Context-driven Deep Learning Forecasting for Wastewater Treatment Plants

MD NAZMUL KABIR SIKDER*, Virginia Tech, USA
FERAS A. BATARSEH, Virginia Tech, USA

Wastewater-treatment utilities face various operational challenges that could benefit from *embodied AI* and other advanced cyber-physical technologies. These challenges include optimizing pump schedules, managing energy and chemical consumption during extreme weather events, and interpreting sensor data for water-quality treatment. Addressing these issues requires accurate short-term, multi-step forecasting tools to provide reliable real-time decision support, particularly during heavy-rainfall events that can overwhelm operations. Leading water-system operators and vendors in the United States report that tools capable of forecasting 4–6 hours ahead can significantly enhance resource management, including energy, chemicals, and manpower. However, accurate short-term forecasting is particularly difficult because of the non-linearities and seasonal variations inherent in plant data, which limit effective decision-making. To address these challenges, we propose $\mathbf{cP_2O}$, a context-driven forecasting solution, a novel hybrid deep-learning architecture integrating dynamic context extraction with hierarchical, dilated long-short-term-memory (LSTM) cells. The proposed model utilizes internal water-system data, such as flow rates and tunnel levels, along with exogenous variables including weather, river flow, and demographic information to derive relevant context. It captures both short-term fluctuations and long-term dependencies in water-level data, while an internal attention mechanism dynamically weighs the importance of exogenous information. We validate the model on two full-scale utilities: tunnel-water-level forecasting at DC Water's Blue Plains facility and nitrate-level prediction at AlexRenew. Relative to strong baselines, cP_2O reduces mean absolute percentage error by 22 % and 19 %, respectively, and its 90 % prediction bands cover 90.5 % ± 3.2 % of observations (5.9 % below, 3.6 % above). By dynamically incorporating contextual information, especially under critical conditions, the model delivers reliable real-time forecasts that enhance resource allocation and strengthen the overall resilience of wastewater-treatment operations.

CCS Concepts: • **Computing methodologies** → **Cyber-physical systems**; *Machine learning*.

Additional Key Words and Phrases: Context, Hybrid Deep Learning, Wastewater Treatment, Short-term Forecasting, Dilated Recurrent Neural Networks, Attention

1 INTRODUCTION

The management of Wastewater Treatment Plants (WWTPs) is becoming increasingly complex due to factors such as urban population growth, climate change, and aging infrastructure [27]. Operators face numerous challenges, including optimizing pump schedules, controlling energy and chemical consumption during extreme weather conditions, and accurately interpreting sensor data for water quality treatment [41]. Events such as heavy rainfall can strain these systems beyond their capacity, leading to overflows of untreated water that pose

*Corresponding author

Authors' Contact Information: Md Nazmul Kabir Sikder, nazmulkabir@vt.edu, Virginia Tech, Bradley Department of Electrical and Computer Engineering, Arlington, VA, USA; Feras A. Batarseh, batarseh@vt.edu, Virginia Tech, Department of Biological Systems Engineering, Arlington, VA, USA.

ACM 2378-9638/2025/6-ART
https://doi.org/10.1145/3744350

significant environmental and public health risks, potentially violating the standards and regulations of the US Environmental Protection Agency (EPA) [5].

Municipalities are increasingly turning to sensor technology and Artificial Intelligence (AI) for operational improvements, inspection, and data analysis [12]. AI is revolutionizing wastewater management by addressing operational challenges, minimizing risks, and contributing to environmental sustainability [43].

One key application of AI in this sector is predictive maintenance. AI algorithms monitor equipment conditions to predict maintenance needs, reducing downtime and emergency repairs [35, 40, 43]. Another significant application involves optimizing treatment processes. AI analyzes sensor data to optimize chemical dosing, energy consumption, and overall system efficiency, leading to cost savings and a reduced environmental footprint [2, 55, 63]. In addition, AI enables real-time monitoring and alerts by continuously monitoring wastewater quality, detecting anomalies, and providing early warnings to prevent contamination and ensure compliance with environmental regulations [22, 46, 48]. Furthermore, water pipeline inspections are enhanced through AI-powered drones and sensors, which detect problems such as cracks, corrosion, or blockages more efficiently and accurately than traditional methods [1, 54, 61]. Despite these significant advances in AI applications, accurately forecasting short-term fluctuations in complex WWTPs remains a considerable challenge [35].

1.1 Motivation

There is an increasing demand for accurate short-term forecasting tools in WWTPs to support real-time decision making and improve emergency preparedness [23]. Industry experts and major utilities in the United States have noted that predictive models with a forecasting horizon of 4 to 6 hours could substantially improve WWTP resource management and operational efficiency [35]. Despite this demand, conventional statistical methods and many recent Machine Learning (ML) and Deep Learning (DL) solutions struggle to effectively capture the complex nonlinear behaviors and seasonal patterns inherent in WWTP data [35, 57, 72].

Traditional ML models, such as Exponential Smoothing (ES) [26], Auto-Regressive Integrated Moving Average (ARIMA) [3], and XGBoost [14], often fail to capture temporal relationships in multivariate time series data due to a lack of mechanisms, such as recurrence, to model dependencies between sequential data points [18, 26]. Furthermore, these models typically require extensive preprocessing steps, such as decomposition or deseasonalization, which add complexity to the forecasting process [34]. Traditional ML models also face limitations in capturing long-term and seasonal dependencies, given their restricted receptive fields [36]. Furthermore, separate feature selection procedures are often necessary, resulting in inefficient training processes [35]. A notable limitation of many existing models is their focus on point predictions, which restricts their ability to assess predictive uncertainty [71].

Compounding these challenges is the limited incorporation of external contextual factors, which further restricts the practical effectiveness of these models [7, 9, 45, 47, 60, 64, 66, 68]. Few models explicitly address seasonality, and existing solutions frequently lack mechanisms to manage forecast bias, compromising reliability [50]. In typical WWTP operations, external factors—weather conditions, river flows, demographic shifts, and economic activities—exert significant influence yet remain unmonitored by internal systems. For instance, real-time weather data can predict inflow surges from heavy rainfall, while demographic trends provide insights into monthly or weekly water usage patterns.

In this study, we integrate these external variables into a forecasting model to enhance the predictive accuracy of critical WWTP variables, such as wastewater tunnel levels (L_T) and nitrate concentrations (L_{NO_3}). By incorporating external context—detailed further in Appendix A—our approach allows for more accurate short-term forecasting, supports proactive operational decisions, improves system resilience, and reduces operational costs. This integration bridges the gap left by traditional models and leverages comprehensive contextual information to optimize WWTP performance under varying operational and environmental conditions.

Moreover, our proposed solution aligns with the principles of Embodied Artificial Intelligence (Embodied AI), as it enables AI-driven forecasting grounded in real-time sensory input and control over physical infrastructure. The model interacts with real-world wastewater systems via multi-modal sensor data (e.g., pump activity, tunnel water levels, rainfall), and informs time-sensitive operational decisions that directly impact physical processes. Additionally, by dynamically incorporating contextual factors such as weather patterns and urban infrastructure behavior, the model demonstrates the core embodied AI trait of perception-action coupling in such complex cyber-bio-physical environments.

1.2 Contributions

In this paper, we introduce Contextualized Predictive Process Optimization (cP_2O), a context-driven forecasting solution using DL tailored for WWTPs. Leveraging a novel hybrid DL architecture, cP_2O accurately predicts key WWTP variables by integrating a dynamic context extraction stage with hierarchically dilated [13] Long Short-Term Memory (LSTM) cells. This integration enables the model to capture both short-term fluctuations and long-term dependencies influenced by exogenous variables. Additionally, an internal attention mechanism dynamically allocates weights to contextual information alongside utility data, enhancing the model's sensitivity to important input features and addressing missing context within utility data.

The primary contributions of this work are as follows:

(1) **Hybrid Architecture with Context Integration:** We develop a hybrid model that combines dynamic context extraction with dilated LSTMs and an attention mechanism. This architecture processes raw WWTP time series data without extensive preprocessing, effectively capturing short-term and long-term temporal dependencies.
(2) **Dynamic Context Extraction:** Our model introduces a context extraction stage that incorporates exogenous variables—such as weather data, river flows, and demographic trends—to generate dynamic context vectors. These vectors enhance predictive capabilities by integrating external influences significantly impacting WWTP operations.
(3) **Attention Mechanism for Feature Weighting:** We implement an internal attention mechanism that dynamically assigns weights to input features based on relevance. This reduces the reliance on manual feature selection, allowing the model to focus on the most impactful variables and thereby enhancing forecasting accuracy.
(4) **Multi-Step Ahead Forecasting with Uncertainty Estimation:** cP_2O provides both point forecasts and predictive intervals for multiple time steps ahead (4 to 6 hours). This dual output equips decision-makers with insights into forecast uncertainty, crucial for effective risk assessment and real-time operational planning.
(5) **Bias Reduction through Quantile Loss Function:** We employ a quantile loss function to mitigate forecast bias, particularly during peak or extreme events. This approach ensures more balanced predictions by reducing the influence of outliers—such as abrupt water level surges—on model performance.

We validate the effectiveness of cP_2O through two key experiments conducted on real-world data:

(1) **Tunnel Wastewater Level Forecasting:** This experiment focuses on forecasting influent water levels in tunnels and reservoirs at the Blue Plains Advanced WWTP, operated by DC Water[1]. Accurate short-term forecasting is essential for managing the facility's extensive infrastructure, optimizing pump operations, and preventing system overloads. By providing 4 to 6-hour ahead predictions for tunnel wastewater levels, cP_2O facilitates improved pump scheduling, reducing energy consumption and mitigating the risk of untreated water overflow into the environment [35].

[1] https://dcwater.com/

(2) **Chemical Variable Prediction:** This experiment targets the prediction of critical chemical variables—specifically, pH, ammonia (NH_4), and nitrate (NO_3) levels—at AlexRenew[2]. Effective monitoring of these variables is vital for nutrient removal and compliance with water quality standards essential for ecosystem protection. Using cP_2O, we develop a predictive model for chemical sensor values, offering a cost-effective and reliable solution for continuous monitoring that enhances process control and supports regulatory compliance [63].

In both experiments, cP_2O outperforms traditional models, reducing Mean Absolute Percentage Error (MAPE) by up to 22% compared to existing models and achieving lower Root Mean Squared Error (RMSE). These results demonstrate the potential of cP_2O to enhance resource allocation, drive energy savings, and bolster the resilience of WWTP operations, particularly under extreme weather conditions.

The remainder of this paper is structured as follows: Section 2 reviews related work on short-term time series forecasting and DL approaches. Section 3 provides research questions on the theoretical and practical aspects of the proposed context modeling approach for WWTPs. Section 4 provides an overview of the cP_2O model architecture. Sections 5 and 6 detail the experimental setup and present the results of the tunnel wastewater level forecasting and chemical variable prediction experiments. Section 7 discusses the findings and concludes with implications and directions for future research.

2 RELATED FORECASTING STUDIES

Short-term forecasting is essential for optimizing operations in WWTPs, as it directly impacts operational efficiency, resource allocation, and system resilience. However, it remains a critical challenge due to factors such as non-stationarity, high dimensionality, and complex seasonality inherent in time series data [35].

2.1 Traditional Forecasting Methods

Traditional statistical methods such as ARIMA [3], ES [26], and Kalman Filtering [28] have long been used for time series forecasting. While these methods are effective for linear data patterns, they often struggle with non-linearities, complex seasonality, and integrating exogenous variables such as weather conditions or market trends [27]. Furthermore, they are limited in capturing long-term dependencies, vital for accurate forecasting in modern applications.

As a response to these limitations, ML and DL techniques have gained prominence due to their flexibility in handling complex patterns in high-dimensional time series data. Techniques such as Support Vector Machines (SVMs) [39], Neural Networks (NNs) [19], and Recurrent Neural Networks (RNNs) [16] have shown promising results. More advanced approaches, such as Convolutional Neural Networks (CNNs) [37] and LSTM networks [30], have been widely applied for time series forecasting [57]. These models have improved the ability to handle complex, nonlinear dynamics and high-dimensional time series data. However, they often require extensive preprocessing and domain knowledge to select relevant features and handle raw time series data effectively [72]. Furthermore, the absence of mechanisms to dynamically weigh input feature importance limits their capacity to emphasize the most impactful variables at each time step, which may lead to suboptimal forecasting performance [25].

Hybrid approaches address these limits by combining algorithms and exploiting the strengths of each. For instance, Kim et al. [32] proposed a hybrid Recurrent Inception CNN model for capturing short- and long-term dependencies, significantly improving forecast accuracy. Similarly, [42] employed a hybrid ensemble approach combining LightGBM, XGBoost, and NNs, achieving more accurate and robust forecasts than single models. Despite these advancements, hybrid models encounter challenges that limit their effectiveness in real-world applications, especially in systems heavily influenced by external factors. A primary limitation lies in the

[2] https://alexrenew.com/

insufficient integration of external contextual variables, such as weather patterns, river flow information, or demographic trends [47]. While hybrid models have advanced the ability to capture temporal dependencies, they often fall short in effectively leveraging exogenous variables, which can reduce accuracy in domains such as WWTPs where external influences are critical [35]. Moreover, hybrid models frequently struggle to adapt to changing external conditions due to a lack of dynamic context extraction mechanisms [60]. Their reliance on manual feature selection and the absence of mechanisms to dynamically weigh input features further constrain their performance [25].

2.2 Context-Aware Forecasting Methods

Context-aware forecasting models have gained attention to address the limitations of traditional and hybrid methods in integrating external influences. In the context of WWTPs, external factors such as weather conditions, river levels, demographic changes, economic trends, and environmental variables constitute crucial contextual information [47]. Recent studies have highlighted the value of context-aware models in enhancing forecasting accuracy and robustness. For instance, Solomon et al. [60] highlighted the role of external data, such as weather and economic indicators, in improving model performance when dealing with systems influenced by multiple factors. Similarly, [63] demonstrated that incorporating real-time weather data into models enhances forecasting accuracy, especially in scenarios where sudden changes in external conditions impact the system's behavior.

Context-enhanced hybrid models can capture both short- and long-term effects, improving forecast reliability [25]. For instance, [66], the absence of dynamic context extraction mechanisms limits the models' adaptability to shifting external conditions and their capacity to capture complex relationships between exogenous variables and the target time series. Additionally, a lack of mechanisms to dynamically weigh input features based on real-time relevance further restricts the efficacy of these models [68].

Several recent studies further support the importance of contextual variables in environmental forecasting models. For instance, Farmanifard et al. [21] demonstrated that integrating environmental and historical contextual information—such as meteorological variables—substantially improved cyclone trajectory predictions in a hybrid context-aware deep learning model. In wastewater applications, Cheng et al. [15] showed that influent flow, temperature, and biochemical oxygen demand (BOD) are among the most impactful variables for forecasting WWTP behavior. Similarly, Yu et al. [74] emphasized that spatiotemporal context, including location-specific environmental signals and seasonal cycles, plays a critical role in improving forecast accuracy for Earth system models. These findings align with our approach and support the inclusion of diverse contextual information to improve forecasting in WWTPs.

The review of current literature highlights several critical gaps in forecasting models for WWTPs:

- Traditional ML and DL models often struggle to integrate external factors, such as weather data, river conditions, and demographic trends, limiting their applicability in WWTPs where external influences are crucial [35, 47].
- Many state-of-the-art models require extensive preprocessing and domain-specific knowledge to select relevant features and manage raw time series data effectively [72].
- Most existing models do not dynamically weigh input features based on their relevance, resulting in suboptimal performance in systems heavily influenced by changing external variables [25].
- A majority of time series forecasting models lack uncertainty estimation capabilities, which are essential for risk assessment and decision-making in critical infrastructure [51].
- Many forecasting models struggle to mitigate forecast bias during peak events or extreme conditions, reducing their reliability and deployability in real-world applications [32].

By addressing these critical gaps—such as integrating external contextual variables, reducing the need for extensive preprocessing, dynamically weighing input features, estimating uncertainty, and mitigating forecast bias during extreme events—this work aims to significantly improve the accuracy and resilience of forecasting models in WWTPs.

3 RESEARCH QUESTIONS

This section introduces the research questions addressed in this paper, emphasizing both the theoretical and practical aspects of the proposed context modeling approach for WWTPs. Mathematical formulations are presented using vector notations to describe the model behavior succinctly.

(1) **RQ1: Does incorporating contextual data into forecasting models significantly improve the accuracy of short-term predictions in WWTPs compared to models that do not use context?** Given a WWTP dataset $\mathbf{D} \in \mathbb{R}^{T\times N}$, representing multiple time series of WWTP variables (e.g., inflow, water levels), a context matrix $\mathbf{C} \in \mathbb{R}^{T\times M}$, containing exogenous variables (e.g., weather data, river flow), and the true output at time $t+1$ as $\mathbf{y}_{t+1}$, we consider two models: a Context-Driven Model $\hat{\mathbf{y}}_{t+1}^{\text{context}} = f_{\text{context}}(\mathbf{D}_t, \mathbf{C}_t; \boldsymbol{\theta}_{\text{context}})$ and a Without Context Model $\hat{\mathbf{y}}_{t+1}^{\text{no-context}} = f_{\text{no-context}}(\mathbf{D}_t; \boldsymbol{\theta}_{\text{no-context}})$. The prediction error $\mathcal{E}$ is defined as $\mathcal{E}(\hat{\mathbf{y}}, \mathbf{y}) = \mathbb{E}[L(\hat{\mathbf{y}}, \mathbf{y})]$, where $L(\cdot,\cdot)$ is a loss function. We hypothesize that incorporating contextual data improves the model's accuracy: $\mathcal{E}(\hat{\mathbf{y}}_{t+1}^{\text{context}}, \mathbf{y}_{t+1}) < \mathcal{E}(\hat{\mathbf{y}}_{t+1}^{\text{no-context}}, \mathbf{y}_{t+1})$.

(2) **RQ2: Can the cP$_2$O model adapt to different forecasting tasks across various WWTPs, maintaining high accuracy for both tunnel water level forecasting and nitrate level prediction?** To assess this, we apply the same modeling approach to different WWTPs. For WWTP A, with dataset $\mathbf{D}^{(A)}$ and context $\mathbf{C}^{(A)}$, we train the model with parameters $\boldsymbol{\theta}^{(A)}$, leading to predictions $\hat{\mathbf{y}}_{t+1}^{(A)} = f(\mathbf{D}_t^{(A)}, \mathbf{C}_t^{(A)}; \boldsymbol{\theta}^{(A)})$. Similarly, for WWTP B, with dataset $\mathbf{D}^{(B)}$ and context $\mathbf{C}^{(B)}$, we obtain $\hat{\mathbf{y}}_{t+1}^{(B)} = f(\mathbf{D}_t^{(B)}, \mathbf{C}_t^{(B)}; \boldsymbol{\theta}^{(B)})$. We hypothesize that the model achieves comparable performance metrics across WWTPs A and B for different tasks, demonstrating flexibility and generalizability: $\mathcal{E}(\hat{\mathbf{y}}_{t+1}^{(A)}, \mathbf{y}_{t+1}^{(A)}) \approx \mathcal{E}(\hat{\mathbf{y}}_{t+1}^{(B)}, \mathbf{y}_{t+1}^{(B)})$.

(3) **RQ3: Does the integration of an attention mechanism in cP$_2$O enhance the model's ability to weigh input features, thereby improving forecasting accuracy dynamically?** To evaluate the impact of the attention mechanism, we compare a Model with Attention $\hat{\mathbf{y}}_{\text{att},\,t+1} = f_{\text{att}}(\mathbf{D}_t, \mathbf{C}_t; \boldsymbol{\theta}_{\text{att}})$ and a Model without Attention $\hat{\mathbf{y}}_{\text{no-att},\,t+1} = f_{\text{no-att}}(\mathbf{D}_t, \mathbf{C}_t; \boldsymbol{\theta}_{\text{no-att}})$. We hypothesize that the attention mechanism improves the model's accuracy: $\mathcal{E}(\hat{\mathbf{y}}_{\text{att},\,t+1}, \mathbf{y}_{t+1}) < \mathcal{E}(\hat{\mathbf{y}}_{\text{no-att},\,t+1}, \mathbf{y}_{t+1})$.

(4) **RQ4: Can the quantile loss function employed in cP$_2$O effectively reduce forecast bias during peak events or extreme conditions, and improve uncertainty estimation in multi-step ahead forecasting?** We compare a Model with Quantile Loss Function, using predictions $\hat{\mathbf{y}}_{\text{quantile},\,t+1}$ and parameters $\boldsymbol{\theta}_{\text{quantile}}$, with a Model with Standard Loss Function, using predictions $\hat{\mathbf{y}}_{\text{standard},\,t+1}$ and parameters $\boldsymbol{\theta}_{\text{standard}}$. Forecast bias $\mathcal{B}$ is defined as $\mathcal{B} = \mathbb{E}[\hat{\mathbf{y}}_{t+1} - \mathbf{y}_{t+1}]$. We hypothesize that the quantile loss function reduces forecast bias during extreme conditions: $\mathcal{B}_{\text{quantile}} < \mathcal{B}_{\text{standard}}$. Additionally, the model improves uncertainty estimation, which is measured by metrics such as the prediction interval.

By addressing these research questions, we aim to validate the effectiveness of cP$_2$O in improving forecasting accuracy through context integration. We assess its scalability across different WWTPs, and demonstrate the contributions of the attention mechanism and quantile loss function in enhancing model performance.

4 METHOD: SHORT-TERM FORECASTING FOR WWTPS

In this section, we present the materials and methods for the proposed short-term forecasting model for WWTPs. Short-term forecasting in WWTPs aims to predict key variables—such as inflow or water levels—over a short horizon, typically within 4–6 hours. The cP$_2$O model performs forecasts based on past observations and external influencing factors. Specifically, we aim to forecast several hours into the future by predicting the sequence $\{\mathbf{y}_{t+1}, \mathbf{y}_{t+2}, \ldots, \mathbf{y}_{t+H}\}$, using the historical observations $\{\mathbf{y}_{t-M+1}, \mathbf{y}_{t-M+2}, \ldots, \mathbf{y}_t\}$ as input. Here, $\mathbf{y}_t$ represents the system's state vector at time t, M denotes the length of the historical time series used as input, and H specifies the number of time steps in the forecast horizon.

4.1 Context Extraction and Forecasting Stages

As depicted in Figure 1, the proposed forecasting model comprises two interconnected stages: the **context extraction stage** and the **forecasting stage**. The context extraction stage processes historical data from external sources—such as weather variables (rainfall, temperature), river data, and demographic or economic indicators—that provide additional context for the forecasting task. Additionally, we select a representative subset of WWTP data to incorporate context that captures historical patterns.

However, concatenating all context variables into a single high-dimensional input vector becomes computationally impractical. To address this challenge, our context model processes each context variable individually. Multiple context variables are processed in parallel within a batch structure for computational efficiency. At each time step, we flatten the outputs from the batch and generate a single context vector $\mathbf{r}_t$. An optional modulation can yield a general context vector $\mathbf{r}'_t$ for each time step; however, performance may decrease for high-dimensional datasets. The exogenous variables undergo preprocessing steps using a dynamic smoothing component, which includes normalization and deseasonalization.

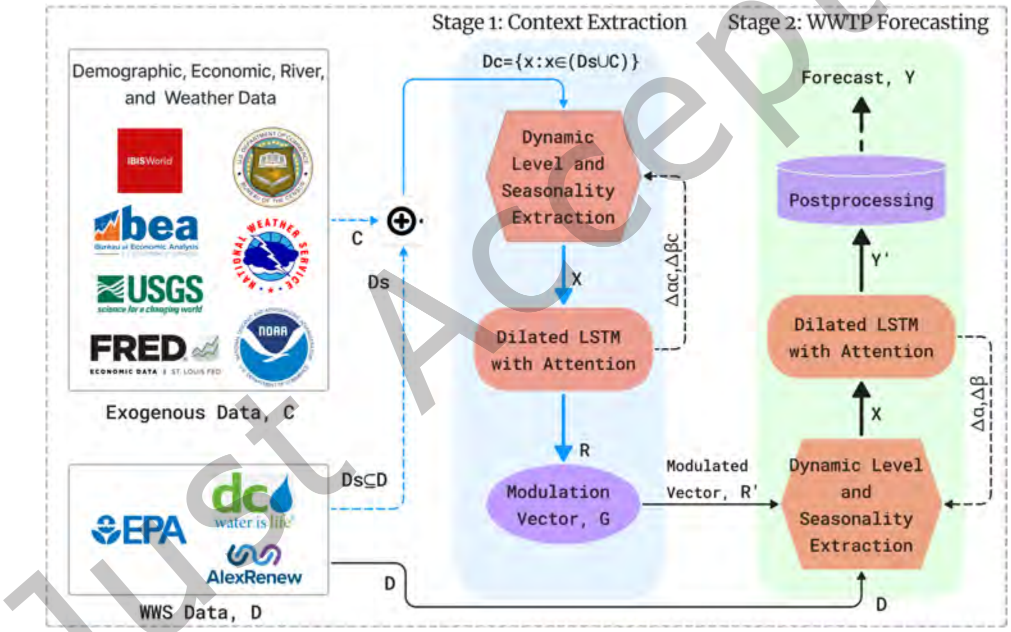

Stage 1: Context Extraction
Stage 2: WWTP Forecasting
Demographic, Economic, River, and Weather Data
IBISWorld
bea
USGS
FRED
NOAA
Exogenous Data, C
Dc={x:x∈(Ds∪C)}
C
Ds
Dynamic Level and Seasonality Extraction
X
Δαc,Δβc
Dilated LSTM with Attention
R
Modulation Vector, G
Modulated Vector, R'
Forecast, Y
Postprocessing
Y'
Dilated LSTM with Attention
X
Δα,Δβ
Dynamic Level and Seasonality Extraction
Ds⊆D
EPA
dc water is life
AlexRenew
WWS Data, D
D
D

Fig. 1. The diagram illustrates a two-stage forecasting framework that integrates exogenous data with WWTP data to improve forecasting accuracy. In Stage 1 (Context Extraction), relevant contextual information is extracted from external sources such as weather variables (e.g., rainfall, temperature), river data, and demographic or economic indicators. This stage includes preprocessing steps such as normalization and deseasonalization, using a dynamic smoothing component to generate a context vector ($\mathbf{R}'$) for each time step. In Stage 2 (Forecasting), this context vector is combined with internal WWTP sensor data—such as pump activity and inflow levels—after undergoing similar preprocessing. The integrated data is input into a hierarchically dilated LSTM architecture with attention for multi-step forecasting and uncertainty estimation. Postprocessing is then applied to produce final point forecasts and predictive intervals, offering both accurate predictions and uncertainty measures.

The context extraction stage and the forecasting stage are synchronized in their time steps, ensuring that contextual information aligns with the internal data. The forecasting stage processes the WWTP's internal sensor data, such as pump activity and inflow levels. Similar to the context stage, the internal data undergoes dynamic smoothing for preprocessing. Before feeding the data into the dilated LSTM cells, the input is augmented by concatenating it with the context vector $\mathbf{r}_t$ from the context extraction stage. This integration enhances the forecasting model's understanding of external contextual information, enhancing its predictive capabilities.

When the number of exogenous series is relatively limited, we assign a weighting vector $\mathbf{h}$ to each series. This vector has the same length as the context vector $\mathbf{r}_t$ and is initially set to ones. The purpose of $\mathbf{h}$ is to adjust the general context information, customizing it to the specific needs of each series. Importantly, $\mathbf{h}$ remains constant and does not change across different time steps.

To reduce forecasting errors, the model parameters are optimized collectively, foregoing the application of a separate loss function for the context stage output. Instead, the entire model is trained end-to-end. Given computational resource constraints and batch size limitations, a batching mechanism is employed to handle multiple exogenous variables simultaneously. However, practical considerations often require limiting the number of exogenous variables. A solution is to apply WWTP domain knowledge to identify a representative subset of context series from the WWTP data. This can involve generating linear combinations of a few key base series and integrating relevant exogenous variables.

The final outcomes of the model are point forecasts and predictive intervals, providing accurate predictions along with uncertainty estimations essential for risk assessment and decision-making in WWTP operations. By leveraging external factors alongside internal WWTP data and by efficiently integrating context information, the proposed model achieves greater forecasting accuracy compared to models that rely solely on internal data.

4.2 cP$_2$O Architecture

The architecture of the proposed forecasting model, referred to as cP$_2$O, integrates data preprocessing and post processing, dynamic pattern extraction, and dilated LSTM cells with an attention mechanism. Each time series variable—both from the wastewater treatment plant (WWTP) and exogenous data sources—is decomposed by the dynamic smoothing component into its level (M_t) and seasonal (N_t) components.

As illustrated in Figure 1, time series data—including context variables C and utility data D—are processed to extract level, seasonality, and other patterns dynamically. This processing enables the forecasting stage to focus on more stable and normalized inputs. The prepared data undergoes normalization and the addition of calendar features (e.g., day of the week) to enhance forecasting accuracy. The context vectors $\mathbf{r}_t$, computed in the context extraction stage, are concatenated with the plant data before being fed into the forecasting stage. The LSTM network, equipped with dilated cells and an internal attention mechanism, processes the combined data to capture temporal dependencies and leverage the learned context from the first stage. In the post-processing step, the normalized forecast values are converted back to their original scale by applying inverse normalization and reintroducing the previously extracted patterns. The proposed model produces three key outputs:

1. Point forecasts of the target variable for multiple time steps ahead (e.g., 4–6 hours).
2. Prediction intervals (lower and upper bounds) for uncertainty estimation.
3. Adjustments to the smoothing parameters ($\Delta\gamma_t$ and $\Delta\delta_t$) to capture changes in level and seasonality over time.

4.2.1 *Preprocessing and Input Pattern Generation.*

The preprocessing stage of our forecasting model serves two main purposes. First, it transforms the time series data into a format that is compatible with LSTM networks. Second, it generates input and output patterns that are compiled into training datasets for the model's training process. Within both the context extraction and forecasting phases of the cP$_2$O architecture, we utilize dynamic smoothing based on the Holt-Winters method with multiplicative seasonality [33]. The essential equations for updating the level and seasonal components are:

$$M_t = \gamma_t \frac{X_t}{N_{t-P}} + (1 - \gamma_t) M_{t-1} \tag{1}$$

$$N_t = \delta_t \frac{X_t}{M_t} + (1 - \delta_t) N_{t-P} \tag{2}$$

Here, M_t denotes the level component, N_t represents the seasonal component, X_t is the observed value at time t, P corresponds to the seasonal period (e.g., daily, weekly), and $\gamma_t, \delta_t \in [0, 1]$ are smoothing coefficients. These coefficients are dynamically modified by the LSTM network, which learns adjustments ($\Delta\gamma_t$ and $\Delta\delta_t$) during training to accommodate changes in the time series.

The preprocessing module reformats the time series for compatibility with the dilated LSTM. The main input for both context extraction and forecasting stages is the sequence immediately before the forecasted period. Let Ω_t^{in}, spanning 24 hours, represent the input window for the t-th sequence, and let Ω_t^{out}, spanning 4 hours, represent the output window. These windows are advanced by 4 hours to create subsequent input and output sequences. The input sequence undergoes deseasonalization, normalization, and a logarithmic transformation to mitigate the influence of outliers during the learning process. Vector $\mathbf{x}_t = [x_\tau]_{\tau \in \Omega_t^{\text{in}}} \in \mathbb{R}^{24}$ represents the input sequence after preprocessing. Deseasonalization removes weekly seasonal patterns, while normalization using the mean μ_t removes long-term trends within the input window. This scaling ensures that all preprocessed series are on a comparable scale, promoting effective cross-learning across multiple series.

To enrich the input data for the forecasting phase, we augment the input patterns with additional features, including the series' level and seasonality, calendar information, and the context vector from the context stage. The enhanced input vector is defined as in Equation (3):

$$\mathbf{x}_t' = \left[\mathbf{x}_t,\ \tilde{\mathbf{n}}_t,\ \mathbf{r}_t, \log_{10}(\mu_t),\ \mathbf{c}_t^y,\ \mathbf{c}_t^m,\ \mathbf{c}_t^w\ \right] \tag{3}$$

The vector $\tilde{\mathbf{n}}_t \in \mathbb{R}^4$ represents four seasonal components specifically tailored to the forecast horizon, predicted using the Holt-Winters model (Equations 1 and 2) for $\tau \in \Omega_t^{\text{out}}$. The term $\log_{10}(\mu_t)$ captures the local level of the WWTP data, providing a logarithmic scale representation of the mean value within the input window. Additionally, $\mathbf{c}_t^y$, $\mathbf{c}_t^m$, and $\mathbf{c}_t^w$ are one-hot encoded vectors that indicate the week of the year, the day of the month, and the day of the week, respectively, offering temporal context. Finally, $\mathbf{r}_t$ denotes the context vector obtained from the context extraction stage, enriching the forecasting model with external contextual information.

For the context extraction stage, an enriched input pattern follows a similar structure to $\mathbf{x}_t'$, but excludes the context vector $\mathbf{r}_t$. The output pattern aligns with the target sequence defined by Ω_t^{out}. This pattern is derived by normalizing the original sequence, as in Equation (4), to ensure consistency across different series for error calculation by the loss function.

$$\mathbf{y}_\tau = \frac{X_\tau}{\mu_t}, \quad \text{where } \tau \in \Omega_t^{\text{out}} \tag{4}$$

Here, X_τ represents the original time series value at time τ. We train the model using patterns generated by shifting the input and output windows by 4 hours, which creates sequences for training. The LSTM in the main stage generates a vector predicting the next 4-hour forecasts, as specified in Equation (5):

$$\hat{\mathbf{y}}_t^{\text{LSTM}} = [\hat{\mathbf{y}}_\tau^{\text{LSTM}}]_{\tau \in \Omega_t^{\text{out}}} \in \mathbb{R}^4 \tag{5}$$

These predicted values are reverted to the original scale during post-processing, according to Equation (6):

$$\hat{X}_\tau = \exp(\hat{y}_\tau^{\text{LSTM}})\, \mu_t\, \tilde{\mathbf{n}}_{t,\tau}, \quad \text{where } \tau \in \Omega_t^{\text{out}} \tag{6}$$

The loss function utilizes the normalized predictions to maintain consistency, as shown in Equation (7):

$$\hat{\mathbf{y}}_\tau = \frac{\hat{X}_\tau}{\mu_t} \tag{7}$$

Furthermore, the LSTM predicts two vectors representing the lower and upper bounds of the prediction intervals: $\underline{\hat{\mathbf{y}}}_t^{\text{LSTM}}$ and $\hat{\overline{\mathbf{y}}}_t^{\text{LSTM}}$. These are transformed into actual values using the same post-processing steps as the point forecasts.

The optimization algorithm uses the discrepancies between the predicted output patterns and the actual output patterns to adjust all model parameters, including those of the exponential smoothing components, the LSTM, and the adjustments $\Delta\gamma_t$ and $\Delta\delta_t$. Importantly, the context stage generates context vectors $\mathbf{r}_t$, which do not have target values; however, its parameters are updated in conjunction with those of the main stage to minimize the overall forecasting error.

4.2.2 Dilated LSTM with Attention Mechanism. In this subsection, we present the customized dilated LSTM cells designed to identify contextual events and seasonal patterns in time series data, including exogenous and utility data (see Figure 2b). These cells, inspired by the concepts in [13] and [59], are equipped with an internal attention mechanism to weigh input features dynamically. Each dilated LSTM cell maintains two hidden states ($\mathbf{h}$-states) and two cell states ($\mathbf{c}$-states), all vectors in $\mathbb{R}^h$, where h is the dimension of the hidden state. Specifically:

(1) Recent states: $\mathbf{c}_{t-1}^i$ and $\mathbf{h}_{t-1}^i$, which store information from the immediate past time step $t-1$, similar to a standard LSTM cell [30].
(2) Delayed states: $\mathbf{c}_{t-d}^i$ and $\mathbf{h}_{t-d}^i$, which hold information from an earlier time step $t-d$ with $d > 1$. The dilation factor $d \in \mathbb{N}$ represents the number of time steps of delay and is crucial for capturing dependencies at different time scales. Incorporating delayed states effectively expands the receptive field and enhances the cell's ability to model long-term and seasonal patterns.

Gating Mechanisms: Inspired by both the LSTM and GRU architectures [17, 30], our cell employs two distinct gating mechanisms to manage the cell state $\mathbf{c}_t^i$: the Update Gate ($\mathbf{u}_t^i$) and the Forget Gate ($\mathbf{f}_t^i$). The Update Gate determines the extent to which the candidate cell state $\tilde{\mathbf{c}}_t^i$ contributes to the new cell state, effectively incorporating new information. Meanwhile, the Forget Gate controls the influence of the recent cell state $\mathbf{c}_{t-1}^i$, determining how much of the past information should be retained. Any remaining influence is assigned to the delayed cell state $\mathbf{c}_{t-d}^i$, weighted by $\mathbf{1} - \mathbf{u}_t^i - \mathbf{f}_t^i$. This design ensures a balanced integration of recent and historical information, enhancing the cell's memory capabilities and its ability to capture both short-term and long-term dependencies.
Cell Operations The cell's operations at each time step t for layer i are defined as follows:

$$\mathbf{f}_t^i = \sigma\left(\mathbf{W}_f^i \mathbf{x}_t^i + \mathbf{V}_f^i \mathbf{h}_{t-1}^i + \mathbf{U}_f^i \mathbf{h}_{t-d}^i + \mathbf{b}_f^i\right) \tag{8}$$

$$\mathbf{u}_t^i = \sigma\left(\mathbf{W}_u^i \mathbf{x}_t^i + \mathbf{V}_u^i \mathbf{h}_{t-1}^i + \mathbf{U}_u^i \mathbf{h}_{t-d}^i + \mathbf{b}_u^i\right) \tag{9}$$

$$\mathbf{o}_t^i = \sigma\left(\mathbf{W}_o^i \mathbf{x}_t^i + \mathbf{V}_o^i \mathbf{h}_{t-1}^i + \mathbf{U}_o^i \mathbf{h}_{t-d}^i + \mathbf{b}_o^i\right) \tag{10}$$

$$\tilde{\mathbf{c}}_t^i = \tanh\left(\mathbf{W}_c^i \mathbf{x}_t^i + \mathbf{V}_c^i \mathbf{h}_{t-1}^i + \mathbf{U}_c^i \mathbf{h}_{t-d}^i + \mathbf{b}_c^i\right) \tag{11}$$

The input vector at time t for layer i is represented by $\mathbf{x}_t^i \in \mathbb{R}^n$. The hidden state vectors, $\mathbf{h}_{t-1}^i$ and $\mathbf{h}_{t-d}^i$, capture information from the recent and delayed time steps, respectively, and both reside in $\mathbb{R}^h$. The weight matrices $\mathbf{W}_*^i \in \mathbb{R}^{h \times n}$, $\mathbf{V}_*^i$, and $\mathbf{U}_*^i \in \mathbb{R}^{h \times h}$ (where $*$ corresponds to the gates f, u, o, or the cell state c) determine the transformations applied to the inputs and hidden states. Bias terms for each gate are denoted by $\mathbf{b}_*^i \in \mathbb{R}^h$.

The sigmoid function $\sigma(\cdot)$ and hyperbolic tangent function $\tanh(\cdot)$ serve as the activation functions, applied element-wise to introduce non-linearity. Element-wise multiplication is denoted by $\otimes$, facilitating the interaction between various vectors during the gating and state-update processes.

Cell State Update: The cell state $\mathbf{c}_t^i \in \mathbb{R}^h$ is updated by combining the candidate cell state $\tilde{\mathbf{c}}_t^i$, the recent cell state $\mathbf{c}_{t-1}^i$, and the delayed cell state $\mathbf{c}_{t-d}^i$, weighted by the gating vectors:

$$\mathbf{c}_t^i = \mathbf{u}_t^i \otimes \tilde{\mathbf{c}}_t^i + \mathbf{f}_t^i \otimes \mathbf{c}_{t-1}^i + \left(\mathbf{1} - \mathbf{u}_t^i - \mathbf{f}_t^i\right) \otimes \mathbf{c}_{t-d}^i \tag{12}$$

Here, $\mathbf{1} \in \mathbb{R}^h$: Vector of ones and $\mathbf{u}_t^i, \mathbf{f}_t^i \in \mathbb{R}^h$: Gate vectors with elements in $[0, 1]$.

Constraints on Gates: To ensure proper weighting and stability:

$$0 \leq \mathbf{u}_{t,k}^i, \mathbf{f}_{t,k}^i \leq 1, \quad \mathbf{u}_{t,k}^i + \mathbf{f}_{t,k}^i \leq 1, \quad \forall k \in \{1, 2, \ldots, h\}$$

This ensures that the weights assigned to $\tilde{\mathbf{c}}_t^i$, $\mathbf{c}_{t-1}^i$, and $\mathbf{c}_{t-d}^i$ sum to at most 1 element-wise, and the remaining weight $(1 - \mathbf{u}_{t,k}^i - \mathbf{f}_{t,k}^i)$ is non-negative.

Hidden State Computation: The hidden state $\mathbf{h}_t^i$ is computed as:

$$\mathbf{h}_t^i = \mathbf{o}_t^i \otimes \tanh(\mathbf{c}_t^i) \tag{13}$$

Where $\mathbf{o}_t^i \in \mathbb{R}^h$ is the output gate vector controlling the exposure of the cell state.

Attention Mechanism Integration: Our model incorporates an internal attention mechanism to weigh input features dynamically. The input vectors for the two layers are defined as:

$$\mathbf{x}_t^1 = \mathbf{x}_t \tag{14}$$

$$\mathbf{x}_t^2 = \mathbf{x}_t \otimes \exp(\mathbf{m}_t) \tag{15}$$

Here, $\mathbf{x}_t \in \mathbb{R}^n$: Original input vector at time t and $\mathbf{m}_t \in \mathbb{R}^n$: Attention vector derived from the hidden state of the first layer.

Derivation of the Attention Vector: The attention vector $\mathbf{m}_t$ is obtained by partitioning the hidden state $\mathbf{h}_t^1$ of the first layer:

$$\mathbf{h}_t^1 = [\mathbf{h}_{t,\text{recurrent}}^1; \mathbf{m}_t] \tag{16}$$

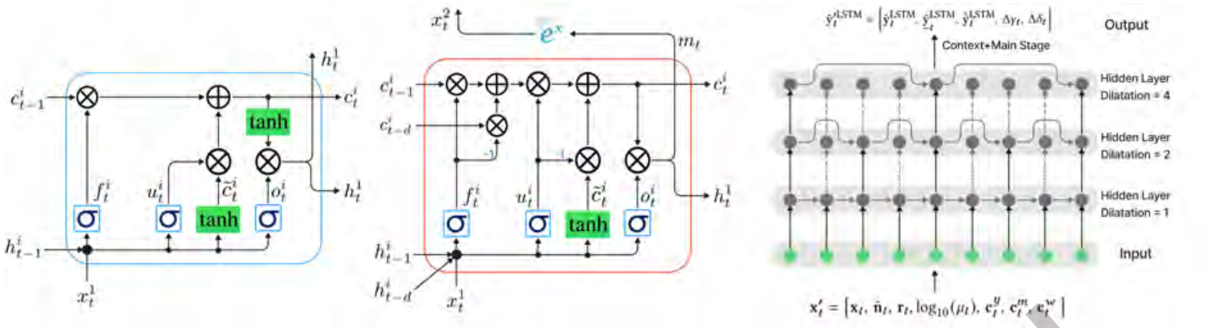
Here, $\mathbf{h}_{t,\text{recurrent}}^1 \in \mathbb{R}^{s_h}$: Portion of the hidden state used for recurrent processing, and $\mathbf{m}_t \in \mathbb{R}^n$: Attention vector used to modulate the input for the next layer.

After applying an exponential function to ensure positive weights, the attention vector modulates the inputs to the second layer. This mechanism allows the model to focus on the most relevant features at each time step.

Overall Network Architecture: Figure 2c depicts the overall architecture of the LSTM network, comprising three layers with dilation factors of 1, 2, and 4, respectively. Stacking layers with hierarchical dilations, the model captures features across multiple time scales, enhancing its ability to model seasonal and long-term patterns. To prevent vanishing gradients when adding more layers, we integrate ResNet-style shortcut connections between layers [29]. Additionally, the input vector $\mathbf{x}_t$ is supplied to all layers, enhancing the learning of complex patterns.

Embedding Layer: We employ a linear embedding layer to transform binary calendar vectors ($\mathbf{c}_t^w$, $\mathbf{c}_t^m$, and $\mathbf{c}_t^y$) into continuous vectors of dimension d, reducing dimensionality and capturing temporal features effectively. This embedding is learned during the training process.

Output Layer: The final linear output layer produces the model's outputs, which serve multiple purposes in the main forecasting stage. These outputs include point forecasts, represented as $\hat{\mathbf{y}}_t^{\text{LSTM}} \in \mathbb{R}^H$, where H denotes the


c_{t-1}^i
c_t^i
h_t^1
tanh
f_t^i
u_t^i
$\tilde{c}_t^i$
o_t^i
σ
h_{t-1}^i
x_t^1
x_t^2
e^x
m_t
c_{t-d}^i
h_{t-d}^i
-1
$\hat{y}_t'^{\text{LSTM}} = [\hat{y}_t^{\text{LSTM}}, \underline{\hat{y}}_t^{\text{LSTM}}, \hat{\bar{y}}_t^{\text{LSTM}}, \Delta\gamma_t, \Delta\delta_t]$
Output
Context+Main Stage
Hidden Layer Dilatation = 4
Hidden Layer Dilatation = 2
Hidden Layer Dilatation = 1
Input
$\mathbf{x}_t' = [\mathbf{x}_t, \hat{\mathbf{n}}_t, \mathbf{r}_t, \log_{10}(\mu_t), \mathbf{c}_t^y, \mathbf{c}_t^m, \mathbf{c}_t^w]$

(a) Traditional LSTM Cell

(b) Dilated LSTM Cell with attention

(c) cP2O architecture (dashed line link is absent in the context stage)

Fig. 2. cP$_2$O architecture: (a) A traditional LSTM cell (b) Dilated LSTM cell with attention mechanism (c) Hierarchical dilated LSTM architecture (the dashed line link is absent in the context stage)

forecast horizon. Additionally, the layer provides lower and upper bounds for prediction intervals, denoted by $\underline{\hat{\mathbf{y}}}_t^{\text{LSTM}}$ and $\hat{\bar{\mathbf{y}}}_t^{\text{LSTM}}$, respectively, to estimate uncertainty. Lastly, the layer outputs adjustments to the smoothing coefficients, $\Delta\gamma_t$ and $\Delta\delta_t$, which help dynamically update the level and seasonal components of the time series. The complete output vector is:

$$\hat{\mathbf{y}}_t'^{\text{LSTM}} = \left[\hat{\mathbf{y}}_t^{\text{LSTM}}, \underline{\hat{\mathbf{y}}}_t^{\text{LSTM}}, \hat{\bar{\mathbf{y}}}_t^{\text{LSTM}}, \Delta\gamma_t, \Delta\delta_t\right] \tag{17}$$

In the context extraction stage, the output consists of: Context vector $\mathbf{r}_t^{(i)} \in \mathbb{R}^u$ for the i-th series and adjustments of $\Delta\gamma_t$ and $\Delta\delta_t$. For a context batch containing K series, the context vectors are concatenated as:

$$\mathbf{r}_t = \left[\mathbf{r}_t^{(1)}, \mathbf{r}_t^{(2)}, \ldots, \mathbf{r}_t^{(K)}\right] \in \mathbb{R}^{uK} \tag{18}$$

This combined context vector $\mathbf{r}_t$ is integrated into the input of the main forecasting stage, enriching it with contextual information.

4.3 Loss Function

Our model provides point forecasts for up to 4–6 hours ahead, along with the lower and upper bounds of the prediction intervals for each forecasted point. We apply the following loss function:

$$\mathcal{L}_\theta = \ell\left(y_\theta, \hat{y}_{p^*,\theta}\right) + \lambda\left[\ell\left(y_\theta, \hat{y}_{p_1,\theta}\right) + \ell\left(y_\theta, \hat{y}_{p_2,\theta}\right)\right] \tag{19}$$

where the pinball loss function $\ell(y, \hat{y}_p)$ is defined as:

$$\ell(y, \hat{y}_p) = \begin{cases} p(y - \hat{y}_p), & \text{if } y \geq \hat{y}_p \\ (p-1)(y - \hat{y}_p), & \text{if } y < \hat{y}_p \end{cases} \tag{20}$$

In these equations, y_θ represents the normalized observed value at time θ, while $\hat{y}_{p,\theta}$ denotes the normalized predicted value at time θ for the p-th quantile. The parameter $p^* = 0.5$ corresponds to the median forecast, which serves as the point forecast. The quantiles p_1 and p_2 specify the lower and upper bounds of the prediction interval, typically set to $p_1 = 0.05$ and $p_2 = 0.95$, providing a 90% confidence interval. Finally, $\lambda \geq 0$ is a weighting coefficient that controls the relative importance of the prediction interval components within the overall loss function, allowing for flexible optimization of both point forecasts and uncertainty estimates.

The normalized observed value y_θ is obtained by:

$$y_\theta = \frac{X_\theta}{\mu_t} \tag{21}$$

where X_θ is the original time series value at time θ, and μ_t is the mean of the input window Ω_t^{in} as defined in the preprocessing stage.

The normalized predicted values $\hat{y}_{p,\theta}$ are derived from the LSTM outputs and adjusted during post-processing:

$$\hat{X}_{p,\theta} = \exp\left(\hat{y}_{p,\theta}^{\text{LSTM}}\right)\ \mu_t\ \tilde{n}_{t,\theta} \tag{22}$$

$$\hat{y}_{p,\theta} = \frac{\hat{X}_{p,\theta}}{\mu_t} \tag{23}$$

Here, $\hat{y}_{p,\theta}^{\text{LSTM}}$ is the LSTM output for the p-th quantile at time θ, $\tilde{n}_{t,\theta}$ is the forecasted seasonal component from the exponential smoothing model, and $\hat{X}_{p,\theta}$ is the predicted value in the original scale before normalization.

By operating on normalized values, the loss function ensures that errors have a consistent impact on the learning process across multiple time series with varying scales and error magnitudes.

This loss function consists of three components. The first component, the point forecast loss $\ell\left(y_\theta,\ \hat{y}_{p^*,\theta}\right)$, represents the loss associated with the point forecast. When $p^* = 0.5$, the pinball loss becomes symmetric and is equivalent to the mean absolute error (MAE). The second component, the lower prediction interval loss $\ell\left(y_\theta,\ \hat{y}_{p_1,\theta}\right)$, corresponds to the loss associated with the lower bound of the prediction interval, encouraging the lower quantile predictions to be below the observed values with a probability of p_1. The third component, the upper prediction interval loss $\ell\left(y_\theta,\ \hat{y}_{p_2,\theta}\right)$, represents the loss associated with the upper bound of the prediction interval, ensuring that the upper quantile predictions are above the observed values with a probability of $1 - p_2$.

The pinball loss function is asymmetric, with the degree of asymmetry determined by the quantile levels p. This three-part structure allows for the simultaneous optimization of both point forecasts and prediction intervals, with the coefficient λ controlling the relative emphasis on each component. When $\lambda = 1$, all components are equally weighted; reducing λ places more focus on optimizing the point forecast.

Additionally, the pinball loss function helps mitigate forecast bias by assigning different penalties to positive and negative errors, ensuring asymmetric error handling. By adjusting p^* to values less than or greater than 0.5, we can reduce tendencies toward positive or negative biases [20]. This approach can also be applied to adjust biases in the prediction intervals.

The proposed cP$_2$O method embodies core aspects of Embodied AI. It dynamically integrates real-time sensory inputs into decision-making loops for WWTP physical operations. These sensory inputs include rainfall, river flows, tunnel levels, and demographic trends. Unlike standard intelligent control methods, cP$_2$O does not rely solely on static setpoints or internal sensor data. Instead, cP$_2$O continuously perceives and adapts to changing external environmental and operational contexts. This perception-action coupling allows the model to directly inform and control WWTP processes in real-time. Examples of controlled processes include pump activation and chemical dosing. Therefore, cP$_2$O explicitly operationalizes Embodied AI principles within complex cyber-physical systems.

5 EXPERIMENTAL DESIGN

In this section, we assess the performance of our proposed model on Short-Term Water Level Forecasting tasks for WWTP. We present the dataset, the training and optimization methodologies, the baseline models used for comparison, and our experimental results. The section concludes with an ablation study and a discussion of the findings.

5.1 WWTP Data

5.1.1 Blue Plains Advanced WWTP: DC Water. The Blue Plains Advanced WWTP at DC Water incorporates a sophisticated tunnel system designed to mitigate the overflow of stormwater and sewage during heavy rain events. Historically, the plant would reach its maximum capacity during such events, leading to untreated water being discharged directly into the river or causing widespread flooding in the city's sewer system. To address this issue, DC Water implemented an underground tunnel system that acts as a water retention mechanism. This tunnel captures excess stormwater and sewer overflows, temporarily storing them until the plant can process and treat the water post-rainfall.

The sewer system serving Washington, D.C., parts of Maryland, and Virginia connects to the tunnel system at multiple critical junctures. These connections are facilitated by Combined Sewer Overflow (CSO) structures [6], which divert excess water from the sewer network into the tunnel system when sewer levels exceed certain thresholds. This preemptive diversion prevents overflow within the city's sewer infrastructure. Additionally, rain gauges and flow meters are strategically placed across the system to monitor water levels and trigger the diversion process when needed.

At the endpoint of the tunnel, a micro-treatment facility begins the treatment of the overflow water before it enters the larger WWTP. The plant's pumping system is key to managing inflows, with small pumps handling routine inflows and large pumps activated during peak events when the tunnel reaches capacity. Interestingly, energy efficiency plays a significant role in operational decisions. Small pumps, although slower, are more energy-efficient over prolonged use, while large pumps consume substantially more energy but can de-water the tunnel much faster. This trade-off requires utility operators to balance operational efficiency with energy costs, particularly in scenarios where heavy inflow can be predicted in advance.

To monitor tunnel water levels, DC Water employs various level indicators at multiple points along the tunnel. These indicators stage water levels and are used to guide operational decisions regarding pump activation. Depending on the event, operators may switch between different level indicators to ensure the most accurate measurements are used. Predictive models are employed to forecast water levels, providing operators with a 4-hour window to prepare pump operations. This predictive capability allows for the efficient scheduling of either small or large pumps, optimizing energy use and ensuring that the tunnel does not overflow.

Figure 3 shows a few identified peaks in tunnel water levels at DC Water during critical events, along with corresponding sensor data such as rain gauges, pump activity, and flow sensors. Several peaks in water levels were not associated with substantial rain gauge readings, pump activity, or flow sensor data. These peaks suggest the presence of events or anomalies not directly captured by DC Water's internal sensors, such as external factors such as upstream river flow changes or unrecorded inflow sources. Analysis using the National Oceanic and Atmospheric Administration's (NOAA) [11] storm events database indicates that these anomalies align with external events not directly captured by the plant's internal sensors. The following are the matched events from NOAA:

(1) August 10, 2022 (See 1st-row plots of Figure 3): A flash flood event was triggered by a weak boundary overhead and anomalously high moisture levels, resulting in slow-moving thunderstorms. This caused significant water level rises in areas such as Rock Creek Parkway and Rhode Island Avenue NE. Although the plant's rain gauges recorded minimal rainfall, the flash flooding had a pronounced impact on water levels.
(2) November 2, 2022 (See 2nd-row plots of Figure 3): Despite the absence of notable rainfall or pump activity, a water level peak was observed. This anomaly is likely attributable to upstream river inflows or unmonitored urban runoff entering the sewer system, which the plant sensors failed to detect.
(3) December 15, 2022 (See 4th-row plots of Figure 3): A peak in water levels coincided with a heavy rainfall event that overwhelmed certain parts of the system. Although the plant sensors only partially captured

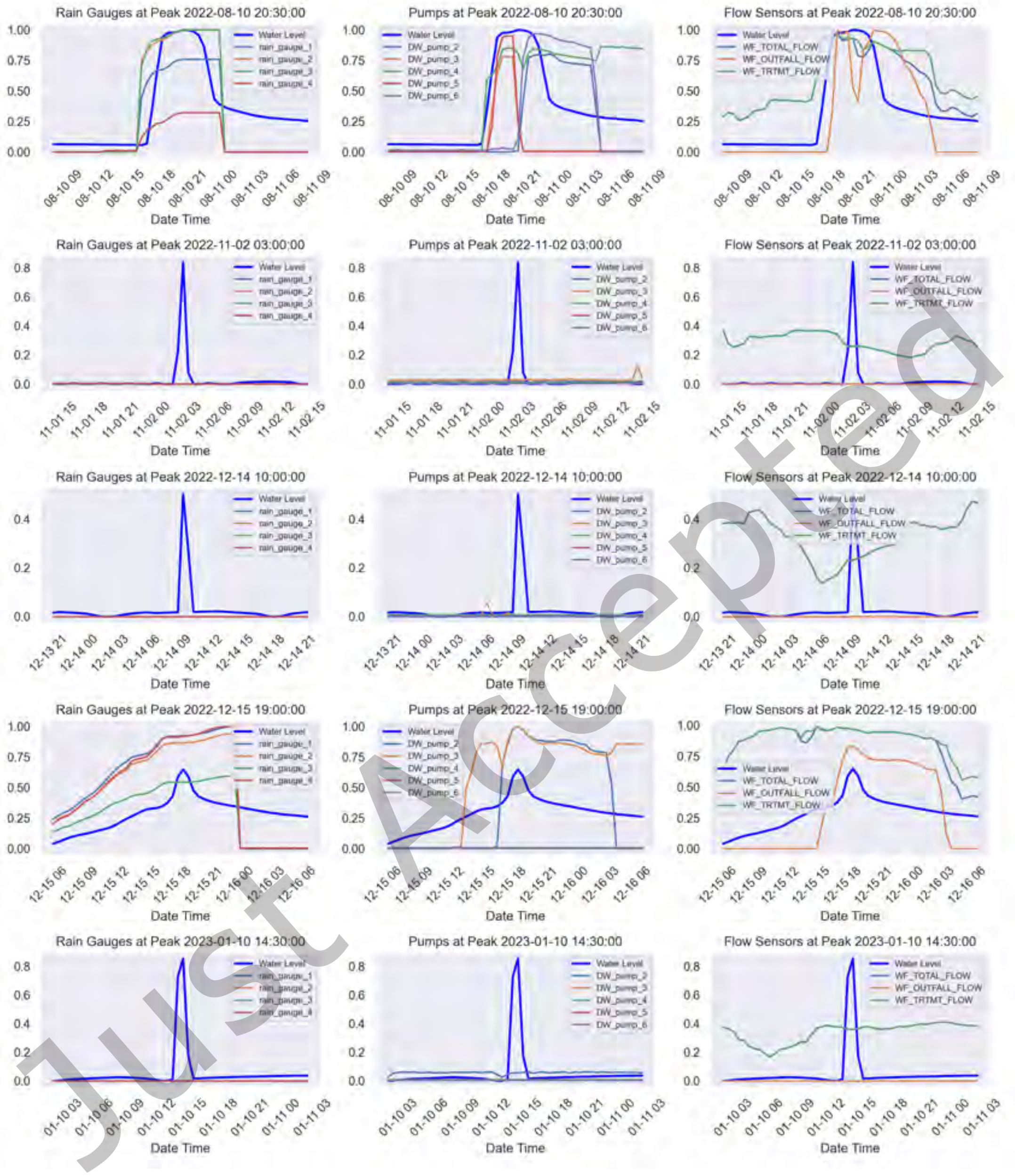


Fig. 3. DC Water time series plots of rain gauges, pump activities, and flow sensors at peak water level events. Each row represents a specific peak event, with the left column showing rain gauge data, the middle column displaying pump activity, and the right column illustrating flow sensor readings. The peaks are aligned to highlight their relationship with water levels and various operational parameters across different dates. The plotted data emphasizes the dynamic response of the system during high water level events, offering insights into how rainfall, pump activation, and flow sensor readings correspond to each peak.

Table 1. Grouped tag names and descriptions for DC Water tunnel system and AlexRenew chemicals data

No.	Variable group	Data Description	Variable Count
		DC Water Tunnel System Data	
1	DIV_FLOW_CSO_X	Diversion Flow to tunnel from various CSO locations	7
2	DIV_FLOW_MPMP	Diversion Flow to tunnel from Main Pump Station	2
3	DIV_FLOW_JBAB	Diversion Flow to tunnel from JBAB	1
4	DIV_FLOW_POP	Diversion Flow to tunnel from Poplar Point	1
5	DIV_FLOW_MST	Diversion Flow to tunnel from M Street	1
6	TUNNEL_LEVEL	Tunnel Level measurements at various locations	7
7	OF_RVR_STCT	Overflow to River Structure	2
8	OF_BRL_CSO	Overflow to River Barrels (A, B, C) from various CSO locations	9
9	TIDE_GATE_LKG	Tide Gate Leakage (River level)	2
10	PLANT_FLOW	Plant influent and complete treatment flow	2
11	PLANT_OUTFALL	Plant outfall flow	1
12	TDP_FLOW	Flow at TDP (TDP-2 to TDP-6)	5
13	RAIN_GAUGE_DAY	Rain Gauge measurements at various pump stations	4
		AlexRenew Chemicals Data	
14	INFLUENT_EFFLUENT_FLOW	Total influent and effluent flows (MGD)	3
15	SOLID_TSS	Average dewatering central TSS (mg/L)	1
16	DISSOLVED_OXYGEN	Dissolved Oxygen (average, minimum, maximum) (mg/L)	3
17	AMMONIA_NH3	Ammonia (average, minimum, maximum) (mg/L)	3
18	NITRATE_NO3	Nitrate (average, minimum, maximum) (mg/L)	3
19	pH	pH levels (minimum, maximum)	2
20	TEMPERATURE	Water, air, and central (average, minimum, maximum) (F)	5
21	REACTOR_DECANT_FLOW	Reactor decant flow (GPD)	1
22	WAS_FLOW	Waste activated sludge flow (GPD)	1
23	PROCESS_AIR	Process air to CPT (scfm)	1
24	CARBON	Carbon transferred to CPT (gal)	1
25	PRECIPITATION	Precipitation (inches)	1

the inflow, external factors such as rapid urban runoff likely played a significant role in the observed spike.

These events highlight the limitations of relying solely on internal plant sensors for forecasting. Incorporating contextual data—such as real-time weather data, upstream river flow rates, and external urban flood monitoring—can provide critical insights into such anomalies. By including this context, the cP$_2$O model can better identify whether these peaks are outliers or valid operational events driven by external conditions, enhancing decision-making and operational efficiency.

5.1.2 AlexRenew Chemicals Dataset. The AlexRenew Wastewater Treatment Plant (WWTP), located in Alexandria, focuses on treating wastewater for its service areas. This real-world dataset provides a comprehensive range of parameters for wastewater treatment processes, which are highly relevant for short-term WWTP forecasting. The dataset includes crucial water quality indicators and flow rates, such as total influent and effluent flow, dissolved oxygen (DO) levels, ammonia (NH_3) and nitrate (NO_3), pH levels, and temperatures across multiple reactors.

Table 1 summarizes the grouped tag names and descriptions for both the DC Water tunnel system and the AlexRenew chemicals data, providing an overview of the variables included in the datasets. In addition to these primary water quality attributes, external weather data has been incorporated into the AlexRenew dataset. This data, sourced from NOAA [11] and the National Weather Service (NWS) [69], includes parameters such as precipitation and atmospheric temperature. By merging these datasets, the study provides a more integrated view of how external conditions influence the water treatment process at AlexRenew.

The integration of the AlexRenew dataset with NOAA weather data provides a richer context for understanding the variability in wastewater treatment performance. By combining internal water quality parameters with external environmental factors, these datasets allow for better modeling of inflow levels during extreme weather events and provide deeper insights into how external conditions impact the treatment process.

The merged dataset offers a unique opportunity to apply predictive models for short-term wastewater inflow forecasting. The inclusion of weather-related data, such as rainfall and temperature, enhances the forecasting model's ability to capture the dynamic interactions between external conditions and WWTP performance. As a result, this holistic approach improves the accuracy of predictions, aiding in more efficient operational decisions for WWTPs.

5.2 Exogenous Contextual Data

Developing a comprehensive forecasting model for WWTPs necessitates the integration of multiple external data sources. These exogenous contextual variables provide valuable insights into factors that affect water quality and treatment processes. Below is a detailed breakdown of potential data sources and their relevant variables:

5.2.1 Weather Data.

- Sources: NOAA [11], NWS [69], Weather Underground [24].
- Variables: Precipitation, temperature, humidity, wind speed, atmospheric pressure.

Weather conditions significantly influence both the volume and characteristics of wastewater inflow. For example, precipitation events can lead to increased inflow due to runoff and infiltration, while temperature and humidity affect evaporation rates and biological activity within the WWTP.

5.2.2 River Data (Water Quality and Flow).

- Sources: United States Geological Survey (USGS) [52] and EPA [5]
- Variables: River flow rates, water temperature, pH levels, dissolved oxygen, chemical contaminants, biological oxygen demand (BOD), nutrient levels (nitrogen, phosphorus).

Understanding river data is crucial for assessing natural water quality and evaluating the impact of effluents from the treatment plant on river ecosystems. Variables such as flow rates and water quality parameters help model the interactions between wastewater discharge and the receiving water bodies.

5.2.3 Demographic Data.

- Sources: United States Census Bureau [53], IBIS World [58]
- Variables: Population density, household size, urbanization rate.

Demographic factors affect both the volume and composition of wastewater generated. Higher population densities and urbanization rates typically result in increased wastewater production, while household size can influence per capita water patterns in the United States.

5.2.4 Economic Data.

- Sources: Bureau of Economic Analysis (BEA) [10], Federal Reserve Economic Data (FRED) [44]
- Variables: Employment rates, industrial output, types of businesses.

Economic activity influences the types and quantities of industrial effluents entering the WWTP. Variables such as industrial output and business types help predict variations in wastewater characteristics due to industrial discharges.

5.3 Training, Optimization, and Evaluation Setup

The datasets from DC Water and AlexRenew encompass time series variables spanning from 2019 to 2023; however, many of these time series contain missing values within this period. For model development and hyperparameter tuning, we divide the data into training and validation sets, allocating 80% for training and 20% for validation. Hyperparameters were primarily selected to minimize the forecasting error on the validation set while ensuring near-zero forecast bias by adjusting q^*, as previously discussed in the loss function section. Each training epoch comprises n_o "sub-epochs," determined experimentally, with each sub-epoch covering a complete pass through all available data. Specifically, n_o was set to 10 for DC Water and 15 for AlexRenew.

Algorithm 1 Training and Evaluating cP$_2$O

```
**Input:** Training data $\mathcal{D}\text{train} \in \mathbb{R}^{n \times T}$, Testing data $\mathcal{D}\text{test} \in \mathbb{R}^{n \times T}$, Context data $\mathcal{D}\text{C} \in \mathbb{R}^{m \times T}$, Initial model parameters $\Theta$, Batch size $\mathcal{B}$, Learning rate $\eta$, Max epochs $E_{\max}$
**Output:** Forecasts $\hat{\mathbf{u}} \in \mathbb{R}^{n \times T}$, Trained parameters $\Theta$, Loss metrics $\mathcal{L}$
**for** $e = 1$ to $E_{\max}$ **do**
    Split training data into batches $\mathcal{B}_i \in \mathbb{R}^{b \times T}$
    **for** each batch $\mathcal{B}_i$ **do**
        Initialize LSTM$_\Theta$ and ContextLSTM$_\Theta$
        **for** $t = 1$ to $T$ **do**
            Extract per series context including $\tilde{\mathbf{n}}_t$ and $\mu_t$
            Forecast $\hat{\mathbf{u}}_t = f(\mathcal{B}_i, \mathcal{D}_{\text{C},i}, \Theta)$
        **end for**
        Compute loss $\mathcal{L} = ||\hat{\mathbf{u}} - \mathbf{u}||_2^2 + \lambda \cdot \text{reg}(\Theta)$
        Update parameters $\Theta \leftarrow \Theta - \eta \nabla_\Theta \mathcal{L}$
    **end for**
    **if** saveResults == True **then**
        Save parameters $\Theta$ and intermediate results
    **end if**
**end for**
**return** Final forecasts $\hat{\mathbf{u}}$, trained parameters $\Theta$
```

We implement a training regimen that progressively increases batch sizes while simultaneously decreasing learning rates. Given the limited number of series, we begin with an initial batch size of 16, which is expanded to 64 starting at epoch 5. To further minimize the validation error, we employ a decaying learning rate schedule: 5×10^{-3} for epochs 1–5, 3×10^{-3} for epochs 6–7, 10^{-3} for epochs 8–9, and 10^{-4} from epoch 10 onward. We trained the model for a total of 50 epochs.

During each epoch, updates are conducted based on the average error accumulated over up to $T_b = 40$ forward steps, progressing one day at a time within each batch. The starting point within each batch is selected randomly, which may result in fewer than 40 steps. Each batch contains a randomly chosen subset of b series.

We explore multiple input sequence lengths during early experimentation to assess their influence on forecasting performance. Ultimately, we select a sequence length of 48 time steps (equivalent to a day of hourly observations) as it yields the best forecasting accuracy on the validation set. This choice also aligns with domain knowledge, as operators at WWTPs often consider 1–2 days of historical data sufficient to anticipate trends and anomalies for the next 4–6 hours. A longer history does not provide meaningful improvements but adds unnecessary computational cost, while shorter sequences result in reduced performance due to loss of temporal context.

The dimensions of the **c**-state and **h**-state were set to $s_z = 165$ and $s_h = 80$, respectively. These dimensions were determined through experimentation, beginning with $s_z = 100$ and $s_h = 50$, and incrementally increasing them to improve model performance. The output vector size O_v is calculated as the difference between the **c**-state and **h**-state sizes.

Our model architecture comprises three blocks, each containing one cell with dilation factors of 1, 2, and 4, as depicted in Figure 2c. These dilation factors were selected experimentally to correspond with the seasonal patterns in the data. We apply the pinball loss function for quantile regression, using quantile values of $q^* = 0.62$, $q_1 = 0.039$, and $q_2 = 0.981$. The weighting coefficient in the loss function is set to $\lambda = 0.35$ to ensure that the average central loss during training is higher than the losses for the lower and upper intervals.

We employed embedding layers for time-related variables with dimensions set to 10, determined through experimentation. The context batch size C_b was configured to 20, encompassing 5 variables from each of four distinct context groups. We include all series without missing values to simplify the implementation and enable the processing of all context series within a single batch.

Finally, we apply an ensemble size of $E = 20$ (chosen because it fits GPU memory and gives diminishing returns beyond 20); however, as few as 5 ensemble members are sufficient. Ensemble forecasts are combined using a straightforward mean aggregation. The ensemble version of our model, referred to as cP$_2$Oe, is designed to enhance the robustness and uncertainty quantification of forecasts produced by the base cP$_2$O architecture. While cP$_2$O represents a single DL model trained on the full dataset, cP$_2$Oe is multiple independently trained DL models of cP$_2$O; each initialized differently and potentially trained on different data subsets. This ensemble strategy helps reduce prediction variance and provides a more stable estimate in the presence of noise or anomalies in the input data. In our experiments, predictions from the ensemble members are aggregated using a simple mean operator to produce the final output. This approach, though computationally more intensive during inference, enables cP$_2$Oe to produce more reliable and generalized forecasts, particularly valuable in real-world, high-variability wastewater systems. A comparative analysis of cP$_2$O and cP$_2$Oe is presented in Section 6.

The training was performed using the Adam optimizer. The model's weight and bias matrices were initialized with specific dimensions to accommodate the input and hidden state sizes. Details of these matrices are provided in Appendix B. Training and evaluation steps are outlined in Algorithm 1.

The training and evaluation of all models were conducted on a machine equipped with *Intel Core i9-12900K*, 32 GB RAM, and an NVIDIA RTX 4090 GPU with 24 GB VRAM, using the CUDA platform for GPU acceleration. The average training time for each model was approximately 2 hours, depending on the dataset size and model configuration. For inference, each model produced predictions with an average latency of 15-20 seconds per sequence on the same machine, including preprocessing and postprocessing steps. This demonstrates the computational feasibility of our approach for real-time and near-real-time forecasting in operational wastewater systems.

5.4 Baseline Models

To assess the performance of our proposed model, we compare it with a variety of baseline models spanning statistical methods, ML, and DL techniques. The Naive model predicts the water level profile for day i by replicating the profile from day $i-7$. The ARIMA model [8] employs an autoregressive integrated moving average methodology, while the **ES** model [26] utilizes exponential smoothing. Another baseline, Prophet [65], applies modular additive regression with seasonal components and nonlinear trends.

In the realm of ML and DL models, we include several approaches. The GRNN [62] stands for General Regression Neural Network, and the SVM [18] model uses Support Vector Machines for time series regression tasks. Among recurrent neural networks, we evaluate the LSTM [30], an LSTM network designed to capture temporal dependencies. The MTGNN [70] is a Graph Neural Network tailored for multivariate time series forecasting. We also examine the N-BEATS model [49], a deep neural network with a hierarchical doubly residual architecture, and DeepAR [56], an autoregressive recurrent neural network for probabilistic forecasting.

Additionally, we assess the WaveNet model [67], which employs dilated convolutions for autoregressive forecasting. The XGBoost model [14] utilizes the eXtreme Gradient Boosting algorithm for regression tasks. Lastly, we include the **P_2O** model [35], a recent multivariate multi-step attention-LSTM model used as a baseline in time series forecasting.

These baseline models were evaluated under the same experimental setup. Some models, such as ARIMA, Prophet, and XGBoost, struggle with handling the complex seasonality and exogenous variables present in WWTP data. More advanced models such as LSTM, MTGNN, and N-BEATS perform better, but the inclusion of external context data in the proposed model offers significant advantages in predictive accuracy.

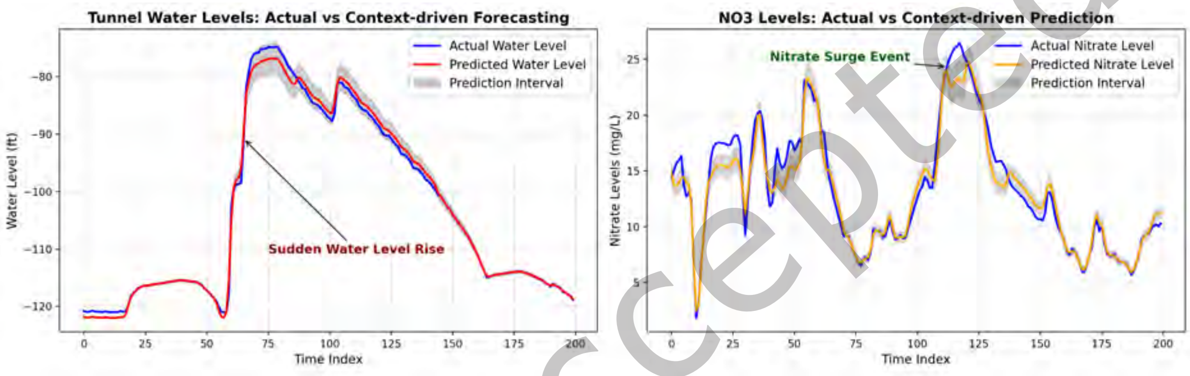


Fig. 4. The cP_2Oe forecasting results on WWTP data are presented, with actual values shown in blue, forecasts in red, and predictive intervals depicted in light gray shades (The 200-point window subset is used purely for visualization and clarity and does not indicate the model's input size during inference).

6 RESULTS: FORECASTING PERFORMANCE EVALUATION

The forecasting performance metrics, summarized in Table 2, include RMSE, MAPE, Interquartile Range of Absolute Percentage Error (iqrAPE), Standard Deviation of Percentage Error (StDPE), Peak Detection Rate (PDR), and Mean Percentage Error (MPE), as defined in Equation (24). Our proposed model is evaluated in two variations: cP_2O and its ensemble version, cP_2Oe. For comparison, we also include the performance of our previous model, P_2O, as reported in prior work [35].

$$\begin{aligned}
&\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} && \text{MdAPE} = \text{median}\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\right) \\
&\text{PDR} = \frac{\text{Number of Correctly Detected Peaks}}{\text{Total Number of True Peaks}} && \text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \\
&\text{iqrAPE} = Q_3\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\right) - Q_1\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\right) && \text{MPE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i} \times 100\right) \\
&\text{StDPE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i} \times 100 - \text{MPE}\right)^2}
\end{aligned} \tag{24}$$

Table 2. Performance metrics comparison across models for tunnel water level forecasting and nitrate level predictions

	Tunnel water level forecast (DC Water)								NO_3 level prediction (AlexRenew)						
Model	RMSE	MAPE	MPE	iqrAPE	StDPE	MdAPE	PDR	Score	RMSE	MAPE	MPE	iqrAPE	StDPE	MdAPE	Score
Naive [38]	6.98	5.15	0.25	3.35	7.82	4.94	50.25	1.02	3.45	5.20	-0.20	3.40	7.60	5.00	5.10
ARIMA [8]	4.71	3.32	-0.02	3.05	5.20	3.08	75.85	17.89	2.90	3.15	**0.02**	3.10	5.35	3.20	15.25
ES [26]	4.34	3.18	0.01	2.74	5.10	2.92	77.45	26.25	2.75	3.05	0.05	2.70	5.25	2.90	20.30
SVM [18]	3.86	2.89	-0.15	2.55	4.54	2.65	85.55	34.70	2.30	2.55	0.08	2.45	4.55	2.35	40.15
LGBM [31]	3.75	2.82	0.02	2.60	4.41	2.63	83.63	37.30	2.50	2.70	-0.10	2.50	4.70	2.55	35.60
XGB [14]	3.62	2.71	**0.00**	2.44	4.30	2.48	84.24	47.15	2.35	2.60	0.04	2.40	4.60	2.45	43.25
Prophet [65]	4.09	3.60	-0.12	2.85	4.76	3.45	60.55	9.77	3.20	3.80	0.15	3.20	6.90	4.40	8.10
N-WE [73]	3.26	2.55	-0.14	2.32	4.12	2.35	86.45	57.33	2.25	2.50	0.10	2.30	4.35	2.30	45.50
GRNN [62]	3.25	2.53	-0.10	2.30	4.12	2.32	86.63	60.40	2.20	2.45	-0.05	2.25	4.30	2.35	48.75
MLP [19]	3.86	2.97	0.05	2.80	4.67	2.73	80.85	33.68	2.55	2.85	0.12	2.75	5.15	2.65	30.20
LSTM [30]	3.75	2.82	0.01	2.60	4.41	2.63	83.63	37.30	2.50	2.75	-0.07	2.55	4.80	2.60	32.85
ANFIS [4]	4.98	3.70	-0.12	3.69	6.25	3.22	65.36	12.89	3.10	3.65	0.18	3.60	6.55	3.20	12.75
MTGNN [70]	3.69	2.95	-0.02	2.62	4.49	2.70	82.28	32.80	2.40	2.90	0.08	2.60	4.90	2.65	33.50
DeepAR [56]	4.84	3.50	-0.50	3.00	4.95	3.31	70.78	16.99	3.00	3.40	-0.45	2.95	5.25	3.25	18.65
WaveNet [67]	4.15	3.12	-0.80	2.77	4.58	2.90	78.06	28.15	2.70	3.00	-0.75	2.70	4.95	2.90	25.80
N-BEATS [49]	3.57	2.62	0.04	2.41	4.21	2.40	85.05	48.95	2.30	2.55	-0.03	2.35	4.50	2.40	46.85
P_2O [35]	3.35	2.71	0.18	2.25	3.90	2.48	86.51	84.15	2.20	2.35	0.03	2.40	4.60	2.50	60.50
cP_2O	2.96	2.15	-0.18	1.75	3.22	1.55	93.54	90.85	2.00	1.95	0.05	1.90	3.25	1.70	89.32
cP_2Oe	**2.94**	**2.10**	-0.18	**1.71**	**3.19**	**1.50**	**93.54**	**94.85**	**1.95**	**1.90**	0.05	**1.85**	**3.20**	**1.65**	**90.35**

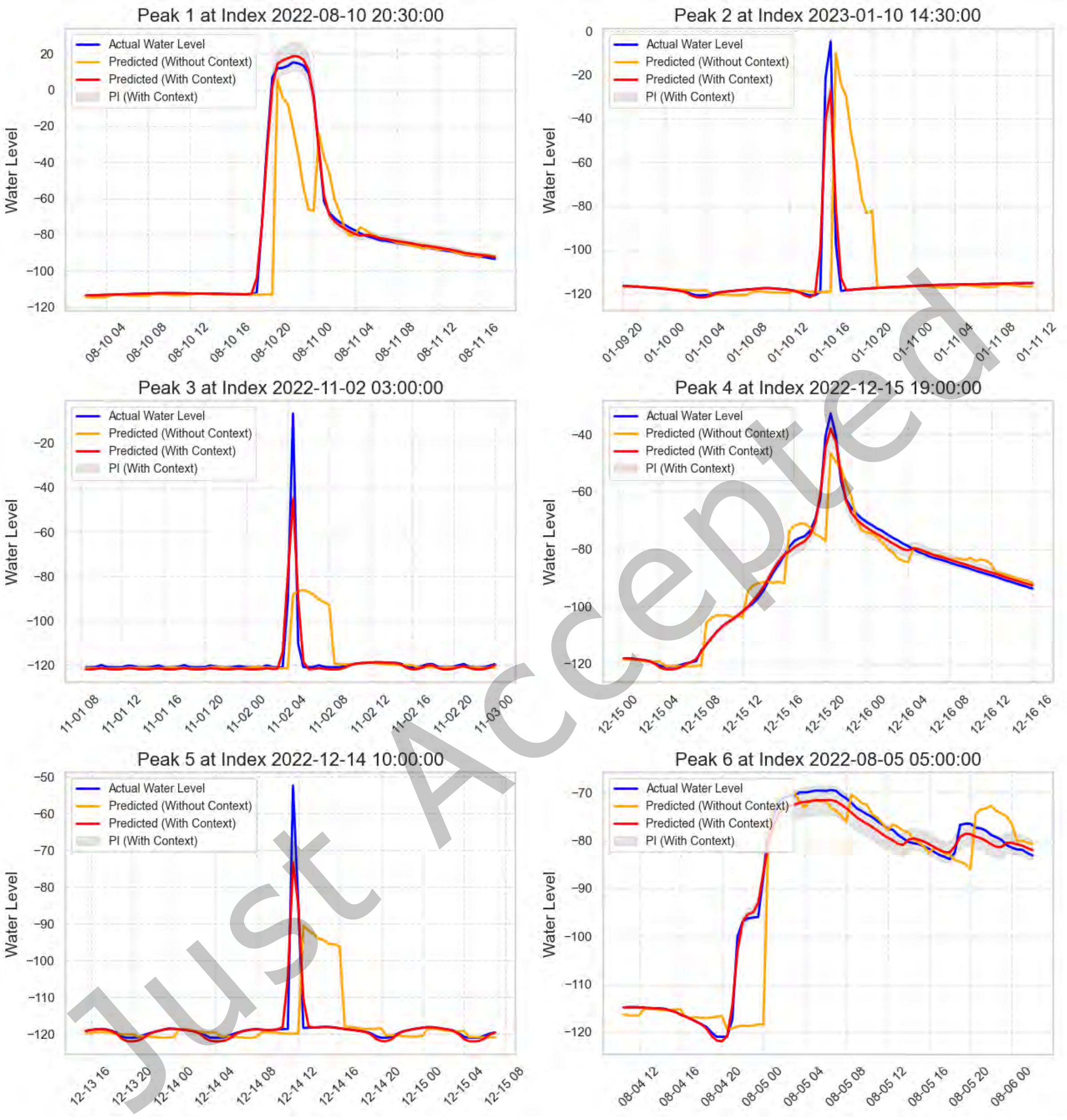
It is important to note that separate models were trained for each dataset—DC Water and AlexRenew—using the same architecture and methods, but different training data specific to each WWTP. This approach allows the models to capture the unique characteristics and patterns inherent in each dataset while leveraging the strengths of the proposed architecture.

As shown in Table 2, our proposed hybrid models outperform state-of-the-art methods across most metrics on both datasets. Particularly, the ensemble model cP_2Oe demonstrates superior accuracy, achieving the lowest MAPE values of 2.10% for the DC Water model and 1.90% for the AlexRenew model.

Figure 4 illustrates the predictions and dynamic predictive intervals for a test period. Observations fall within the predictive intervals approximately 90.51% ± 3.21% of the time, while 5.86% ± 1.85% fall below and 3.63% ± 1.47% above. The narrow iqrAPE and StDPE further highlight each cP_2Oe model's ability to maintain consistent predictions for its respective dataset.

The proposed cP_2Oe models significantly improve accuracy compared to the earlier version, P_2O, when trained on their respective datasets. For the DC Water dataset, MAPE is reduced from 2.71% to 2.10%, a reduction of approximately 22%, and RMSE is reduced by about 12% (from 3.35 to 2.94). For the AlexRenew dataset, MAPE is reduced from 2.35% to 1.90%, a reduction of approximately 19%, solidifying the cP_2Oe models' positions as the top-performing models for each dataset.

The lowest StDPE and iqrAPE values in Table 2 demonstrate that cP_2Oe models deliver more consistent and less variable predictions compared to baseline models such as Prophet and DeepAR on both datasets. For instance, it achieves the lowest iqrAPE (1.71% for DC Water and 1.85% for AlexRenew) and StDPE (3.19% for DC Water and


Peak 1 at Index 2022-08-10 20:30:00

Peak 2 at Index 2023-01-10 14:30:00

Peak 3 at Index 2022-11-02 03:00:00

Peak 4 at Index 2022-12-15 19:00:00

Peak 5 at Index 2022-12-14 10:00:00

Peak 6 at Index 2022-08-05 05:00:00

Fig. 5. Actual (blue) vs. predicted tunnel water levels without context (P_2O-orange) and with context (cP_2O- red). The shaded region shows the ±95% prediction intervals. Clearly observe around peak events (vertical dashed line) that predictions incorporating context (red) consistently track actual values closer than those without context (orange), especially immediately after peak occurrence.

3.20% for AlexRenew), highlighting their robustness in controlling the spread of percentage errors specific to each dataset.

Although the DC Water model exhibits a slightly negative MPE value (indicating a minor tendency to under-predict), the MPE is balanced in the AlexRenew model, suggesting that the predictions are well-calibrated for both datasets. Additionally, our models are more robust to outliers, as observed from the MdAPE values; cP_2Oe attains the lowest MdAPE values, such as 1.50% for DC Water and 1.65% for AlexRenew. The cP_2Oe model for DC Water also excels in PDR, identifying 93.54% of peaks, indicating its robustness in detecting extreme events such as floods.

These results validate our hypothesis for Research Question 2, demonstrating that the cP_2O model is both adaptable and effective across different WWTPs and forecasting tasks. By training separate models using the same architecture and methods on diverse datasets—specifically for tunnel water level forecasting at DC Water and nitrate level prediction at AlexRenew, we consistently achieve high accuracy tailored to each dataset's unique characteristics and forecasting objectives.

To evaluate the statistical significance of our forecasting performance, we perform a pairwise Giacomini-White test for conditional predictive ability. The test scores in Table 2 were calculated by comparing the forecast errors, including MAPE, RMSE, MPE, and PDR, of the models to determine if one model's predictive performance was statistically superior to another's at the $\alpha = 0.05$ significance level. For the DC Water experiment, we include PDR in the comparison; for AlexRenew, we apply the rest of the key metrics. The results, labeled as "*Score*" in Table 2, reveal that each cP_2Oe model achieved significantly lower forecasting errors than other models in over 90% of pairwise comparisons for their respective datasets. This confirms the models' superiority, particularly the cP_2Oe configuration, which benefits from exogenous input, validating the design choice of dynamic, context-driven adjustments.

Figure 5 showcases dynamic adjustments during peak events for the DC Water dataset. Notably, the proposed model cP_2O aligns well with extreme water level increases compared to our previous model P_2O, a key requirement for critical event forecasting. This plot and the results from Table 2 support the hypothesis of Research Question 4: using a quantile loss function employed in cP_2O effectively reduces forecast bias during peak events or extreme conditions and improves uncertainty estimation in multi-step-ahead forecasting.

6.1 Ablation Study

To assess the contributions of key components in the cP_2O model, we conducted an ablation study by systematically removing the context stage, the attention mechanism, and the dilated LSTM layers. This study was performed on both the DC Water and AlexRenew datasets, with results presented in Table 3.

(1) **Without Context Stage:** Removing the context stage led to performance declines across both datasets. For DC Water, MAPE increased from 2.10% to 2.48%, RMSE from 2.94 to 3.32, and PDR decreased from 93.54% to 85.50%. For AlexRenew, MAPE rose from 1.90% to 2.25% and RMSE from 1.90 to 2.20. This underscores the critical role of external contextual data (e.g., weather, influent characteristics) in enhancing prediction accuracy and robustness. The consistent impact supports our hypothesis for Research Question 1: incorporating contextual data significantly improves the accuracy of short-term predictions in WWTPs.

(2) **Without Attention Mechanism:** Excluding the attention mechanism increased errors on both datasets. In DC Water, RMSE rose to 3.50, MAPE to 2.60%, and StDPE to 3.80%; in AlexRenew, RMSE increased to 2.10, MAPE to 2.05%, and StDPE to 3.50%. This highlights the models' reduced capacity to dynamically adjust to varying input importance over time, leading to less accurate and more variable predictions. These results confirm our hypothesis for Research Question 3: integrating an attention mechanism in cP_2O enhances the models' ability to weigh input features effectively, thereby improving forecasting accuracy.

(3) **Without Dilated LSTM Layers:** Dilated links are mainly valuable for the fast, high spikes—see peaks 2,3,5 in Figure 5. A plain three-layer LSTM that sees the last 24 h is already able to track the slower-rising peaks 1, 4, 6, so headline metrics fall only modestly when dilation is removed. In the full test set, DC Water RMSE rises from 2.94 to 3.80, MAPE from 2.10% to 2.85%, and PDR drops from 93.54% to 80.50%; AlexRenew shows a similar but smaller shift. However, when we score *only* the spike windows (peaks 2, 3, 5), RMSE worsens by ≈28% and PDR by 13%, confirming that dilated links are critical for sudden anomalies. The global drop looks small because (i) the forecast horizon is short (4–6 h); (ii) the network size is unchanged; (iii) context inputs (weather, river flow) already supply much long-range information; and (iv) averages are dominated by normal timesteps, which dilute the benefit on the rare spikes. Overall, the dilated LSTM layer is essential for capturing rapid, high-impact events while still supporting long-term trends.

Table 3. Impact of key components of cP_2O on forecasting performance

Model Variant	**Tunnel water level forecast (DC Water)**				**NO_3 level prediction (AlexRenew)**		
	MAPE (%)	**RMSE**	**PDR** (%)	**StDPE** (%)	**MAPE** (%)	**RMSE**	**StDPE** (%)
cP_2O (Full Model)	**2.10**	**2.94**	**93.54**	**3.19**	**1.90**	**1.90**	**3.20**
Without Context Stage	2.48	3.32	85.50	3.50	2.25	2.20	3.50
Without Attention Mechanism	2.60	3.50	83.00	3.80	2.05	2.10	3.50
Without Dilated LSTM Layers	2.85	3.80	80.50	4.00	2.20	2.30	3.80

Overall, the ablation study confirms that each component significantly enhances the cP_2O models' performance across both datasets. Including external context data improves adaptability to changing conditions, the attention mechanism allows dynamic weighting of input features, and dilated LSTM layers enable capturing temporal dependencies over multiple scales. The consistent performance degradation when components are removed validates our design choices and supports our research hypotheses. These findings demonstrate the effectiveness of the full cP_2O architecture for short-term forecasting in wastewater treatment plants across different settings.

To summarize the key findings and explicitly answer our research questions:

- **RQ1**: Does the integration of external contextual data improve forecasting accuracy in WWTPs?
  Yes. The inclusion of contextual features (e.g., weather and influent parameters) significantly reduced error metrics across both datasets, as confirmed by the ablation study.
- **RQ2**: Can the proposed cP_2O model generalize across different WWTP environments?
  Yes. The model demonstrated consistent high performance on both DC Water and AlexRenew datasets, showcasing its adaptability to different operational and contextual conditions.
- **RQ3**: Does incorporating attention mechanisms enhance the model's dynamic responsiveness to varying input significance?
  Yes. Removing the attention module led to a noticeable increase in forecasting errors and reduced interpretability, confirming its contribution to improved performance.
- **RQ4**: Does the use of quantile loss and ensemble modeling improve forecasting under extreme events?
  Yes. The ensemble version (cP_2Oe) with quantile-based loss showed higher peak detection rates (PDR) and narrower prediction intervals, indicating stronger performance during rare but impactful events.

6.2 Attention-Based Context Importance for Six Incidents (DC Water)

While the ablation study highlights the overall contributions of context, attention, and dilation, it does not directly reveal which *context group* (Weather, River Data, Economic, Demographic, or Main Plant Data) is most influential for specific peak events. To address this gap, we examined the internal attention weights for the six major incidents **in the DC Water dataset only**, as illustrated in Figure 5[3]. Table 4 shows the normalized attention distribution across the five context groups for these events, where larger values indicate that the model places greater emphasis on that specific group.

Overall, **Weather** and **River Data** often dominate in short-horizon surges driven by external rainfall or upstream inflows, while **WWTP Data** (e.g., pump operations, flow routing) takes precedence in events characterized by internal operational triggers. Although **Demographic** and **Economic** factors receive lower attention scores for these specific peak windows, they still provide useful background for baseline usage patterns. Collectively, these findings corroborate the results of our ablation experiments and reinforce that quickly changing external conditions—especially weather and river levels—are critical for capturing extreme peak incidents in the DC Water system.

Table 4. Attention-based context importance (average weights) on six different events from Figure 5

Incident	Context Groups Attention Weights					Primary Driver
	Weather	**River**	**Economic**	**Demographic**	**WWTP**	
#1	0.26	0.22	0.11	0.07	0.34	WWTP Data
#2	0.33	0.27	0.10	0.08	0.22	Weather Data
#3	0.21	0.25	0.15	0.09	0.30	WWTP Data
#4	0.28	0.30	0.07	0.06	0.29	River Data
#5	0.32	0.25	0.09	0.05	0.29	Weather Data
#6	0.24	0.28	0.12	0.06	0.30	WWTP Data

Key Observations.

- **Weather vs. River Data:** Incidents #2 and #5 are dominated by *Weather* factors (e.g., heavy rainfall), whereas #4 highlights *River Data* as the main external driver, suggesting surges in upstream flow or tidal backflow played a critical role.
- **WWTPs Dynamics:** Incidents #1, #3, and #6 show relatively larger allocations to internal operational features such as pump scheduling or active flow routing, indicating that utility decisions helped shape the peak formation.
- **Demographic and Economic Context:** Although these factors have lower attention for short-horizon peaks, they remain valuable for characterizing daily/weekly flow baselines. They can also capture long-term usage shifts, even if overshadowed by immediate environmental triggers.

The distribution of attention weights in Table 4 confirms the importance of *rapidly changing external signals* and *dynamic internal operations* for accurate peak detection at DC Water. By adapting its focus among these diverse data sources, the model is better equipped to predict and mitigate critical events in real time.

[3] We used the 2- 4 hour window preceding each peak to calculate average attention scores. These six events are unique to the DC Water system; hence, no similar analysis was conducted for AlexRenew

7 DISCUSSION AND CONCLUSIONS

In this paper, we introduced cP_2O, a hybrid DL model designed for short-term forecasting in wastewater treatment plants (WWTPs). Our model integrates contextual data through a two-stage framework that combines dynamic smoothing and hierarchical dilated LSTMs with an attention mechanism. By leveraging both internal sensor data and exogenous variables—such as weather conditions and influent characteristics—the model captures both long-term and short-term temporal dependencies, as well as external influences on WWTP operations. We trained separate models for the DC Water and AlexRenew datasets using the same architecture and methods, demonstrating the model's adaptability to different WWTPs.

The results, summarized in Table 2, show that our proposed models outperform several baseline approaches—including ARIMA, Exponential Smoothing, Prophet, and other recent ML and DL models—in terms of accuracy and robustness across both datasets. Specifically, the ensemble model cP_2Oe achieved the lowest MAPE values of 2.10% for the DC Water model and 1.90% for the AlexRenew model, indicating superior predictive performance. The models also exhibited lower RMSE, iqrAPE, and StDPE values, highlighting their consistency and precision.

Integrating context variables significantly enhanced the models' ability to provide reliable predictions. By introducing a two-stage framework that processes both internal and external data, cP_2O can adapt to complex temporal patterns, including multiple seasonality and sudden fluctuations caused by exogenous factors such as weather events or operational changes. This capability is particularly important in WWTPs, where accurate short-term predictions help optimize operations, manage resource allocation, and prevent system overload during high-demand periods.

The ablation study confirmed that each component of the cP_2O model—including the context stage, the attention mechanism, and the dilated LSTM layers—contributed significantly to improved performance on both datasets. Removing the context stage led to increases in MAPE and RMSE and a decrease in Peak Detection Rate (PDR), underscoring the importance of external contextual information. Excluding the attention mechanism resulted in higher error metrics and reduced the models' ability to focus on relevant input features dynamically. Eliminating the dilated LSTM layers impaired the models' capacity to capture temporal dependencies over multiple time scales. These findings validate our design choices and support our research hypotheses, demonstrating that the integration of contextual data, the dynamic weighting of input features, and capturing temporal dependencies is critical for enhancing forecasting accuracy in WWTPs.

Our model has been successfully deployed at DC Water; detailed information about its deployment performance and evaluation can be found in Appendix C. Future work will focus on collaborating with other regional facilities to test the model in diverse contexts, improving its scalability to handle larger datasets, and incorporating additional categories of external variables to further enhance forecasting accuracy.

While the models demonstrated strong performance across a range of metrics, there are limitations to consider. The models' ability to effectively handle multiple time series (beyond a certain number of variables) may be constrained by computational resources and the complexity of the architecture. However, for datasets of the size used in this study, cP_2O proved to be a highly efficient and powerful approach. Future work will explore methods to improve computational efficiency, such as employing model compression techniques or distributed training frameworks, and enhancing the model's robustness to missing or noisy contextual data. Additionally, incorporating real-time adaptive learning mechanisms could allow the model to self-update in response to changing operational conditions, further boosting its utility in dynamic environments.

We also plan to conduct a detailed study to understand the impact of different context groups on forecasting performance during extreme events. By dynamically altering the context series and analyzing their contributions, we aim to make the model more explainable and trustworthy for utility operators. This approach will not only enhance operational confidence but also provide insights into the critical factors driving model predictions, particularly under high-stress scenarios.

In conclusion, the cP_2O model offers a robust and adaptable solution for short-term forecasting in WWTPs. By effectively integrating contextual data and advanced DL techniques, the models offer significant improvements over existing methods, aiding WWTP operators in making informed decisions and optimizing plant operations. The consistent performance across different datasets underscores the model's generalizability and effectiveness, validating our approach for enhancing forecasting accuracy in diverse WWTP environments.

DECLARATION OF COMPETING INTEREST

The authors declare no competing interests.

ACKNOWLEDGMENTS

We sincerely thank Pierric Parret from DC Water for providing the dataset, utility operational demonstrations, and assistance with deploying the forecasting model. We are equally grateful to David Roberts from AlexRenew for supplying their utility operational data. Our appreciation also extends to the A3 Lab (https://ai.bse.vt.edu/) members for their continuous support and valuable technical feedback.

DATA AVAILABILITY

Due to the confidential nature of the data and related security considerations, data samples are available upon approval from the utility and by request to the authors. Our team's data repositories are available here: https://github.com/AI-VTRC/.

REFERENCES

[1] Jonathan M Aitken, Mathew H Evans, Rob Worley, Sarah Edwards, Rui Zhang, Tony Dodd, Lyudmila Mihaylova, and Sean R Anderson. 2021. Simultaneous localization and mapping for inspection robots in water and sewer pipe networks: A review. *IEEE access* 9 (2021), 140173–140198.

[2] Gulzar Alam, Ihsanullah Ihsanullah, Mu Naushad, and Mika Sillanpää. 2022. Applications of artificial intelligence in water treatment for optimization and automation of adsorption processes: Recent advances and prospects. *Chemical Engineering Journal* 427 (2022), 130011.

[3] Siddharth Arora and James W. Taylor. 2018. Rule-based autoregressive moving average models for forecasting load on special days: A case study for France. *European Journal of Operational Research* 266, 1 (2018), 259–268.

[4] Ulker Guner Bacanli, Mahmut Firat, and Fatih Dikbas. 2009. Adaptive neuro-fuzzy inference system for drought forecasting. *Stochastic Environmental Research and Risk Assessment* 23 (2009), 1143–1154.

[5] Robert K Bastian, Peter E Shanaghan, and Brian P Thompson. 2020. Use of wetlands for municipal wastewater treatment and disposal–Regulatory issues and EPA policies. *Constructed Wetlands for Wastewater Treatment* (2020), 265–278.

[6] Alice Botturi, E Gozde Ozbayram, Katharina Tondera, Nathalie I Gilbert, Pascale Rouault, Nicolas Caradot, Oriol Gutierrez, Saba Daneshgar, Nicola Frison, Çağrı Akyol, et al. 2021. Combined sewer overflows: A critical review on best practice and innovative solutions to mitigate impacts on environment and human health. *Critical Reviews in Environmental Science and Technology* 51, 15 (2021), 1585–1618.

[7] Oussama Boussif, Ghait Boukachab, Dan Assouline, Stefano Massaroli, Tianle Yuan, Loubna Benabbou, and Yoshua Bengio. 2024. Improving* day-ahead* Solar Irradiance Time Series Forecasting by Leveraging Spatio-Temporal Context. *Advances in Neural Information Processing Systems* 36 (2024).

[8] George EP Box, Gwilym M Jenkins, and Gregory C Reinsel. 1970. *Time series analysis: forecasting and control*. Holden-Day San Francisco.

[9] Tom Boyle and Andrew Ravenscroft. 2012. Context and deep learning design. *Computers & Education* 59, 4 (2012), 1224–1233.

[10] Edward C Budd and Daniel B Radner. 1975. The Bureau of Economic Analysis and Current Population Survey Size Distributions: Some Comparisons for 1964. In *The Personal Distribution of Income and Wealth*. NBER, 449–559.

[11] Western Regional Climate Center, Hydrometeorological Prediction Center, and Environmental Modeling Center. 2003. National Oceanic and Atmospheric Administration (NOAA). *NOAA* (2003).

[12] Fi-John Chang, Li-Chiu Chang, and Jui-Fa Chen. 2023. Artificial intelligence techniques in hydrology and water resources management. , 1846 pages.

[13] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A Hasegawa-Johnson, and Thomas S Huang. 2017. Dilated recurrent neural networks. *Advances in neural information processing systems* 30 (2017).

[14] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 785–794.

[15] Tuoyuan Cheng, Fouzi Harrou, Farid Kadri, Ying Sun, and Torove Leiknes. 2020. Forecasting of wastewater treatment plant key features using deep learning-based models: A case study. *Ieee Access* 8 (2020), 184475–184485.

[16] Kyunghyun Cho et al. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[17] Kyunghyun Cho, Bart Van Merriënboer, Çaglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[18] Harris Drucker, Christopher JC Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1997. Support vector regression machines. *Advances in neural information processing systems* 9 (1997).

[19] Grzegorz Dudek. 2016. Neural networks for pattern-based short-term load forecasting: A comparative study. *Neurocomputing* 205 (2016), 64–74.

[20] Grzegorz Dudek, Paweł Pełka, and Slawek Smyl. 2021. A hybrid residual dilated LSTM and exponential smoothing model for midterm electric load forecasting. *IEEE Transactions on Neural Networks and Learning Systems* 33, 7 (2021), 2879–2891.

[21] Sahar Farmanifard, Ali Asghar Alesheikh, and Mohammad Sharif. 2023. A context-aware hybrid deep learning model for the prediction of tropical cyclone trajectories. *Expert Systems with Applications* 231 (2023), 120701.

[22] J Flores, B Arcay, and J Arias. 2000. An intelligent system for distributed control of an anaerobic wastewater treatment process. *Engineering Applications of Artificial Intelligence* 13, 4 (2000), 485–494.

[23] Guangtao Fu, Yiwen Jin, Siao Sun, Zhiguo Yuan, and David Butler. 2022. The role of deep learning in urban water management: A critical review. *Water Research* 223 (2022), 118973.

[24] Rivka Galchen. 2015. Weather underground. *New Yorker* 13 (2015), 34–40.

[25] Ruina Gao, Liang Du, Ponnuthurai Nagaratnam Suganthan, Qing Zhou, and Kwok-Fai Yuen. 2022. Random vector functional link neural network-based ensemble deep learning for short-term load forecasting. *Expert Systems with Applications* 206 (2022), 117784.

[26] Everette S. Gardner. 1985. Exponential smoothing: The state of the art. *Journal of Forecasting* 4, 1 (1985), 1–28.

[27] Alireza Gheisi, Mark Forsyth, and Gh Naser. 2016. Water distribution systems reliability: A review of research literature. *Journal of Water Resources Planning and Management* 142, 11 (2016), 04016047.

[28] Andrew C. Harvey. 1990. *Forecasting, structural time series models and the Kalman filter.* Cambridge university press.

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[30] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[31] Yun Ju, Guangyu Sun, Quanhe Chen, Min Zhang, Huixian Zhu, and Mujeeb Ur Rehman. 2019. A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting. *Ieee Access* 7 (2019), 28309–28318.

[32] Jongwoo Kim, Jaeyoon Moon, Euisung Hwang, and Piljae Kang. 2019. Recurrent inception convolution neural network for multi short-term load forecasting. *Energy and Buildings* 194 (2019), 328–341.

[33] Anne B Koehler, Ralph D Snyder, and J Keith Ord. 2001. Forecasting models and prediction intervals for the multiplicative Holt–Winters method. *International Journal of Forecasting* 17, 2 (2001), 269–286.

[34] Vaia I Kontopoulou, Athanasios D Panagopoulos, Ioannis Kakkos, and George K Matsopoulos. 2023. A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet* 15, 8 (2023), 255.

[35] Ajay Kulkarni, Mehmet Yardimci, Md Nazmul Kabir Sikder, and Feras A Batarseh. 2023. P2O: AI-driven framework for managing and securing wastewater treatment plants. *Journal of Environmental Engineering* 149, 9 (2023), 04023045.

[36] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*. 95–104.

[37] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[38] Michael S Lewis-Beck and Tom W Rice. 1984. Forecasting presidential elections: A comparison of naive models. *Political behavior* 6 (1984), 9–21.

[39] Wenya Li, Qing Shi, Muhammad Sibtain, Daoliang Li, and Delene E. Mbanze. 2020. A hybrid forecasting model for short-term power load based on sample entropy, two-phase decomposition, and whale algorithm optimized support vector regression. *IEEE Access* 8 (2020), 166907–166921.

[40] Yiqi Liu, Pedram Ramin, Xavier Flores-Alsina, and Krist V Gernaey. 2023. Transforming data into actionable knowledge for fault detection, diagnosis and prognosis in urban wastewater systems with AI techniques: A mini-review. *Process Safety and Environmental Protection* 172 (2023), 501–512.

[41] Arti Malviya and Dipika Jaspal. 2021. Artificial intelligence as an upcoming technology in wastewater treatment: a comprehensive review. *Environmental Technology Reviews* 10, 1 (2021), 177–187.

[42] Manel Massaoudi et al. 2021. A novel stacked generalization ensemble-based hybrid LGBM-XGB-MLP model for short-term load forecasting. *Energy* 214 (2021), 118874.

[43] Anthony Njuguna Matheri, Belaid Mohamed, Freeman Ntuli, Esther Nabadda, and Jane Catherine Ngila. 2022. Sustainable circularity and intelligent data-driven operations and control of the wastewater treatment plant. *Physics and Chemistry of the Earth, Parts A/B/C* 126 (2022), 103152.

[44] Michael W McCracken and Serena Ng. 2016. FRED-MD: A monthly database for macroeconomic research. *Journal of Business & Economic Statistics* 34, 4 (2016), 574–589.

[45] Hao Miao, Yan Fei, Senzhang Wang, Fang Wang, and Danyan Wen. 2022. Deep learning based origin-destination prediction via contextual information fusion. *Multimedia Tools and Applications* (2022), 1–17.

[46] Anita Mohanty, Subrat Kumar Mohanty, and Ambarish G Mohapatra. 2024. Real-Time Monitoring and Fault Detection in AI-Enhanced Wastewater Treatment Systems. In *The AI Cleanse: Transforming Wastewater Treatment Through Artificial Intelligence: Harnessing Data-Driven Solutions*. Springer, 165–199.

[47] Balamurali Murugesan, Kaushik Sarveswaran, Sharath M Shankaranarayana, Keerthi Ram, Mohanasankar Sivaprakasam, et al. 2020. A context based deep learning approach for unbalanced medical image segmentation. In *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 1949–1953.

[48] Rejoan Kobir Nishan, Shapla Akter, Rayhanul Islam Sony, Md Mozammal Hoque, Meratun Junnut Anee, and Amzad Hossain. 2024. Development of an IoT-based multi-level system for real-time water quality monitoring in industrial wastewater. *Discover Water* 4, 1 (2024), 43.

[49] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2019. N-BEATS: Neural basis expansion analysis for time series forecasting. In *International Conference on Learning Representations (ICLR)*.

[50] Tim N Palmer and David L T Anderson. 1994. The prospects for seasonal forecasting—A review paper. *Quarterly Journal of the Royal Meteorological Society* 120, 518 (1994), 755–793.

[51] Piotr Piotrowski, Dariusz Baczyński, Mateusz Kopyt, and Tomasz Gulczyński. 2022. Advanced ensemble methods using machine learning and deep learning for one-day-ahead forecasts of electric energy production in wind farms. *Energies* 15, 4 (2022), 1252.

[52] Mary C Rabbitt. 1989. *The United States Geological Survey, 1879-1989*. Vol. 1050. US Government Printing Office.

[53] Michael Ratcliffe, Charlynn Burd, Kelly Holder, and Alison Fields. 2016. Defining rural at the US Census Bureau. *American community survey and geography brief* 1, 8 (2016), 1–8.

[54] Rakiba Rayhana, Yutong Jiao, Amirhossein Zaji, and Zheng Liu. 2020. Automated vision systems for condition assessment of sewer and water pipelines. *IEEE Transactions on Automation Science and Engineering* 18, 4 (2020), 1861–1878.

[55] Soma Safeer, Ravi P Pandey, Bushra Rehman, Tuba Safdar, Iftikhar Ahmad, Shadi W Hasan, and Asmat Ullah. 2022. A review of artificial intelligence in water purification and wastewater treatment: Recent advancements. *Journal of Water Process Engineering* 49 (2022), 102974.

[56] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[57] K Sathya, K Nagarajan, G Carlin Geor Malar, S Rajalakshmi, and P Raja Lakshmi. 2022. A comprehensive review on comparison among effluent treatment methods and modern methods of treatment of industrial wastewater effluent from different sources. *Applied Water Science* 12, 4 (2022), 70.

[58] Lauren Setar and Matthew MacFarland. 2012. Top 10 fastest-growing industries. *Special Report IbisWorld* (2012).

[59] Slawek Smyl, Grzegorz Dudek, and Pawel Pelka. 2022. ES-dRNN with dynamic attention for short-term load forecasting. In *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[60] Adir Solomon, Mor Kertis, Bracha Shapira, and Lior Rokach. 2022. A deep learning framework for predicting burglaries based on multiple contextual factors. *Expert Systems with Applications* 199 (2022), 117042.

[61] Vitor Sousa, José P Matos, and Natércia Matias. 2014. Evaluation of artificial intelligence tool performance and uncertainty for predicting sewer structural condition. *Automation in Construction* 44 (2014), 84–91.

[62] Donald F Specht. 1991. A general regression neural network. *IEEE transactions on neural networks* 2, 6 (1991), 568–576.

[63] Chhayly Sreng. 2024. *Trustworthy Soft Sensing in Water Supply Systems using Deep Learning*. Ph. D. Dissertation. Virginia Tech.

[64] Gary Stein and Avelino J Gonzalez. 2014. Learning in context: enhancing machine learning with context-based reasoning. *Applied intelligence* 41 (2014), 709–724.

[65] Sean J Taylor and Benjamin Letham. 2018. Forecasting at scale. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1515–1524.

[66] Moshe Unger, Alexander Tuzhilin, and Amit Livne. 2020. Context-aware recommendations based on deep learning frameworks. *ACM Transactions on Management Information Systems (TMIS)* 11, 2 (2020), 1–15.

[67] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*. 125.

[68] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. 2023. Micn: Multi-scale local and global context modeling for long-term series forecasting. In *The eleventh international conference on learning representations*.
[69] Past Weather. 2009. National Weather Service. *NWS* (2009).
[70] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xinghua Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 753–763.
[71] Jie Yan, Yongqian Liu, Shuang Han, Yimei Wang, and Shuanglei Feng. 2015. Reviews on uncertainty analysis of wind power forecasting. *Renewable and Sustainable Energy Reviews* 52 (2015), 1322–1330.
[72] Liping Yang, Joshua Driscol, Sarigai Sarigai, Qiusheng Wu, Christopher D Lippitt, and Melinda Morgan. 2022. Towards synoptic water monitoring systems: a review of AI methods for automating water body detection and water quality monitoring using remote sensing. *Sensors* 22, 6 (2022), 2416.
[73] Dragomir Yankov, Dennis DeCoste, and Eamonn Keogh. 2006. Ensembles of nearest neighbor forecasts. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*. Springer, 545–556.
[74] Manzhu Yu, Qunying Huang, and Zhenlong Li. 2024. Deep learning for spatiotemporal forecasting in Earth system science: a review. *International Journal of Digital Earth* 17, 1 (2024), 2391952.

APPENDIX A: CONTEXT DEFINITION FOR WWTPS

In this appendix, we provide a detailed mathematical definition of the context variables used in the cP$_2$O model, and explain how they are integrated into the forecasting model.

A.1 Context Variables Representation

Let $\mathbf{D}_t \in \mathbb{R}^N$ represent the vector of WWTP internal variables at time t, where N is the number of internal variables (e.g., influent flow rate, water levels). The context variables, denoted by $\mathbf{C}_t \in \mathbb{R}^M$, capture external factors influencing the WWTP, where M is the number of context variables (e.g., weather data, river flow rates, demographic data).

The combined input vector at time t is defined as:

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{D}_t \\ \mathbf{C}_t \end{bmatrix} \in \mathbb{R}^{N+M}$$

The context variables $\mathbf{C}_t$ can include, but are not limited to:

- *Weather Data*: Precipitation (P_t), temperature (T_t), humidity (H_t), wind speed (W_t).
- *River Data*: River flow rates (R_t), water levels (L_t).
- *Demographic Data*: Population density (ρ_t), urbanization rate (U_t).
- *Economic Data*: Industrial output (I_t), employment rates (E_t).

These variables provide external information that influences WWTP operations and are crucial for improving forecasting accuracy.

A.2 Integration into the Model

The cP$_2$O model incorporates context variables through an enriched input vector and a context extraction stage. The model consists of two main components:

(1) *Context Extraction Stage*: Processes context variables to generate a context vector $\mathbf{r}_t$.
(2) *Forecasting Stage*: Utilizes both internal variables and the context vector to make predictions.

A.2.1 Context Extraction Stage. The context extraction stage employs a dilated LSTM network to process context variables over a sequence of past time steps. Let $\Omega_t^{\text{ctx}} = \{t - T_c + 1, \dots, t\}$ represent the context input window of length T_c. The context LSTM processes the sequence $\{\mathbf{C}_\tau\}_{\tau=t-T_c+1}^{t}$ to generate the context vector $\mathbf{r}_t$:

$$\mathbf{r}_t = \text{LSTM}_{\text{ctx}}\left(\{\mathbf{C}_\tau\}_{\tau=t-T_c+1}^{t};\ \boldsymbol{\theta}_{\text{ctx}}\right) \in \mathbb{R}^u$$

where $\boldsymbol{\theta}_{\text{ctx}}$ are the parameters of the context LSTM, and u is the dimension of the context vector.

A.2.2 Forecasting Stage Input Enhancement. The context vector $\mathbf{r}_t$ is concatenated with the internal variables to form the enhanced input vector for the forecasting stage:

$$\mathbf{x}'_t = \begin{bmatrix} \mathbf{D}_t \\ \mathbf{r}_t \end{bmatrix} \in \mathbb{R}^{N+u}$$

The forecasting LSTM processes the sequence $\{\mathbf{x}'_\tau\}_{\tau=t-T_f+1}^{t}$, where $\Omega_t^{\text{in}} = \{t - T_f + 1, \dots, t\}$ is the input window of length T_f, to generate the forecasted output $\hat{\mathbf{y}}_{t+1}$:

$$\hat{\mathbf{y}}_{t+1} = \text{LSTM}_{\text{fcast}}\left(\{\mathbf{x}'_\tau\}_{\tau=t-T_f+1}^{t};\ \boldsymbol{\theta}_{\text{fcast}}\right)$$

where $\boldsymbol{\theta}_{\text{fcast}}$ are the parameters of the forecasting LSTM.

A.3 Attention Mechanism in Context Integration

The forecasting LSTM incorporates an attention mechanism to dynamically weigh the contributions of the context vector and internal variables. At each time step t, attention weights $\boldsymbol{\alpha}_t \in \mathbb{R}^{N+u}$ are computed:

$$\boldsymbol{\alpha}_t = \text{softmax}\left(\mathbf{W}_a \mathbf{h}_{t-1} + \mathbf{b}_a\right)$$

where $\mathbf{W}_a \in \mathbb{R}^{(N+u)\times s_h}$ and $\mathbf{b}_a \in \mathbb{R}^{N+u}$ are learnable parameters, s_h is the dimension of the hidden state, and $\mathbf{h}_{t-1}$ is the hidden state from the previous time step.

The enhanced input vector is then modulated by the attention weights:

$$\tilde{\mathbf{x}}_t = \boldsymbol{\alpha}_t \odot \mathbf{x}'_t$$

where $\odot$ denotes element-wise multiplication.

The forecasting LSTM processes the modulated input $\tilde{\mathbf{x}}_t$:

$$\mathbf{h}_t,\ \mathbf{c}_t = \text{LSTM}_{\text{fcast}}\left(\tilde{\mathbf{x}}_t,\ \mathbf{h}_{t-1},\ \mathbf{c}_{t-1};\ \boldsymbol{\theta}_{\text{fcast}}\right)$$

where $\mathbf{h}_t$ and $\mathbf{c}_t$ are the hidden and cell states at time t.

A.4 Output Generation

The final forecast is generated by applying a linear transformation to the hidden state $\mathbf{h}_t$:

$$\hat{\mathbf{y}}_{t+1} = \mathbf{W}_o \mathbf{h}_t + \mathbf{b}_o$$

where $\mathbf{W}_o \in \mathbb{R}^{s_y \times s_h}$ and $\mathbf{b}_o \in \mathbb{R}^{s_y}$ are the output weight matrix and bias vector, and s_y is the dimension of the output vector.

A.5 Summary of Notation

- $\mathbf{D}_t \in \mathbb{R}^N$: Internal WWTP variables at time t.
- $\mathbf{C}_t \in \mathbb{R}^M$: Context variables at time t.
- $\mathbf{x}_t \in \mathbb{R}^{N+M}$: Combined input vector.
- $\mathbf{r}_t \in \mathbb{R}^u$: Context vector extracted by the context LSTM.
- $\mathbf{x}'_t \in \mathbb{R}^{N+u}$: Enhanced input vector for forecasting.
- $\boldsymbol{\alpha}_t \in \mathbb{R}^{N+u}$: Attention weights.
- $\tilde{\mathbf{x}}_t \in \mathbb{R}^{N+u}$: Modulated input vector after applying attention.
- $\mathbf{h}_t$, $\mathbf{c}_t$: Hidden and cell states of the forecasting LSTM.
- $\hat{\mathbf{y}}_{t+1}$: Forecasted output at time $t+1$.
- $\boldsymbol{\theta}_{\text{ctx}}$, $\boldsymbol{\theta}_{\text{fcast}}$: Parameters of the context and forecasting LSTMs, respectively.

A.6 Mathematical Formulation of the Forecasting Function

Combining the components, the forecasting function can be summarized as:

$$\hat{\mathbf{y}}_{t+1} = f\left(\mathbf{D}_t,\ \mathbf{C}_t;\ \boldsymbol{\theta}\right) = \mathbf{W}_o \mathbf{h}_t + \mathbf{b}_o$$

where $\mathbf{h}_t$ is obtained through the following steps:

$$\mathbf{r}_t = \mathrm{LSTM}_{\mathrm{ctx}}\left(\{\mathbf{C}_\tau\}_{\tau=t-T_c+1}^{t};\ \boldsymbol{\theta}_{\mathrm{ctx}}\right) \tag{25}$$

$$\mathbf{x}'_t = \begin{bmatrix} \mathbf{D}_t \\ \mathbf{r}_t \end{bmatrix} \tag{26}$$

$$\boldsymbol{\alpha}_t = \mathrm{softmax}\left(\mathbf{W}_a \mathbf{h}_{t-1} + \mathbf{b}_a\right) \tag{27}$$

$$\tilde{\mathbf{x}}_t = \boldsymbol{\alpha}_t \odot \mathbf{x}'_t \tag{28}$$

$$\mathbf{h}_t,\ \mathbf{c}_t = \mathrm{LSTM}_{\mathrm{fcast}}\left(\tilde{\mathbf{x}}_t,\ \mathbf{h}_{t-1},\ \mathbf{c}_{t-1};\ \boldsymbol{\theta}_{\mathrm{fcast}}\right) \tag{29}$$

Equation (25) computes the context vector $\mathbf{r}_t$ by processing the sequence of context variables through the context LSTM. This captures temporal patterns in the external factors influencing the WWTP. Equation (26) forms the enhanced input vector by concatenating the internal variables $\mathbf{D}_t$ with the context vector $\mathbf{r}_t$. Equation (27) calculates the attention weights $\boldsymbol{\alpha}_t$ based on the previous hidden state $\mathbf{h}_{t-1}$, allowing the model to dynamically focus on the most relevant features at each time step. Equation (28) modulates the enhanced input vector with the attention weights, effectively weighting each feature according to its importance. Equation (29) processes the modulated input through the forecasting LSTM to update the hidden and cell states, capturing the temporal dependencies in the data. The final forecast $\hat{\mathbf{y}}_{t+1}$ is generated by applying a linear transformation to the updated hidden state.

A.7 Dimensions Clarification

For clarity, we specify the dimensions of the key variables:

- $\mathbf{D}_t \in \mathbb{R}^N$: Column vector of internal variables.
- $\mathbf{C}_t \in \mathbb{R}^M$: Column vector of context variables.
- $\mathbf{r}_t \in \mathbb{R}^u$: Column vector from context LSTM.
- $\mathbf{x}'_t \in \mathbb{R}^{N+u}$: Column vector (concatenation of $\mathbf{D}_t$ and $\mathbf{r}_t$).
- $\boldsymbol{\alpha}_t \in \mathbb{R}^{N+u}$: Column vector of attention weights.
- $\tilde{\mathbf{x}}_t \in \mathbb{R}^{N+u}$: Column vector of modulated input.
- $\mathbf{h}_t \in \mathbb{R}^{s_h}$: Hidden state vector.
- $\hat{\mathbf{y}}_{t+1} \in \mathbb{R}^{s_y}$: Output vector.

A.8 Context Variables Examples

As previously mentioned, the context variables $\mathbf{C}_t$ can include various external factors:

- *Weather Data* (M_{weather} variables):
- Precipitation (P_t)
- Temperature (T_t)
- Humidity (H_t)
- Wind speed (W_t)
- *River Data* (M_{river} variables):
- River flow rates (R_t)
- Water levels (L_t)
- *Demographic Data* (M_{demo} variables):
- Population density (ρ_t)
- Urbanization rate (U_t)
- *Economic Data* (M_{econ} variables):
- Industrial output (I_t)

- Employment rates (E_t)

Total context variables: $M = M_{\text{weather}} + M_{\text{river}} + M_{\text{demo}} + M_{\text{econ}}$.

A.9 Learning Objective

The model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\text{ctx}}, \boldsymbol{\theta}_{\text{fcast}}, \mathbf{W}_a, \mathbf{b}_a, \mathbf{W}_o, \mathbf{b}_o\}$ are learned by minimizing the loss function defined in the main text, typically involving the pinball loss for quantile regression:

$$\mathcal{L} = \sum_t \ell\,(y_t,\ \hat{y}_t)$$

where y_t is the observed value, and $\hat{y}_t$ is the predicted value.

APPENDIX B HYPERPARAMETERS

In this appendix, we provide a detailed parameter choice for the cP$_2$O and other baseline models.

B.1. cP$_2$O Hyperparameter Choice

- Batch size (B): Initially set to 16 and raised to 64 after the fourth epoch, based on experimental results. Further increases were restricted due to the dataset size.
- Initial seasonality adjustments: Computed as ratios between the first input window's values and the mean values of that window.
- Initial smoothing factors: $S_\alpha = -4$ and $S_\beta = 0.45$, chosen based on the average smoothing coefficients dynamic behavior.
- Learning rate schedule: Initially assigned to 5×10^{-3} for the first five epochs, then reduced progressively to 10^{-4} by epoch 9.
- Dilation rates: Experimentally set to 1, 2, and 4, following the rule of increasing dilations, ideally in an exponential fashion.
- Embedding dimensions: Set to 10, determined through experimentation for time-related variables.
- Embedding layer weight and bias matrices: Shared across paths, defined by $W \in \mathbb{R}^{90\times4}$ and $b \in \mathbb{R}^4$.
- Training steps per batch (T_b): Set to 40, based on trial and error.
- Total epochs: Set to 50, during which both batch size increases and learning rate decreases.
- Sub-epoch calculation (S_e): Set to 10 for DC Water and 15 for AlexRenew
- Updates per epoch (U_e): Set to 1500 iterations to ensure significant accuracy improvement per epoch.
- Optimizer: The Adam optimization algorithm, chosen based on experimentation.
- Loss function parameter (λ): Set to 0.35, ensuring that the average central loss during training is higher than the losses for the lower and upper intervals.
- Pinball loss quantiles: Experimentally adjusted to $q^* = 0.62$, $q = 0.039$, and $q = 0.981$.
- Context batch size (C_b): Set to 20, indicating the number of exogenous variables.
- Hidden state size (H_s): Set to 165 for the c-state and 80 for the h-state, based on experimentation.
- Output vector size (O_v): Calculated as the difference between c-state and h-state sizes.
- Forecasting stage weight and bias matrices: Comprising 12 sets of matrices (four per layer across three layers): $W \in \mathbb{R}^{n\times150}$, $V \in \mathbb{R}^{70\times150}$, $U \in \mathbb{R}^{70\times150}$, and $b \in \mathbb{R}^{150}$. For the first layer, $n = 193$; for subsequent layers, $n = 273$.
- Forecasting stage output layer weights and biases: Defined by $W \in \mathbb{R}^{80\times74}$ and $b \in \mathbb{R}^{74}$.
- Context stage weight and bias matrices: Also comprising 12 sets of matrices, with $W \in \mathbb{R}^{m\times150}$, $V \in \mathbb{R}^{70\times150}$, $U \in \mathbb{R}^{70\times150}$, and $b \in \mathbb{R}^{150}$. In the first layer, $m = 247$; in other layers, $m = 327$.
- Context path output layer weights and biases: Defined by $W \in \mathbb{R}^{80\times5}$ and $b \in \mathbb{R}^5$.

- Ensemble method: Simple averaging across the ensemble members.
- Ensemble size (E): Set to 20, depending on the scenario for each experiment.

B.2. Baseline Models Hyperparameter Choice

- ES: Divided into 24 hourly time series, predicted with ‘ets‘ in R.
- Naive: Using the previous step’s value as the prediction.
- ARIMA: Splitting into 24 time series (one per hour), with forecasts generated using ‘auto.arima‘ in R.
- LGBM: Utilizes ‘LGBMRegressor‘ with ‘max_depth‘ set to 10, 500 iterations, and a learning rate of 0.01.
- XGB: Uses ‘XGBRegressor‘ with ‘max_depth‘ of 10 and learning rate 0.05, with additional features.
- SVM: Predictions made using ‘fitrsvm‘ in Matlab. Key hyperparameters are fine-tuned.
- N-WE: Implemented in MATLAB, with a fixed pattern length of 24.
- GRNN: Similar to N-WE, implemented in Matlab with cross-validated smoothing parameters.
- MLP: Uses ‘feedforwardnet‘ in Matlab, with a single hidden layer and trained with Bayesian regularization.
- LSTM: Implemented using ‘lstm‘ in Matlab, with fixed 24 neurons.
- ANFIS: Forecasts with ‘anfis‘ in Matlab, using a Sugeno fuzzy inference system.
- MTGNN: Implemented with default settings from the repository at https://github.com/nnzhan/MTGNN.
- DeepAR: Uses GluonTS with ‘context_length‘ set to seven times the ‘prediction_length‘.
- Prophet: Forecasts generated using the ‘prophet‘ package in R with default settings.
- WaveNet: Uses GluonTS with default hyperparameters.
- N-BEATS: Implemented via GluonTS with ‘context_length‘ set to seven times the ‘prediction_length‘.

APPENDIX C: DEPLOYMENT OF THE FORECASTING MODEL IN DC WATER SCADA SYSTEM

In this appendix, we detail the deployment process of the cP$_2$O forecasting model and present the results obtained during real-time operation in a WWTP setting. The deployment aimed to evaluate the model’s practical performance and its ability to assist operators in decision-making processes.

C.1 Deployment Steps at DC Water

The forecasting model was deployed within the operational environment of the DC Water treatment facility. The deployment architecture included the following components:

- Data Acquisition System: Real-time data streams from sensors and external sources (e.g., weather stations, river flow gauges) were collected via a SCADA system.
- Processing Server: A dedicated Amazon Web Services (AWS) instance equipped with high-performance computational CPUs hosted the forecasting model and managed data processing tasks.
- Model Integration: cP$_2$O model was implemented in Python 3.8 using the PyTorch 1.8 deep learning framework. It was seamlessly integrated into the processing pipeline to receive real-time data inputs and generate forecasts.
- User Interface: An AWS-hosted web application provides a user-friendly interface for monitoring forecasts and interacting with the system.

C.2 Real-Time Forecasting Process

The deployed system operated on a rolling basis, generating forecasts every hour with a 4-hour ahead horizon. The process involved:

(1) Data Ingestion: The latest data from internal sensors ($\mathbf{D}_t$) and context variables ($\mathbf{C}_t$) were ingested into the system.

(2) Preprocessing: Data were cleaned to handle missing values, and features were scaled based on the training data parameters.
(3) Forecast Generation: The cP$_2$O model processed the input data to generate forecasts for the next 4 hours, including prediction intervals.
(4) Visualization and Alerts: Forecasts were visualized on the dashboard, and alerts were triggered if predicted water levels exceeded predefined thresholds.

C.3 Deployment Results

The model's performance was monitored over a period of two months during varying operational conditions, including dry weather and heavy rainfall events. The key results are summarized below.

C.3.1 Overall Performance Metrics. The model maintained high accuracy during deployment, with performance metrics consistent with those observed during validation. Table 5 presents the aggregated metrics over the deployment period.

Table 5. Performance metrics during deployment

Metric	**Overall Value**	**Dry Weather**	**Rainfall Events**
MAPE (%)	2.05	1.80	2.60
RMSE	3.35	3.10	3.90
PDR (%)	95.5	N/A	95.5
Predictive Interval (%)	92.5 ± 1.50	94.0 ± 1.20	90.0 ± 1.80

The model demonstrated slightly higher errors during rainfall events due to increased variability in inflow rates. However, the prediction intervals effectively captured the uncertainty, maintaining a conditional probability close to the desired 90%.

C.3.2 Case Study: Heavy Rainfall and Coastal Flood Events. During two significant flooding events on January 9th and 10th, 2024, at DC Water, the model's ability to forecast inflow surges was critically evaluated. Figure 6 displays the actual and forecasted water levels, prediction intervals, and key event annotations.

The model effectively captured the sharp increase in water levels during the events, accurately predicting both the coastal flood peak (6.19' MLLW) and the Rock Creek flood crest (8.04 ft). These predictions provided operators with a 4-hour advance warning, enabling proactive management actions such as adjusting pump operations and diverting flows to storage tunnels. The integration of context variables, particularly during rainfall events, proved crucial in enhancing prediction accuracy and supporting critical decision-making under extreme weather conditions.

C.4 Challenges and Mitigations

During deployment, several challenges were encountered:

- Data Quality Issues: Occasional sensor malfunctions led to missing or erroneous data. This was mitigated by implementing real-time data validation checks and fallback strategies using historical averages.
- Model Retraining: To maintain accuracy, the model was retrained weekly using the latest data. Automated retraining pipelines were set up to facilitate this process.
- System Integration: Integrating the model into the existing SCADA system required careful coordination to ensure compatibility and data security.

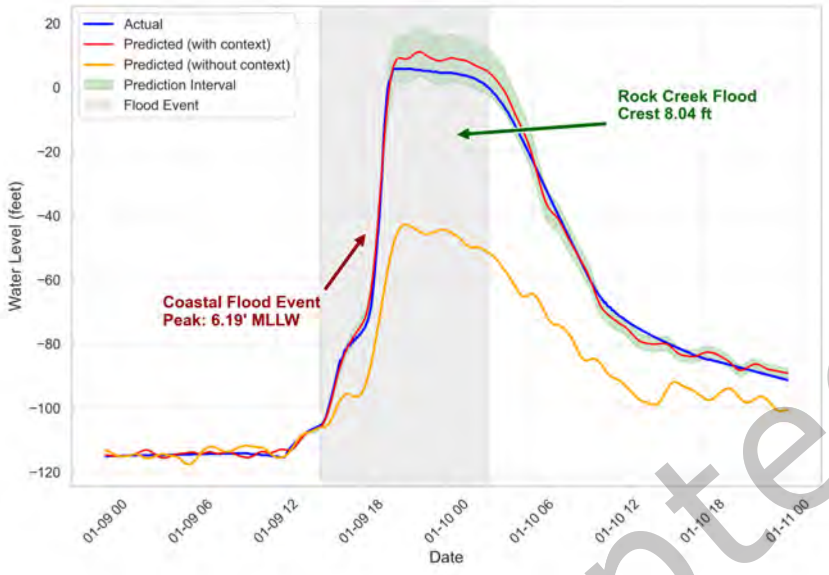


Fig. 6. Forecasts during a heavy rainfall and coastal flood event. The actual water levels are shown in blue, forecasts with context in red, and forecasts without context in orange. The light green shaded area represents the prediction intervals, indicating the model's uncertainty in the forecasts. The gray-shaded regions highlight the duration of flood events. Notably, the prediction intervals capture the uncertainty around both the peak and recession phases, illustrating the model's ability to account for variability during extreme weather events. Without the contextual data, the model (orange line) fails to accurately predict the peak of the flood event. This is because critical information about external factors, such as rainfall and coastal conditions, is absent in the utility data alone. By incorporating exogenous variables, the context-aware model (red line) successfully identifies and forecasts the peak, demonstrating the importance of external data in enhancing predictive accuracy during extreme events.

C.5 Future Improvements

Based on the deployment experience, the following improvements are planned:

- Incorporation of Additional Context Variables: Including more granular weather forecasts and upstream flow data to enhance prediction accuracy during extreme events.
- Enhanced Anomaly Detection: Integrating anomaly detection mechanisms to identify and handle outliers or unexpected patterns in the data.
- Scalability Enhancements: Optimizing the model for deployment across multiple facilities with varying configurations and data sources.

C.6 Conclusion of Deployment

The deployment of the cP_2O model demonstrated its practical applicability and effectiveness in a real-world WWTP environment. The model provided accurate short-term forecasts, aiding operators in making informed decisions and improving operational efficiency. The positive outcomes reinforce the value and need of integrating AI-driven forecasting models into wastewater management systems across the country.