (Mohammadi et al., 2024c). We employed the 'tanh' activation function in the *LSTM* layers to facilitate non-linear learning. The model also included a dropout rate of 0.15 to prevent overfitting. The input to the model at each step consisted of a history of time steps, including all of the system's state variables, while the output was a single-step prediction of the system's state. The training and validation procedure of the base *LSTM* model is explained in (Mohammadi et al., 2024c).

While the input provided to the *LSTM* model consists of a historical record of the system's state $\mathbf{X}_c(t) \in \mathbb{R}^{l \times n_c}$, exogenous state $\mathbf{X}_e(t) \in \mathbb{R}^{l \times n_e}$ and control variables $\mathbf{U}(t) \in \mathbb{R}^{l \times n_u}$, detailed as follows:

$$\mathbf{S}(t) = \mathbf{X}_c(t) + \mathbf{X}_e(t) + \mathbf{U}(t) = \begin{bmatrix} \mathbf{x}_c(t) \\ \mathbf{x}_c(t-1) \\ \vdots \\ \mathbf{x}_c(t-l) \end{bmatrix} + \begin{bmatrix} \mathbf{x}_e(t) \\ \mathbf{x}_e(t-1) \\ \vdots \\ \mathbf{x}_e(t-l) \end{bmatrix} + \begin{bmatrix} \mathbf{u}(t) \\ \mathbf{u}(t-1) \\ \vdots \\ \mathbf{u}(t-l) \end{bmatrix} \tag{16}$$

Where $\mathbf{S}(t) \in \mathbb{R}^{l \times n}$ is the input to the *LSTM* model at time $t$, with $n = n_c + n_e + n_u$ representing the number of the input variables. The output of the model at each time step $t$ will be as follows:

$$\hat{\mathbf{S}}(t+1) = \begin{bmatrix} \hat{\mathbf{x}}(t+1) \\ \hat{\mathbf{x}}(t+2) \\ \vdots \\ \hat{\mathbf{x}}(t+p) \end{bmatrix} \tag{17}$$

Where $p \in \mathbb{Z}^+$ represents the model's output sequence length, which in this study is set to 1, consequently leading to $\hat{\mathbf{S}}(t+1) = \hat{\mathbf{x}}(t+1)$. The model's output can be determined from equations 6 and 7,

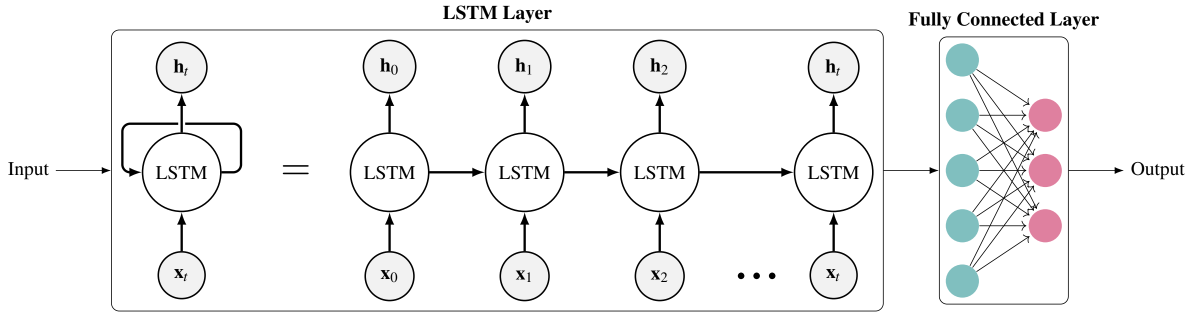


Figure 1: The structure of the ***LSTM*** (Hochreiter and Schmidhuber, 1997b) model for time series forecasting tasks, where $(\mathbf{x}_0, \ldots, \mathbf{x}_t)$, and $(\mathbf{h}_0, \ldots, \mathbf{h}_t)$ represent the input and the hidden state (output) of each LSTM cell (Mohammadi et al., 2024c).

depending on whether exogenous state variables are included in the prediction. The prediction error of *LSTM* model at time $t+1$ can be calculated as:

$$\mathcal{L}_{t+1} = \frac{1}{n_x} \sum_{d=1}^{n_x} \|\hat{\mathbf{x}}_{\{t+1,d\}} - \mathbf{x}_{\{t+1,d\}}\|^2 \quad (18)$$

Common training methods, known as *teacher-forcing* or supervised learning, utilize backpropagation and minimize the single-step loss function for each training batch. Each training batch $\mathbf{D}$ contains $z \in \mathbb{Z}^+$ input-output pairs sampled from the dataset $\mathcal{D}$, where $\mathbf{D}_i = (\mathbf{S}(t)_i, \mathbf{x}(t+1)_i)$. The optimization is performed over the model parameters $\theta$ as follows (Mohammadi et al., 2024a):

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{z} \sum_{d=1}^{n_x} \|(\hat{\mathbf{x}}_{t+1,d})_i - (\mathbf{x}_{t+1,d})_i\|^2 \quad (19)$$

where $\theta$ represents the parameters of the model, and $(\hat{\mathbf{x}}_{t+1,d})_i$ and $(\mathbf{x}_{t+1,d})_i$ are the predicted and true values of the $d$-th dimension of the system's state at time $t+1$ for the $i$-th pair in the batch $\mathbf{D}$.

### 2.4 Experiments Design

To investigate the experimental validation of Theorem 1, three distinct *LSTM* models were trained to evaluate the impact of incorporating additional variables. Table 2 details these models, highlighting their differences. While the models share the same structure and hyperparameters, they vary in including exogenous state variables, such as temperature, flow, maximum critical function value, and process phases, in the model output. The base model, trained with teacher-forcing, and improved versions with iterative training (E1, E2, E3, and E4), as described in (Mohammadi et al., 2024a), which employed a method similar to the "DATA AS DEMONSTRATOR" (DAD) approach. The *DILATE* loss function (Le Guen and Thome, 2019) was also used in the improvements, and the results were compared to those obtained using the Mean Squared Errors (*MSE*) loss function.

This study compares scenarios where only control and state variables are present in the model input ($f_0$), and exogenous state variables are included ($f_A$ and $f_B$). Furthermore, it compares two cases regarding the model output: in one case, the predicted values of the exogenous state variables are used at each simulation step ($f_A$), while in the other case, the actual values of the exogenous state variables are utilized in the simulation ($f_B$).

### 2.5 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 one *NVIDIA Tesla V100* GPU with 32 GB of RAM, all connected via *NVIDIA NVLink*.

## 3 RESULTS

This section presents the experimental results of incorporating exogenous state variables into the multi-step simulation of time series data for wastewater treatment.

### 3.1 Experimental Results

The *LSTM* model, as described in Section 2.3, was trained on various data combinations. To assess its effectiveness, we compared its performance with several state-of-the-art time series prediction models, as reported in (Mohammadi et al., 2024c). The experiments utilized three dataset variations ($f_0$, $f_A$, and $f_B$),

Table 2: The explanations of the trained models.

| Models | Name | Explanation |
|---|---|---|
| IOPPo-10i1o | $f_0$ | Without the exogenous state variables |
| IOPTQCfFiFoPo-15i6o | $f_A$ | Exogenous State Variables in the Model's Prediction |
| IOPTQCfFiFoPo-15i1o | $f_B$ | No Exogenous State Variables in the Model's Prediction |

each incorporating different combinations of exogenous state variables as detailed in Table 2. Moreover, Table 3 displays the average MSE and DTW metrics for the base model and its improved versions (E1, E2, E3, and E4) over one year. The best MSE and DTW values for each model are highlighted in bold, while the best values for each version and dataset variation are underlined.

After analyzing the various model versions and calculating the average MSE and DTW over a year-long daily simulation, the versions with the lowest DTW were selected for further comparison. The hyperparameters of the best-improved versions for each model are listed in Table 4. Additionally, Figure 2 visually represents the daily DTW values over the year, illustrating the error changes. Each point in Figure 2 represents the average loss of a simulation sequence that begins with the data at the start of the day and recursively predicts the output until the end of the day. The initial input to the model for each simulation point consists of a system history, incorporating all input variables over a specific lookback window. For this study, the lookback window was set to 240 steps for all models. The blue color in Figure 2 indicates the number of time steps that include actual values from the dataset used as model input at each step. Once the simulation reaches the point where the number of simulation steps equals the lookback window, it completely runs out of actual time steps. Beyond this point, all inputs to the model are predictions generated by the model itself.

Finally, the best versions of the models were used to simulate a period of 720 steps (24 hours) for different points of the year. The results of these simulations are shown in Figure 3, and the metrics for each point can be found in Table 5.

### 3.1.1 Improvement Performance

The average MSE and DTW in Table 3 are presented by columns to compare the performance of different versions of improved models across the entire dataset. The $f_0$ model, which does not include exogenous state variables, showed significantly higher MSE and DTW values than those that incorporated exogenous state variables. The best version of the $f_0$ model achieves an MSE of **0.4848** and a DTW of **1.7716**. Using the actual values of exogenous state variables at each step ($f_B$) results in lower error values, with the best version achieving an MSE of **0.2165** and a DTW of **1.1646**. This improvement represents a **64.49**% reduction in MSE and a **32.37**% reduction in DTW compared to the best $f_A$ version, and a **55.35**% reduction in MSE and a **34.27**% reduction in DTW compared to the best version of $f_0$ model. These results highlight the significant benefits of incorporating exogenous state variables and using their actual values to enhance the predictive accuracy of the *LSTM*-based simulator.

## 4 DISCUSSION

Incorporating exogenous variables like temperature, flow, maximum critical function value, and process phases into the *LSTM* model significantly enhanced the predictive accuracy and robustness of the wastewater treatment simulator, as evidenced by lower MSE and DTW values. These exogenous variables provided a more comprehensive representation of the system, capturing critical aspects that state variables alone couldn't, such as the impact of temperature and flow on phosphorus removal. This allowed the model to predict system behavior more accurately under varying conditions.

Including exogenous state variables in the model led to a significant reduction in MSE and DTW. The enhanced model, especially when using actual exogenous values at each simulation step, demonstrated robust performance, cutting MSE by **55**% and DTW by **34**% compared to the model without these variables. This reduction indicates that the model's ability to capture complex dynamics of the WWT process improved, resulting in more precise simulations. The $f_A$ model, which included exogenous state variables in the output, consistently showed improved performance across different months and seasons. This highlights the model's ability to predict these additional factors, enhancing its accuracy. Moreover, the $f_B$ model, which incorporated exogenous state variables in the input but not in the output and used their actual values at each simulation step, outperformed the $f_A$ model in all versions. This underscores the importance of exogenous state variables in representing the system's state and their independence from con-

Table 3: The average Mean Squared Error and Dynamic Time Warping data for the base model and improved versions during different months of the year. The best MSE and DTW values for each model and experiment are highlighted in bold and underlined, respectively.

| Models | Base Model | | E1 | | E2 | | E3 | | E4 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | DTW | MSE | DTW | MSE | DTW | MSE | DTW | MSE | DTW |
| $f_0$ | 449.4486 | 56.1453 | 0.5684 | 1.8893 | **0.4848** | 1.8369 | 0.5027 | **1.7716** | 0.5096 | 1.9028 |
| $f_A$ | 28.6279 | 11.1544 | 0.6536 | 1.7978 | 0.6981 | 1.8361 | **0.6098** | **1.7219** | 0.8137 | 2.0372 |
| $f_B$ | 13.1450 | 9.8183 | **0.2165** | **1.1646** | 0.2436 | 1.2566 | 0.3496 | 1.5075 | 0.2788 | 1.3412 |
| Average | 163.7405 | 25.7060 | 0.4795 | **1.6172** | **0.4755** | 1.6432 | 0.4874 | 1.6670 | 0.5340 | 1.7604 |

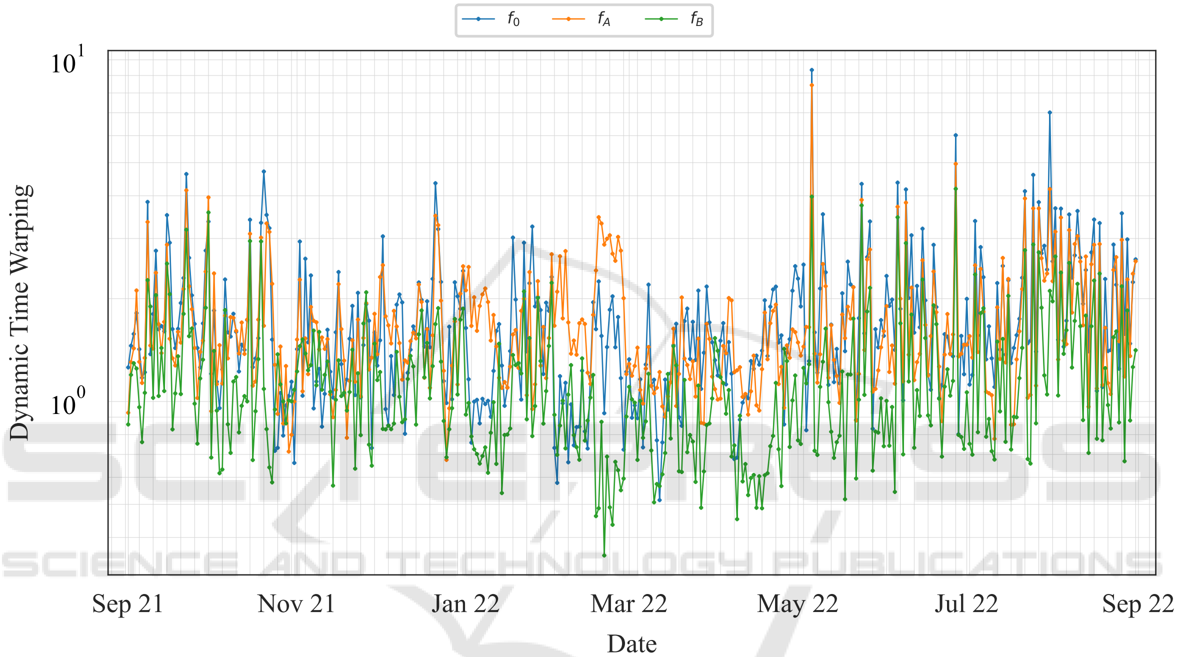


Figure 2: The loss for the best experiments of all dataset types with or without the exogenous state variables.

Table 4: The parameters of the best improved checkpoints for each experiment. Ex.: Experiment number, Ep.: Improvement epochs, Min EL and Max EL: Minimum and Maximum episode length during the improvement, Loss F.: The improvement loss function, and Alpha: Alpha in DILATE loss function.

| Models | Parameters | | | | | |
|---|---|---|---|---|---|---|
| | Ex. | Ep. | Min EL | Max EL | Loss F. | Alpha |
| $f_0$ | E3 | 80 | 240 | 240 | Dilate | 0.6 |
| $f_A$ | E3 | 80 | 240 | 240 | Dilate | 0.6 |
| $f_B$ | E1 | 80 | 240 | 240 | MSE | - |

trol and objective variable changes. Using independent exogenous state variables allows the model to leverage real-time actual values from the dataset, resulting in better simulation and accurately capturing the system's dynamics.

Using actual values of exogenous state variables at each simulation step (model $f_B$) mitigated the accumulation of prediction errors over multiple steps, reducing MSE by **55**% and DTW by **34**%. This approach kept the model's input closely aligned with reality, enhancing overall simulation accuracy. By correcting itself at each step, the model prevented errors from propagating throughout the simulation.

Table 5: The average Mean Squared Error and Dynamic Time Warping data for each model in the different points of the year. The best MSE and DTW values for each point and model are highlighted in bold and underlined, respectively.

| Points | $f_0$ | | $f_A$ | | $f_B$ | |
|---|---|---|---|---|---|---|
| | MSE | DTW | MSE | DTW | MSE | DTW |
| September | 0.9570 | 21.0640 | 1.3240 | 17.2590 | **0.8260** | **15.1950** |
| December | 0.6600 | 13.3860 | 0.7580 | 13.1930 | **0.2960** | **9.4360** |
| March | 0.5800 | 7.2210 | 0.5510 | 6.2090 | **0.1770** | **5.6840** |
| June | 0.7510 | 19.2260 | 1.0440 | 15.5430 | **0.4770** | **13.6430** |
| Average | 0.7370 | 14.8400 | 0.9190 | 13.0510 | **0.4440** | **11.3740** |

Performance improvement was consistent throughout one year in the collected dataset, with

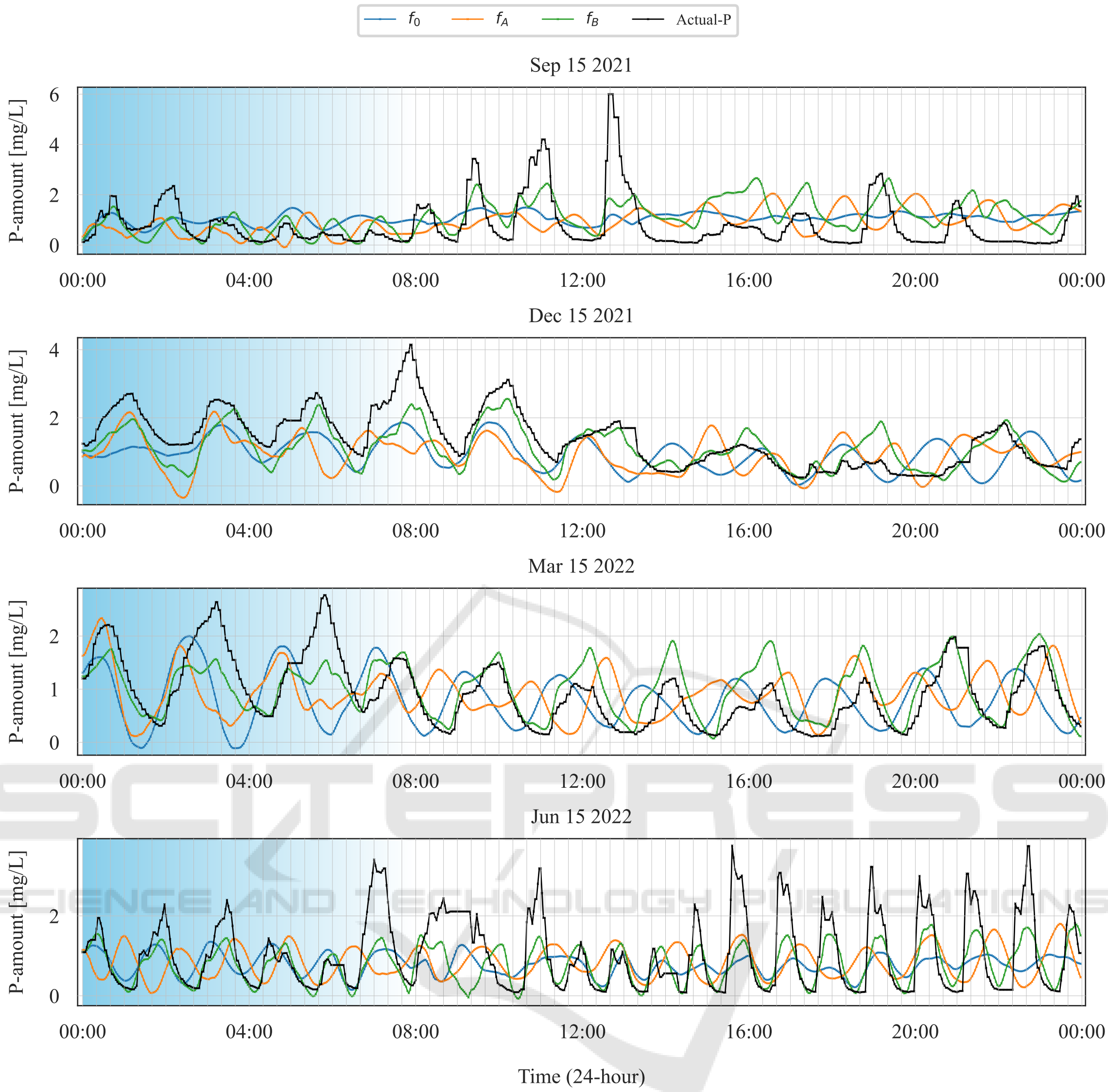


Figure 3: The simulation for the best experiments of each model. The number of past actual values in the input shown by the blue color decreases as the simulation reaches a point where all the input history is the predictions.

some variations observed. Notably, the model's performance improved in March, when the lowest MSE and DTW values were recorded. This suggests that the model can adapt to seasonal changes in the wastewater treatment process. Seasonal variations can affect inflow characteristics and process dynamics. The model's ability to maintain accuracy across these variations is a testament to its effectiveness.

Finally, the iterative improvement method reported in (Mohammadi et al., 2024a) enhanced the accuracy of the models, bringing them closer to the system's dynamics. Including exogenous state variables at each step assists the model in reducing prediction errors, which are often compounded from previous steps.

While including exogenous variables has clear benefits, it also presents challenges. These variables may not always be accurately measured or available in real-time, potentially impacting model performance. Additionally, the increased computational complexity and need for large datasets could limit this approach's applicability in smaller systems with limited data. Despite these challenges, the method significantly improves wastewater treatment simulations.

# 5 CONCLUSIONS

The study successfully demonstrated the advantages of incorporating exogenous state variables into an LSTM-based simulator for wastewater treatment. The improved model, particularly the $f_B$, significantly enhanced prediction accuracy and robustness. The key conclusions are:

- **Enhanced Accuracy:** Including exogenous state variables markedly improved the model's accuracy, as evidenced by lower MSE and DTW values across the year.
- **Error Mitigation:** Using actual values of exogenous state variables at each simulation step reduced MSE by **55**% and DTW by **34**%, effectively mitigating the compounding of prediction errors and leading to more reliable simulations.
- **Broad Applicability:** The model demonstrated robust performance across different seasonal conditions, highlighting its potential applicability in diverse operational settings.
- **Future Work:** Future research could explore integrating more external factors and applying similar methods to other aspects of wastewater treatment. Additionally, attention-based models and GPU training, as suggested by (Mohammadi et al., 2024c), could enhance efficiency and reduce computational time.

The improved LSTM-based simulator represents a significant advancement in wastewater treatment modeling. It offers a powerful tool for optimizing control strategies and enhancing operational efficiency.

# ACKNOWLEDGEMENTS

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. Disclaimer: This publication reflects only the author's view; the Research Executive Agency of the European Union is not responsible for any use that may be made of this information.

# REFERENCES

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

Gao, P., Yang, X., Zhang, R., Guo, P., Goulermas, J. Y., and Huang, K. (2023). Egpde-net: Building continuous neural networks for time series prediction with exogenous variables.

Goodwin, G. C., Graebe, S. F., Salgado, M. E., et al. (2001). *Control system design*, volume 240. Prentice Hall Upper Saddle River.

Gujer, W., Henze, M., Mino, T., Matsuo, T., Wentzel, M. C., and Marais, G. (1995). The activated sludge model no. 2: biological phosphorus removal. *Water science and technology*, 31(2):1–11.

Hansen, L. D., Stentoft, P. A., Ortiz-Arroyo, D., and Durdevic, P. (2024). Exploring data quality and seasonal variations of n2o in wastewater treatment: a modeling perspective. *Water Practice & Technology*, 19(3):1016–1031.

Hansen, L. D., Stokholm-Bjerregaard, M., and Durdevic, P. (2022). Modeling phosphorous dynamics in a wastewater treatment process using bayesian optimized lstm. *Computers & Chemical Engineering*, 160:107738.

Hespanha, J. P. (2018). *Linear systems theory*. Princeton university press.

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

Hochreiter, S. and Schmidhuber, J. (1997b). Long short-term memory. *Neural Computation*, 9:1735–1780.

Krüger A/S (2023). Hubgrade performance plant. https://www.kruger.dk/english/hubgrade-advanced-online-control. Accessed: 2023-11-30.

Le Guen, V. and Thome, N. (2019). Shape and time distortion loss for training deep time series forecasting models. *Advances in neural information processing systems*, 32.

Mohammadi, E., Ortiz-Arroyo, D., Stokholm-Bjerregaard, M., Hansen, A. A., and Durdevic, P. (2024a). Improved long short-term memory-based wastewater treatment simulators for deep reinforcement learning. *arXiv preprint arXiv:2403.15091*.

Mohammadi, E., Rani, A., Stokholm-Bjerregaard, M., Ortiz-Arroyo, D., and Durdevic, P. (2024b). Wastewater treatment plant data for nutrient removal system.

Mohammadi, E., Stokholm-Bjerregaard, M., Hansen, A. A., Nielsen, P. H., Ortiz-Arroyo, D., and Durdevic, P. (2024c). Deep learning based simulators for the phosphorus removal process control in wastewater treatment via deep reinforcement learning algorithms. *Engineering Applications of Artificial Intelligence*, 133:107992.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Venkatraman, A., Hebert, M., and Bagnell, J. (2015). Improving multi-step prediction of learned time series models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.