Modeling phosphorous dynamics in a wastewater treatment process using Bayesian optimized LSTM

Hansen, Laura Debel; Stokholm-Bjerregaard, Mikkel; Durdevic, Petar

Published in:
Computers & Chemical Engineering

DOI (link to publication from Publisher):
10.1016/j.compchemeng.2022.107738

Creative Commons License
CC BY 4.0

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Link to publication from Aalborg University

Citation for published version (APA):
Hansen, L. D., Stokholm-Bjerregaard, M., & Durdevic, P. (2022). Modeling phosphorous dynamics in a wastewater treatment process using Bayesian optimized LSTM. *Computers & Chemical Engineering*, *160*, Article 107738. https://doi.org/10.1016/j.compchemeng.2022.107738

Computers and Chemical Engineering 160 (2022) 107738

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng

Modeling phosphorous dynamics in a wastewater treatment process using Bayesian optimized LSTM

Laura Debel Hansen[a,b,*], Mikkel Stokholm-Bjerregaard[b], Petar Durdevic[a]

[a] AAU Energy, Aalborg University, Niels Bohrs vej 8, Esbjerg, 6700, Denmark
[b] Krüger A/S, Indkildevej 6C, Aalborg Sø, 9210, Denmark

ARTICLE INFO

Article history:
Received 25 October 2021
Revised 2 February 2022
Accepted 14 February 2022
Available online 17 February 2022

Keywords:
Dynamic model
Neural networks
Time series prediction
Hyperparameter tuning
Full scale plant data
Phosphorus

ABSTRACT

This study presents a systematic framework to develop data-driven models for phosphorus concentration in a full-scale wastewater treatment plant (WWTP). The dynamics of wastewater treatment exhibit nonlinear behavior, and are time varying, non-stationary, and coupled in a complex manner, which makes them difficult to predict using mechanistic models. Two long short-term memory (LSTM) models are proposed. The first estimates the phosphorus concentration using data describing environmental conditions and process operation, and the second model which additionally utilizes the previous phosphorus measurement. Additionally, the hyperparameters are tuned using Bayesian optimization, as this is an effective tool to determine the best model and prevent over-fitting and long training duration of the data-driven models. The two models show good prediction performances and are suitable to predict up to 24 hours into the future, with R^2 close to 0.7-0.8 for data well presented in the training data set.

1. Introduction

Wastewater treatment plants (WWTP) continuously face new challenges as more stringent effluent standards arises. Mathematical modeling of WWTPs has, consequently, been widely established as a method used for research, plant design, optimization, design of control strategies, and general understanding of the processes (Garikiparthy et al., 2016; Stentoft et al., 2019). Several mechanistic models (Henze et al., 2000) and benchmark simulation models (Gernaey et al., 2004; Gernaey and Jeppsson, 2014; Vrecko et al., 2006) have been proposed to model the treatment processes of WWTPs. These mechanistic models, also known as white-box models, are derived by first principles meaning they are ”transparent”, as they explicitly explains relations between the model input and output. The contrast to the white-box model is the black-box model which is derived solely from measurement data. In black-box models, the structure and the parameters are both determined from data, meaning that no prior knowledge of the system is needed.

This paper addresses the use of data-driven modeling (DDM) by developing two black-box models to describe the dynamics of wastewater treatment processes by utilizing data from a case plant. The case study of this work considers removal of phosphorus in the form of phosphate (PO_4^{3-}-P) at a WWTP which utilizes the activated sludge process (ASP). This focus point is chosen because modeling and understanding the process of phosphorus removal is of key importance, as phosphorus contributes to eutrophication effects if led directly out to surrounding surface waters (Wilfert et al., 2015). Process understanding and accurate simulation models yields, furthermore, the best foundation for high performance process control.

In this study, we present a systematic approach to model wastewater processes using data-driven models which are optimized to determine the best model parameters for the given task. Furthermore, we wish to discuss how the performance and utilization of a model depends on which input data is passed to the model. As such, the main contributions of this study are the following:

1. Using more than one year of measurements from a full-scale WWTP, we propose two multivariate LSTM models which estimates the phosphorus concentration in an ASP. The two models takes inputs such as: temperature, flow rate, dissolved oxygen, etc. The first model functions as an estimator; hence, its performance does not rely on actual phosphorus measurements for future step ahead predictions. The second model differs from the first, as previous phosphorus measurements are included as an input. This increases the model performance when the

[*] Corresponding author.
E-mail addresses: ldh@energy.aau.dk (L.D. Hansen), pdl@energy.aau.dk (P. Durdevic).

https://doi.org/10.1016/j.compchemeng.2022.107738

measurements are available and makes the model performance dependent on true sensor measurements. With these two options for modeling of complex industrial processes, we leave a choice for the developer to choose the most appropriate model for the desired application.

2. We propose a systematic approach for dynamic modeling of multivariate systems, and suggests that many case studies and complex processes could be modeled effectively using this approach.
3. To ensure the optimum structure for the specific application, we optimize the architecture and training algorithm of the two LSTM models using Bayesian optimization for hyperparameter tuning.

The rest of the paper is organized as follows: A thorough review of related literature is presented in Section 2. The case plant is introduced in Section 3 along with the collected data and computing material. Section 4 presents the basic theory of the LSTM and highlights similarities of the two proposed LSTM networks. Furthermore, the method of hyperparameter tuning using Bayesian optimization is explained in this section. Section 5 presents experimental results of the estimation and future step ahead predictions using the proposed LSTM models. The results are compared and the two models are discussed both from a modeling and plant operation point of view. Finally, Section 6 summarizes this paper.

2. Related work

Modeling of WWTPs has been ongoing for decades with the activated sludge model (ASM) as one of the first successful attempts to model the ASP (Henze et al., 2002). Mechanistic models like the ASM and the subsequent augmented ASM models (ASM2, ASM2d, ASM3) have provided a foundation for many WWTP optimization studies (Hansen et al., 2021; Aguado et al., 2009; Bongards, 1999; Cao and Yang, 2018; Dias and Ferreira, 2009; Besharati Fard et al., 2020; Gaya et al., 2013; Han et al., 2018; Husin et al., 2021; Hwangbo et al., 2021; Keskitalo and Leiviskä, 2015; Meng et al., 2021; Newhart et al., 2019; Pisa et al., 2019b; Stentoft et al., 2019; Zhao et al., 2020). Regarding phosphorus dynamics, Kazadi Mbamba et al. presents a simulation study in which phosphorus is modeled using a mechanistic kinetic model based on equilibrium approach of ion speciation and ion pairing (Kazadi Mbamba et al., 2016). For this study, the benchmark simulation model no. 2 (BSM2) was calibrated in five steps and used to model the case plant. Similar studies have been published, where mechanistic models have been calibrated and utilized for simulation purposes (Solon et al., 2017; Feldman et al., 2017; Kazadi Mbamba et al., 2019; Flores-Alsina et al., 2021). However, the problem remains a major challenge in both academia and industry as the wastewater processes are highly nonlinear, coupled and time-varying dynamic systems containing both physical and biochemical reactions and large time delay features (Newhart et al., 2019; Kazadi Mbamba et al., 2019). Furthermore, a major drawback using mechanistic models of WWTPs is the complexity of the models, demand for prior knowledge and model calibration before utilization (Dias and Ferreira, 2009). In order to describe coupled biological and chemical processes in the system, the mechanistic models often involve a very large number of state variables. To calibrate and utilize known mechanistic models for WWTPs, the model developer must acquire information obtained from laboratory-scale experiments, full-scale plant data such as those collected by online sensors, and default values from the literature (Stentoft et al., 2019; Kazadi Mbamba et al., 2016; 2019). As these three types of information can be difficult to collect, the developer may be interested in another approach to model the processes of the WWTP.

Data-driven modeling (DDM) and system identification is a research area based on computational intelligence (CI) and machine-learning (ML) methods with overlapping contributions from artificial intelligence (AI), soft computing (SC) and data mining (DM) (Solomatine et al., 2008). Recent advancements in DDM and nonlinear system identification provides the developer and the wastewater industry with an alternative to the mechanistic models (Newhart et al., 2019). As a result, DDM has gained acceptability and is increasingly applied to model WWTPs (Husin et al., 2021; Zhao et al., 2020; Newhart et al., 2019; Dürrenmatt and Gujer, 2012). Another reason for the recent increase of applied DDM to WWTP may be due to the already existing data acquisition systems. In several countries - especially in Denmark - WWTPs are monitored using online sensor measurements of the different wastewater processes, making DDM very advantageous as models can be developed for simulation purposes using only data from daily operation of the plant.

In the field of DDM, the multilayer perceptron (MLP) (Nelles, 2020) network and recurrent neural networks (RNN) (Calderon et al., 1996; Narendra and Parthasarathy, 1990) are established methods for DDM. The MLP is the most widely known and used neural network architecture (Solomatine et al., 2008), and referrers to a network of nodes where information is passed forward in the structure (known as feed-forward neural networks (FFNN)). Some of the first examples where DDM is applied to WWTPs are from the 1990s (Côté et al., 1995; Bongards, 1999); however, with the increasing accession to powerful computers, the application of DDM has increased rapidly though the years (Zhao et al., 2020). The most popular method to model the WWTP processes using DDM is the FFNN, and has been applied to pilot scale and full-scale WWTPs in various studies (Meng et al., 2021; Husin et al., 2021; Pisa et al., 2019b; Cao and Yang, 2018; Han et al., 2018; Wunsch et al., 2018; Aguado et al., 2009; Gaya et al., 2013; Bongards, 1999; Besharati Fard et al., 2020; Hwangbo et al., 2020; 2021).

In DDM and ML literature, the FFNN has been outperformed by the recurrent neural network (RNN) when used for dynamic modeling and system identification (Goodfellow et al., 2016; Sinha et al., 2000; Lecun et al., 2015). The RNN takes advantage of its feed-back loop to store past input information (Wang, 2017). As a result, the model complexity and number of layers can be reduced, compared to the FFNN. However, the conventional RNN (also known as a vanilla RNN) is difficult to train as the stored information over several time intervals is limited in a short-term manner due to small or large error feed-back (Wang, 2017). This is also known as the vanishing/exploding gradient issue. To solve the problem of the vanishing gradient, Hochreiter and Schmidhuber proposed the long short-term memory network (LSTM) in 1997 (Hochreiter and Schmidhuber, 1997), which avoids the vanishing/exploding gradient issue by using three gated units in the neuron structure. The LSTM is a very promising neural network architecture with internal dynamics controlled by a forget gate, input gate and output gate (Nelles, 2020; Wang, 2017; Hochreiter and Schmidhuber, 1997; Pascanu et al., 2013). Most studies regarding LSTMs are published after 2015 (Smagulova and James, 2019), however, despite its only recent entry to the field of DDM, the architecture has already shown great performance in modeling and prediction of wastewater treatment processes. The LSTM network has been used to predict ammonium and total nitrogen concentrations in WWTP benchmark scenarios using BSM2 (Pisa et al., 2019a; 2019b), and in full scale case studies to estimate WWTP N_2O emissions (Hwangbo et al., 2021) from a univariate regression perspective. Published work tend to concern benchmark simulation studies or univariate regression tasks using the LSTM as a prediction model. Hence, no studies has been found where data from a full scale WWTP is used to test the estimation and prediction per-

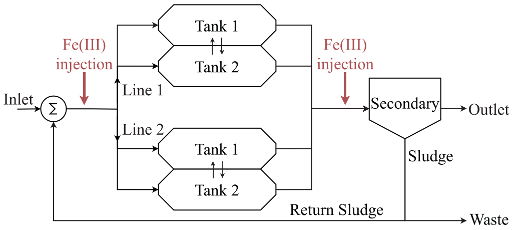


Fig. 1. Schematic of the flow lines in the system.

formance of the LSTM model performing a multivariate regression task.

Furthermore, the authors found that few studies regarding DDM of WWTPs discuss the *hyperparameters* of the model. The training algorithm of data-driven models has parameters which heavily affect how the model is optimized to predict the output. Those are called hyperparameters, and there exists several methods to tune and determine the optimum hyperparameters. Manual tuning is the most basic approach and usually only performs well with good process understanding and expert knowledge within the field (Goodfellow et al., 2016). Another approach is automatic hyperparameter tuning, where several methods exists; Grid search (Bergstra et al., 2011), random search (Bergstra and Bengio, 2012) and model based optimization such as Bayesian optimization (Snoek et al., 2012).

3. Case study: Kolding WWTP

The plant of interest is Kolding central WWTP located in Agtrup Denmark which has a capacity of 125,000 population equivalent (PE), and is currently operating with a load of approximately 65.5% (Blue Kolding, 2021). To remove phosphorus from the wastewater, the plant utilizes both chemical and biological phosphorus removal. Chemical phosphorus removal (CPR) is the most common treatment strategy in which phosphorus is removed from the wastewater using chemical precipitation, where metal salts are added to the wastewater causing the phosphorus to be incorporated into suspended solids (SS) (Bunce et al., 2018). Biological phosphorus removal (BPR) is performed using alternating anaerobic and aerobic/anoxic conditions. Notoriously, BPR is one of the more difficult processes to control in WWTPs, which has been experienced at several WWTPs where the process suddenly can release huge amounts of phosphorus in the effluent for reasons unknown (Ingildsen et al., 2006).

3.1. Operation and design

The wastewater is led from the primary treatment to the biological treatment and distributed via two lines to four tanks in which the ASP takes place. This is illustrated in Fig. 1.

The plant is currently monitored and controlled in Hubgrade^{TM}(Krüger, 2021); an online supervisory control and data acquisition (SCADA) system developed by Krüger[1]. Chemical phosphorus removal (CPR) is utilized to precipitate phosphorus into suspended solids (SS) by addition of iron (Fe(III)) at two locations, which is illustrated in Fig. 1 with red arrows. In addition to chemical precipitation, the plant also utilizes a particular strategy for biological phosphorus removal (BPR) developed by Krüger. The BPR strategy is enabled when the plant experiences low-load conditions, which is usually at night. During the low-load periods, the BPR control signal (BPR_{focus}) is activated and prolonged denitrification periods can be imposed, where anaerobic conditions are obtained to promote BPR. During the anoxic zone, the phosphorus concentration may rise to a critical value, which enables the control signal BPR_{safe}, which enforces aeration in the reactors; establishing aerobic conditions to decrease the phosphorus concentration in the reactor.

The SCADA system (Hubgrade^{TM}) controls all biological an chemical processes of the wastewater treatment plant using different *modules*. Henceforth, the only module of interest is the phosphorus module, and a block diagram of the phosphorus control system with a feed-back controller is shown in Fig. 2. Metal salts are added to the process at two locations (indicated by ①, ②); ① injects to the mixed liquor of wastewater and sludge at the inlet to the biological reactors, and ② injects to the effluent from the biological treatment process (see Fig. 2). The notation used to denote the flow of chemical precipitant to the biological treatment and the settling process is $Q_{Me,bio}$ and $Q_{Me,set}$, respectively.

3.2. Data

Data is acquired through the existing SCADA system, where the sample period varies from 1 to 5 minutes. Thus, all data logged with a sampling period longer than 1 minute are up-sampled in order to preserve the signals dynamics.

Several control signals and sensor measurements are logged, these signals can be categorized in four different types as stated below:

- **Measurements (M)** are sensor measurements of the processes.
- **Control signals (C)** are signals generated by the current phosphorus control module. Those signals are not included when modeling the process, as it would lead to an invalid model, should the control strategy be changed or improved in the future. They are however described to present a thorough system description.
- **Watchdogs (A)** are alarm signals activated when certain signals exceed predefined limits. There are watchdogs monitoring the phosphorus concentration, inlet flow and set-point generation. The activation of a watchdog typically entail a different process mode (see below) to start.
- **Process Modes (P)** are Boolean signals or integer code depicting the process mode in action. There are three process modes of interest to this work; chemical phosphorus removal, biological phosphorus removal and forced aeration when the phosphorus concentration is too high.

The current control scheme is shown in Fig. 2, and available data signals are listed in Table 1.

The input vector consists of 22 data signals, including all crucial watchdogs, process modes and several online measurements. As with many wastewater treatment processes, the phosphorus dynamics are not well understood and known to be difficult to model due to its nonlinearities (Newhart et al., 2019). The phosphorus concentration is affected by the wastewater characteristics, operation conditions and environmental aspects - which all are constantly changing. This is evident from Fig. 3 and will be discussed later in the following sections.

Among the essential control signals are BPR_{focus} and BPR_{safe} along with a phase code, ϕ, describing the conditions governing each biological tank, watchdogs, WD, and the control signal depicting whether or not the CPR strategy is utilized, CPR_{CTRL}.

All biological processes (nitrification, denitrification and BPR) are controlled using integer code to describe the conditions governing each biological tank. A four digit code is used in the online control system to describe the process for each of the biological lines shown in Fig. 1. These codes are referred to as biological phase codes, and they are presumed to have a major impact on

[1] www.kruger.dk/english

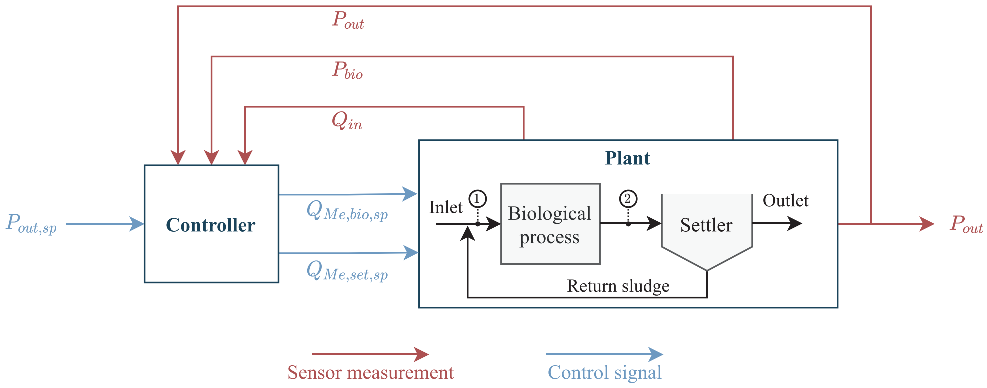


Fig. 2. Block diagram of the system.

Table 1
Notations used for the data signals. The type describes a measurement (M), control signals (C), alarm signals/watchdogs (A) or process modes/control signals (P).

Symbol	Type	Description	Unit
$P_{out,sp}$	C	Set-point for phosphorus in outlet	mg/L
$Q_{Me,bio,sp}$	C	Set-point for flow of chemical precipitant to biology	m^3/h
$Q_{Me,set,sp}$	C	Set-point for flow of chemical precipitant to settler	m^3/h
P_{out}	M	Phosphorus concentration in outlet	mg/L
P_{bio}	M	Phosphorus concentration in process tank	mg/L
Q_{in}	M	Inflow of raw WW and return sludge	m^3/h
$Q_{Me,bio}$	M	Flow for chemical precipitant to biology	m^3/h
$Q_{Me,set}$	M	Flow for chemical precipitant to settler	m^3/h
T	M	Temperature. Can have the subscript $_{bio}$ or $_{set}$	$^{\circ}C$
DO	M	Dissolved oxygen	mg/L
pH	M	pH value of wastewater in biological tank	–
SS	M	Suspended solids in biological tank	kg/m^3
Q_{WS}	M	Flow of waste sludge	m^3/h
Q_{RS}	M	Flow of return sludge	m^3/h
SS_{WS}	M	Suspended solids in waste sludge	kg/m^3
SS_{RS}	M	Suspended solids in return sludge	kg/m^3
BPR_{focus}	P	Enables when anoxic zones are created to enhance BPR	–
BPR_{safe}	P	Enables when forced aeration is used in BPR	–
ϕ	P	Phase code	–
$CTRL$	P	Enables when chemical phosphorus removal is used	–
WD	A	Watchdog	–

Table 2
Biological phase codes definition.

ϕ	Description
ϕ_{in}	The tank that wastewater flows into. Takes a value of 1 or 2
ϕ_{out}	The tank that effluent flows from. Takes a value of 1 or 2
ϕ_1	Conditions in tank 1. Takes a value of 0, 1 or 2.
ϕ_2	Conditions in tank 2. Takes a value of 0, 1 or 2.

the phosphorus concentration in the reactors, especially when BPR is enabled. The meaning of each digit in the phase codes are described in Table 2.

As noted in Table 2, digit ϕ_1 and ϕ_2 can take values between 0–2. They define the conditions in tank 1 and 2, respectively. The three possible options are:

0: Anaerobic conditions without aeration and mixing.
1: Anoxic conditions (denitrification) without aeration and with mixing.
2: Aerobic conditions (nitrification) with aeration and mixing.

An example could be: The wastewater in biology line 1 flows to tank 1 and the effluent out of tank 2. In tank 1 there is nitrification and in tank 2 there is denitrification. The phase code, ϕ, for biology line 1 will accordingly be:

$$\begin{bmatrix}\phi_{in} & \phi_{out} & \phi_1 & \phi_2\end{bmatrix} = \begin{bmatrix}1 & 2 & 2 & 1\end{bmatrix}.$$

Monthly statistical compositions of the measurements (both inputs and the output) are presented in Fig. 3 using boxplots, and a time series plot is shown for the output. Investigating the boxplots of inputs (Fig. 3(a)-(g)), it is evident that the biochemical processes consist of cyclostationary and strongly varying sub-processes. Furthermore, the statistical results illustrated in Fig. 3 show there is a large amount of outliers in the data set. This is due to the nature of the current control algorithm and the ASP, where the phosphorus concentration is driven towards zero while variables such as dissolved oxygen (DO) and inflow of chemical precipitant is controlled in a way where they often take a value of zero.

The process under consideration cannot be actively excited, however, the training data set still has to be designed by selecting a data set from the gathered signals that is as representative as possible. With this aim, the entire data set is divided into three subsets of data; *training data, validation data* and *test data*, and the partitioning is 80%, 10%, 10%, respectively. The data sets are designed such that a full year of data is utilized for model development (training), which should ensure the most representative data set and that information of the cyclostationary processes are incorporated in the model.

During training of the network, training data is used to optimize and update weights and biases each iteration while the validation data is used to select the best parameters in order to avoid overfitting the model to the training data set. Hence, the validation data set is not used directly for parameter optimization, but

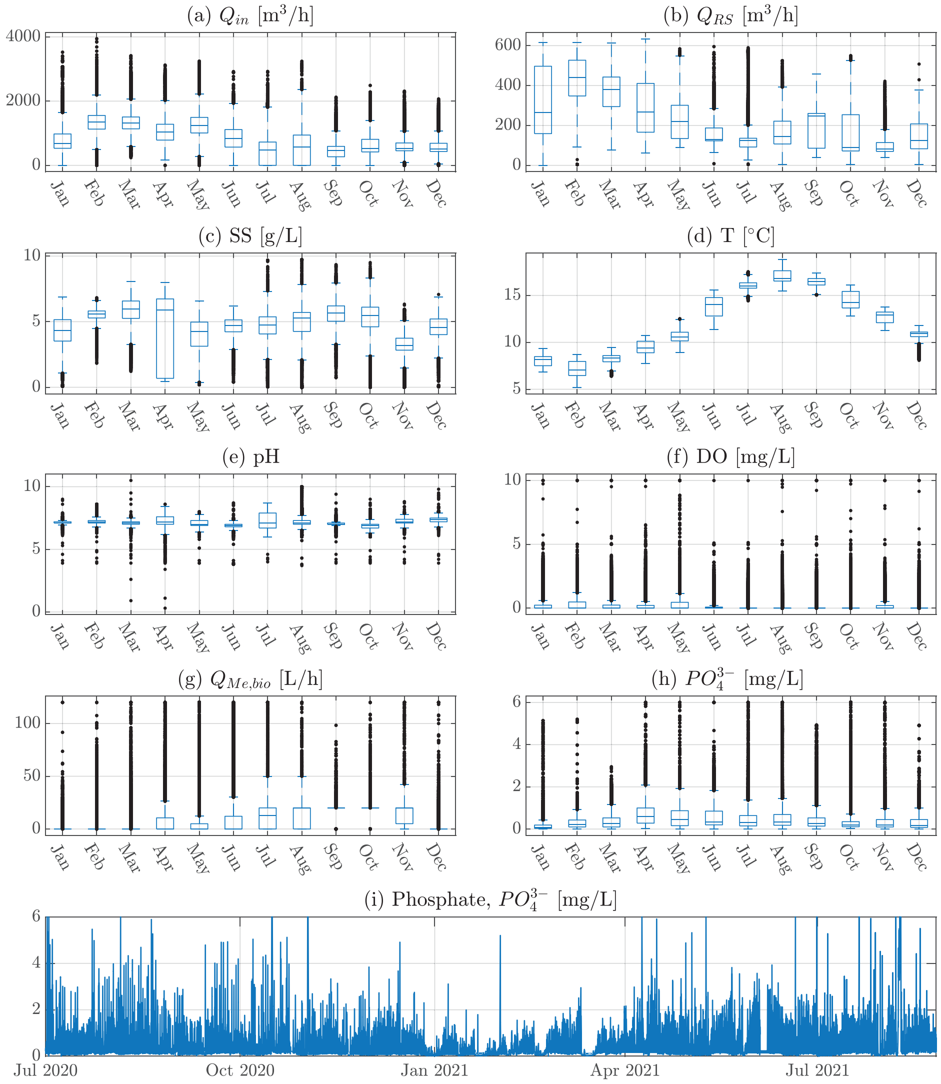


Fig. 3. Graphical descriptions of sensor data from July 2020 to August 2021; (a) - (g): boxplots of inputs, (h): boxplot of the output, and (i): time series plot of the output.

solely to achieve generalizability of the model. The test data set is excluded from the training process and used to evaluate the model performance on new data. If the recovered models shows great performance on new data which is previously unseen, the model is generalizable.

3.3. Software and hardware

The software used in this work is MATLAB R2020b, where the *Deep Learning Toolbox* (MathWorks, a) is used to train and validate the neural network while the *Statistics and Machine Learning Toolbox* (MathWorks, b) is used for the Bayesian Optimization of the hyperparameters. The framework of the machine learning and statistics toolbox (MathWorks, b) is used to perform the Bayesian optimization on the LSTM model using the function *bayesopt*. Computations are performed on a device with *Intel Core i7-6820HQ Quad Core 2.70GHz, 3.60GHz Turbo, 8MB 45W*, with a graphics chip *Nvidia Quadro M3000M w/4GB GDDR5 dedicated memory*.

4. Theory and methods

Two dynamic models with different purpose and applicability are proposed in this work; one to use as an estimator and another to predict the phosphorus concentration using previous sensor measurements. This section describes the structure of the LSTM model, a special type of RNN with gated units that incorporates long term memory to the model, which is used to model the multivariate system. Finally, the Bayesian optimization used for hyperparameter tuning of the LSTM is described.

4.1. Dynamic model

LSTMs are designed to learn long-term dependencies and are widely used for tasks such as speech recognition, natural language processing and other pattern recognition applications. Over the last decade, the LSTM has also been applied for system identification (Wang, 2017; Hwangbo et al., 2021; Lanzetti et al., 2019), where it has shown to outperform more established methods like the vanilla RNN and MLP. The mathematical formulation of a LSTM cell at time t with input $\mathbf{x}_t$ and hidden states $\mathbf{h}_t$ is given in equation (1).

$$\begin{aligned}
\mathbf{f}_t &= \sigma\left(\mathbf{W}_f \mathbf{h}_{t-1} + \mathbf{U}_f \mathbf{x}_t + b_f\right)\\
\mathbf{i}_t &= \sigma\left(\mathbf{W}_i \mathbf{h}_{t-1} + \mathbf{U}_i \mathbf{x}_t + b_i\right)\\
\mathbf{g}_t &= \tanh(\mathbf{W}_g \mathbf{h}_{t-1} + \mathbf{U}_g \mathbf{x}_t + b_g)\\
\mathbf{c}_t &= \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \mathbf{g}_t\\
\mathbf{o}_t &= \sigma\left(\mathbf{W}_h \mathbf{h}_{t-1} + \mathbf{U}_o \mathbf{x}_t + b_o\right)\\
\mathbf{h}_t &= \mathbf{o}_t \circ \tanh \mathbf{c}_t
\end{aligned} \tag{1}$$

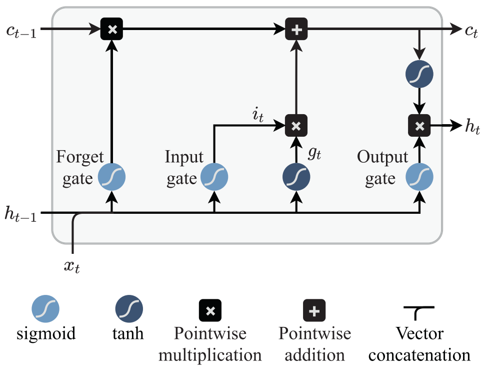


Fig. 4. The LSTM cell architecture.

The matrices $\mathbf{W}_*$ and $\mathbf{U}_*$ contain the weights of the input and recurrent connections where the subscript $_*$ can either be the forget gate (f), input gate (i), output gate (o) or cell update (g). The bias is described by b_*, the cell state by c_t and the operator $\circ$ denotes the element-wise multiplication of two vectors while σ is a sigmoid function. Fig. 4 presents the inner architecture of an LSTM cell.

The two proposed models differ in structure as *Model 1* utilizes the input vector given in Eq. (2) while *Model 2* uses the input vector given in Eq. (3).

$$\mathbf{x}_t = \left[\mathbf{u}_t\right] \tag{2}$$

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{u}_t \\ y_{t-d} \end{bmatrix} \tag{3}$$

In Eqs. (2) and (3), $\mathbf{u_t}$ denotes a vector of the 22 data signals described in Section 3.2, while y_{t-d} denotes the d'th past process output, i.e. the d'th past phosphorus measurement.

In all simplicity, Model 2 is an augmentation of Model 1, where previous sensor measurements of the process output is used as an input, hence making the Model 2 more sensitive to measurement noise, and leaving a demand for a sensor.

In control literature, d is commonly chosen to be 1 to simulate a discrete dynamic model. However, the phosphorus measurements are sampled with 1 minute intervals even though the sensor only provide new measurements every 4–5 minutes. As a result, the signal is up-sampled using zero-order hold, meaning that up to 5 consecutive data points are identical. This issue expresses that $\mathbf{y}_t = \mathbf{y}_{t-1}$ most of the time, meaning that the risk of training a persistence model using $d = 1$ is very high. To work around this issue and ensure that $\mathbf{y}_t \neq \mathbf{y}_{t-d}$, we use $d = 5$ in this work. The reader is referred to Section 5 for a further discussion on the issue of developing a persistence model and the advantages and limitations to the two specific models.

Similar to all other neural networks, the LSTM network can be constructed as deep neural networks, meaning that several LSTM layers can be stacked and connected internally. This yields two structure-related tuning parameters for the LSTM:

- Number of hidden layers
- Number of units in each layer

A model's capacity describes the ability to fit a wide variety of functions, and the number of layers and number of hidden units in a layer are two hyperparameters that affects both the training process and the model capacity. However, a high model capacity is not necessarily preferred, as high capacity may overfit and memorize properties of the training data. Contrary, a model with low capacity may struggle to fit to the training data. In this work, the number of hidden layers and number of hidden units in each layer are included in a Bayesian optimization algorithm along with several other hyperparameters. The aim is to select optimum hyperparameters of the LSTM models, and build the models with a capacity that matches the complexity of the task.

4.2. Hyperparameter tuning

As is the case with most deep learning algorithms, the LSTM network has several hyperparameters that affects many aspects of the algorithm's behavior. Some hyperparameters affect the need for computational power, time or memory cost of running the algorithm. Other hyperparameters affect the performance of the recovered model and its ability to predict accurate results when deployed on new data. Usually, neural networks have between 10–50 hyperparameters, depending on how the model is parameterized and how many parameters the developer has chosen to fix at a default value (Bergstra et al., 2011). In any case, the task of tuning the hyperparameters can be time-consuming and difficult, if done manually, as it requires understanding of how each hyperparameter affects the algorithms and how neural network models achieve good generalization (Goodfellow et al., 2016). Hence, there is great appeal for automatic tuning of hyperparameters, which in many cases is preferred if the computational power is available.

Automatic hyperparameter selection reduces the need for expert knowledge and rules of thumb, but they often require much more computational power (Goodfellow et al., 2016). Henceforth the term *hyperparameter optimization* is used to describe the automatic tuning of hyperparameters for the LSTM models. Several methods exists for automatic tuning of hyperparameters; grid search (Bergstra et al., 2011), random search (Bergstra and Bengio, 2012) and model based optimization such as Bayesian optimization (Snoek et al., 2012) are some of the commonly used methods. Common for all the automatic hyperparameter tuning strategies is that they wrap an optimization algorithm around the problem, hence hides the hyperparameters for the developer. Bayesian optimization is applied in this work to identify the best hyperparameters of an LSTM network, as it has shown great performance in research studies (Victoria and Maragatham, 2021; Snoek et al., 2012). Bayesian optimization is especially advantageous for problems where function evaluation is expensive (high duration), not easily differentiable or expressive.

There are many hyperparameters which can be included to optimize the training and performance of the recovered model. A thorough evaluation of different hyperparameters of the LSTM is presented in (Reimers and Gurevych, 2017) where several LSTM networks for sequence labeling tasks are presented. Contrary to many other studies where the focus lies on identifying the specific configuration that performs best on the given task, the study by Reimers and Gurevych focus on finding design choices that perform robustly. This means that the results are not task specific, but yields good performance when the architecture is applied to new tasks or new domains. In this study, we benefit from the study of hyperparameters (Reimers and Gurevych, 2017), such that we can reduce the number of hyperparameters to be tuned, and focus on five important parameters:

- Learning rate
- Number of layers
- Number of hidden units
- Weight decay coefficient (L_2 regularization)
- Minibatch size

One of the hyperparameters set to default values are the solver, where *adam* has shown to produce stable and better results than other solvers (Reimers and Gurevych, 2017). Similarly, dropout regularization is implemented as it has show superior to no-dropout

(Reimers and Gurevych, 2017), and is implementable in the software used in this work.

Bayesian optimization attempts to minimize an objective function, $f(x)$ in a bounded domain for x. For the task of regression, mean squared error (MSE) is often chosen as the objective function, and is also used in this work. The three key elements to Bayesian optimization:

1. A Gaussian process (GP) model of the objective function. It is assumed that the objective function values are drawn from a Gaussian process so that the observations $\{\mathbf{x}_n, y_n\}_{n=1}^{N}$ are of the form:

   $$y_n \sim \mathcal{N}(f(\mathbf{x}_n), \nu) \tag{4}$$

   In which $\mathbf{x}$ contains the hyperparameters to be optimized and ν is the variance of the noise introduced into the observations.
2. A Bayesian update procedure at each new evaluation of the objective function.
3. An acquisition function, $a(\mathbf{x})$, that determines the next point $\mathbf{x}_{next}$ to be evaluated as $\mathbf{x}_{next} = \text{argmax}_{\mathbf{x}\in\mathcal{X}} a(\mathbf{x})$.

The covariance function is a crucial element of the GP prior, as it expresses the similarity between two function values; $f(x_i)$ and $f(x_j)$. The covariance function (also known as *kernel function*) can be expressed as $k(x_i, x_j|\theta)$, where θ is a vector of kernel parameters. Together, the kernel function and acquisition function constitute two important choices for the Bayesian optimization algorithm. *Expected-improvement-per-second-plus* is utilized as acquisition function, as serves as the default option in MATLAB and, furthermore, show great performance for machine learning applications (Snoek et al., 2012). The *automatic relevance determination (ARD) Matérn 5/2* kernel function is chosen for this work, as it has shown to outperform other (MATLAB built-in) kernel functions such as: *squared exponential, ARD squared exponential*, and *ARD Matern 3/2* (Snoek et al., 2012).

With all choices established, the algorithm is then executed as follows:

1. Evaluate $y_i = f(\mathbf{x}_i)$ for a random point $\mathbf{x}_i$ taken within the hyperparameter bounds.
2. Update the Gaussian process model of $f(\mathbf{x})$ to obtain a posterior over functions.
3. Find the new point that maximizes the acquisition function, $a(\mathbf{x})$

Step 2 and 3 are repeated until a specified stopping criterion, such as fixed number of iterations or a fixed time is reached. The objective function is defined to take the following steps for each iteration, k;

1. Take the values of the hyperparameters as inputs. The *bayesopt* function calls the objective function with the current values, $\mathbf{x}_k$ of the hyperparameters.
2. Define the network with the $\mathbf{x}_k$ hyperparameters.
3. Train and validate the network.
4. Save the validation error and the used hyperparameters $\mathbf{x}_k$.
5. Return the validation error.

5. Results and discussion

Firstly, Bayesian optimization is applied to determine the optimum hyperparameters for the two models. Secondly, the LSTM models are trained on the training data set and evaluated on a test data set, starting with Model 1, which functions as an estimator and can be used for process predictions - continuing to Model 2, which can only be used for process predictions. Lastly, the performance of the two models is compared and evaluated while advantages and disadvantages of both models are discussed.

Table 3
Hyperparameters that are included in the optimization algorithm and the corresponding search bounds.

Hyperparameter	Optimization range
L_2 regularization coef.	$[1 \cdot 10^{-10} \quad 0.01]$
Learning rate	[0.01 1]
Mini-batch size	[[4320 28800]]
Hidden Layers	[1 5]
Hidden units in each hidden layer	[50 400]

Table 4
Optimum hyperparameter values determined using Bayesian optimization.

Hyperparameter	Model 1	Model 2
L_2 regularization coefficient	$1.34 \cdot 10^{-10}$	$1.14 \cdot 10^{-9}$
Learning rate	0.0101	0.0162
Mini-batch size	8066	5167
Hidden Layers	4	2
Hidden units	82	92

5.1. Bayesian optimization

Bayesian hyperparameter tuning is applied to obtain an optimum LSTM structure with a capacity that matches the complexity of the task. The learning rate, weight decay, mini-batch size and depth and width of the network are included in the optimization algorithm, and the appertaining search ranges, where the optimum values are to be found within, are given in Table 3. Default values are defined for the solver and the dropout value; Adam (Kingma and Lei Ba, 2017) is chosen as the solver and the dropout probability is set to 0.5.

The optimization range for the mini-batch size is chosen so that the data batches are containing between 3 and 20 days of data.

To best utilize the power of Bayesian optimization, at least 30 objective function evaluations should be performed (MathWorks, b). The minimum objective of each iteration is tracked and shown in Fig. 5. For both models, we see that the local minimum of the cost function; the optimum hyperparameters; were found no later than iteration 15. However, from Fig. 5a, it is evident that the search space is much more unpredictable compared to Fig. 5b, since the estimated minimum objective oscillates and rarely predicts correctly. This indicates that for the optimization of Model 1, the choice of hyperparameter values impact the performance of the model less than predicted.

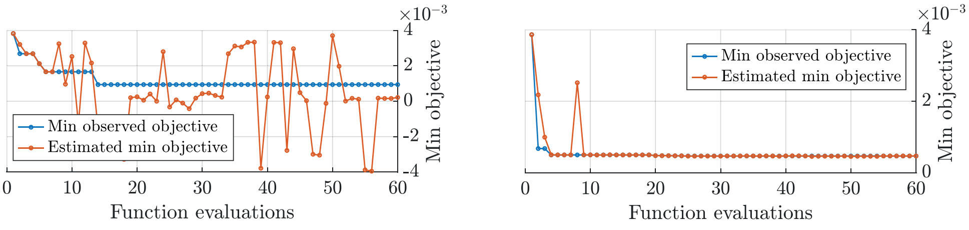
A subset of approximately 4 days is used to perform the Bayesian optimization. The minimum objective was achieved with the hyperparameter values listed in Table 4.

5.2. Model dynamics

Two models are proposed in this work, with a different desired purpose and available input information, as shown in equations (2) and (3). Model 1 was trained and tested with the available plant data. The benefit of this model is that it does not include the phosphorus measurement as an input, meaning that this model functions as an estimator.

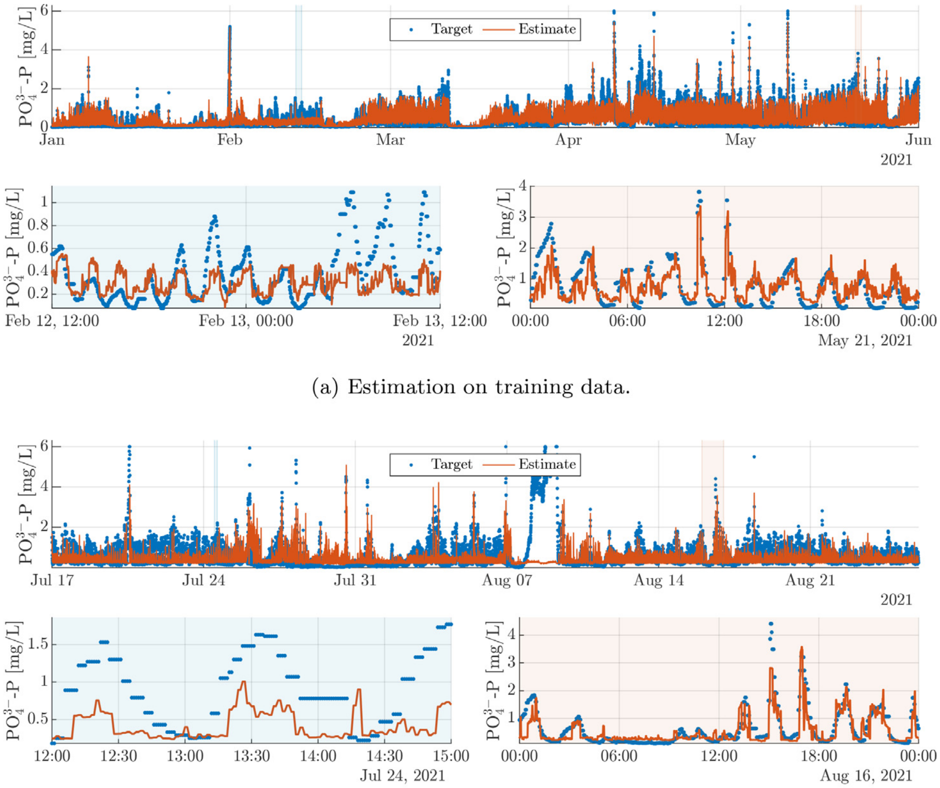
Model 2 was trained and tested on the same data as Model 1, and the performance of the two models is evaluated and compared. The second model is presumed to perform better than the first, but with the disadvantage of relying on true phosphorus sensor measurements as input to the model.

The model quality is typically measured as a function of the error between the (disturbed) process output and the model output (Nelles, 2020). In this work, the optimum LSTM network structures determined using Bayesian optimization will be evaluated based on the following performance measures:



(a) (b)

Fig. 5. Bayesian optimization of hyperparameters in the LSTM; (a) for Model 1 and (b) for Model 2.



(a) Estimation on training data.

(b) Estimation on test data.

Fig. 6. Training and test performance of Model 1. This figure shows (a): a section of the training data set and (b):the entire test data set.

- Mean squared error (MSE)
- Goodness of fit (GoF) given by normalized root mean squared error (NRMSE): GoF = 1 - NRMSE)
- Coefficient of determination (R^2)
- Time series plots
- Cross-correlograms

5.2.1. Model 1: Without P as input

The first model is trained and tested on the data set, and a section of the training data is shown in Fig. 6a. Fig. 6a(top) shows a time series plot of 5 month to visualize the varying system dynamics over longer periods, and how the model in some periods fails to estimate high phosphorus concentrations in the outer range of the data spectrum. In Fig. 6a(top), two periods are marked with blue and red background color, respectively, and these periods are shown in Fig. 6a(bottom left and right) with corresponding background. The two zoomed plots illustrate how the model captures the dynamics of the system. In the period February 13, 00:00 - 12:00, the data shows a significantly higher magnitude of phosphorus concentration compared to the model estimate, indicating unmodeled dynamics.

The model is evaluated on the test data set which is previously unseen to the model, and results are shown in Fig. 6(b). Similarly to the evaluation on the training data set, the model is considerably fit to estimate low concentrations, but lacks precision when the concentration is high.

Note that there are two periods in the test data set, which differs significantly from the rest of the data set. This is the 1-day period from July 29th to July 30th and also the 2-day period from August 7th to august 9th. In those periods the phosphorus concentration is very low (≤ 0.1 [mg/L]) and very high (≥ 3.3 [mg/L]), respectively. The explanation of this is that heavy rainfall was occurring thus activating a watchdog called *storm water mode*. During the two periods, the plant has lead wastewater in and out through tank 2, only performing mechanical treatment of the wastewater (and hence no biological treatment). The phase code for both periods was constant at:

$$\begin{bmatrix}\phi_{in} & \phi_{out} & \phi_1 & \phi_2\end{bmatrix} = \begin{bmatrix}2 & 2 & 0 & 0\end{bmatrix}.$$

Table 5
Statistics for the estimation performance on test data.

	MSE	GoF	R^2
Model 1	0.22	0.10	0.19
Model 2	0.037	0.37	0.87

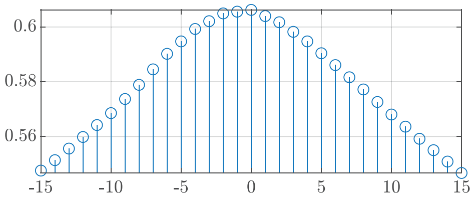


Fig. 7. Cross-correlation between target value and estimate for Model 1.

The two periods represent rare cases of +24 hours with critical conditions (heavy rainfall), where the nutrient removal strategy is given a lower priority. Hence, this kind of system behavior is not represented in the training data, and it can therefore not be described by the model. To obtain a well performing data-driven model, all possible process behavior must be represented within the training data set.

Investigating Fig. 6, it is clear that the process is oscillating with higher amplitudes than the model estimate, which indicates that the model does not receive the necessary information in the inputs to estimate the oscillating behavior properly. The information which triggers the concentration is clearly available for the model, as the model estimate overall follows the dynamic behavior of the process. Since all measurable properties of the wastewater composition are already included for the estimation, it is presumed to lack information about the wastewater fauna which is not yet available through sensor measurements - supporting the claim that the process is complicated and difficult to model.

Several statistics are given in Table 5 to use as performance measures. The table show performance measures only for the test data (excluding the two rare cases with heavy rainfall described above), and serves as a tool for comparing model estimation for both Model 1 and Model 2.

The cross-correlation between two signals provides a measure of similarity, and can be used to evaluate the performance of the model and detect if the model lags or leads the target value. A peak at lag 0 imply that the model estimate follows the dynamic behavior without any lead or lag, and a normalized maximum cross-correlation of 1 indicates proportionality between the model estimate and the target values.

A cross-correlogram of the target value and estimate using Model 1 is shown in Fig. 7. As the dynamics of the process are rather slow compared to the sampling interval of the signals (1 minute), the cross-correlation is naturally high around lag 0. Nevertheless, the maximum cross-correlation of the two signals is found at lags 0 where it takes the normalized value of 0.61, indicating that the model neither leads nor lags the actual process dynamics.

5.2.2. Model 2: With P as input

To properly compare the two network structures, Model 2 is trained and evaluated similarly to Model 1. Fig. 8a(top) shows a time series plot of 5 months to visualize the varying system dynamics over longer periods, and the zoomed plots in the bottom left and right corner shows the same periods as depicted in Fig. 6. Contrary to Model 1, Model 2 performs well on both low and high concentrations of phosphorus. Investigating the test data performance in Fig. 8(b) supports the immediate indication that Model 2 outperforms Model 1.

Table 5 shows the statistics for the estimation on test data. The performance measures clearly show an increased model performance when comparing Model 2 to Model 1. The MSE is reduced while the GoF and R^2 have increased.

When training a neural network where a previous system output (y_{t-d}) is used as an input to the model, it can result in a persistence model, where the estimate, $\hat{y}_t$ is simply given as y_{t-d}. Examining Fig. 8b(bottom left), we see that the model estimate seem to lag the target signal.

Plotting the cross-correlation between the target and estimated value in Fig. 9, we see a maximum cross-correlation value close to 1 at time lag -5.

The combination of high sampling frequency, low sensor resolution of the phosphorus measurement and slow system dynamics all contribute to the overall high cross-correlation around lag 0 in Fig. 9. However, the maximum cross-correlation occurring at time lag -5 indicates that the model uses the previous value as estimate for the next value, or at least depend heavily upon it. The validity of this statement is tested by utilizing the model for future predictions of phosphorus concentration where the sensor measurement is not available. Assuming that all future inputs except the phosphorus concentration are known in the prediction period, Model 2 is then applied to perform k-step ahead predictions by substituting y_{t-d} in equation (3) with the previous model prediction, $\hat{y}_{t-d}$.

The input vector consists of roughly two types of inputs; (1) actuators, which are altered by the SCADA system to control the process, and (2) known disturbances, which are conditions that cannot be controlled by the SCADA system (pH, temperature, etc.) but are measured to monitor the process. The assumption that all future inputs are known in the prediction time period is acceptable for the following reasons:

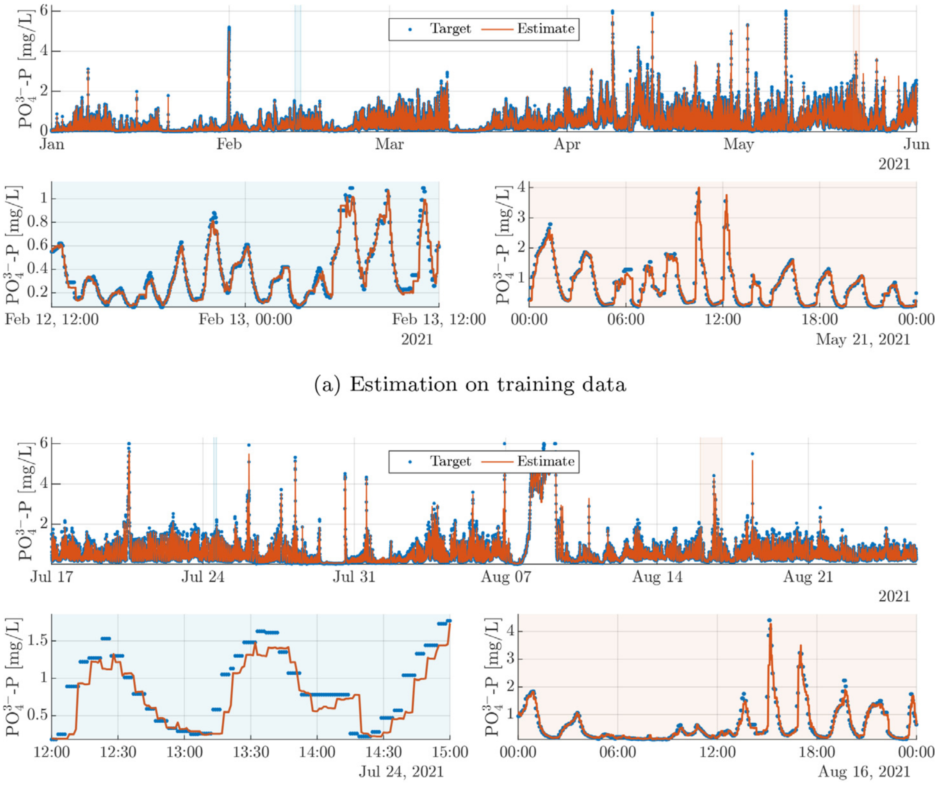
- Actuators are naturally known and can be controlled freely within practical limitations.
- Known disturbances can either be presumed constant for a short time period, or presumed predicted using supplementary/additional dynamic models.

For this work, the known disturbances are presumed predicted using dynamic models with a 100% accuracy. We leave the task of developing those models to future work, and assume ”perfect” models.

Predictive performance of the model is evaluated with data from four periods of the data set, chosen arbitrarily among periods of normal operation without any (known) problems. In Fig. 10, the prediction performance of the two models are compared by utilizing the models to predict 720 and 1440 time steps ahead corresponding to 12 and 24 hours, respectively. Statistical measures for the four periods shown in Fig. 10 are given in Table 6. During two of the periods (shown in Figs. 10(b) and 10(d)), the storm-water-mode is partly activated. Those periods are included to evaluate the model performance when the water flow has increased due to rainfall.

In Fig. 10, the first two hours show estimation of the phosphorus concentration, where Model 2 uses true PO_4^{3-}-P measurements as inputs. The stippled vertical line indicates the transition from estimation to prediction, where in prediction, $\hat{y}_{t-d}$ is used to predict the current concentration, y_t.

Figure 10 (a) shows a period where the two models both struggle to estimate the true amplitude of the phosphorus concentration in the first 4 hours of the timeline. However, when the measured concentration decreases to around 1 mg/L, the two models predict the concentration with high accuracy; hence, indicating that periods of high phosphorus concentration are not well represented in the training data set. This statement is, furthermore, supported by the nature of the current control module, which aims to reduce the phosphorus concentration.



(a) Estimation on training data

(b) Estimation on test data.

Fig. 8. Training and test performance of Model 2. This figure shows (a): a section of the training data set and (b):the entire test data set.

Table 6
Statistics for the four periods in Fig. 10. For convenience, the best performing model for each period with respect to the given metric is marked with bold font.

	MSE		GoF		R^2	
	Model 1	Model 2	Model 1	Model 2	Model 1	Model 2
Period (a)	**0.044**	0.045	**0.27**	0.26	0.46	0.46
Period (b)	0.30	**0.24**	0.33	**0.40**	0.55	**0.64**
Period (c)	**0.18**	0.31	**0.12**	−0.16	**0.22**	−0.35
Period (d)	0.15	**0.094**	0.42	**0.54**	0.66	**0.79**

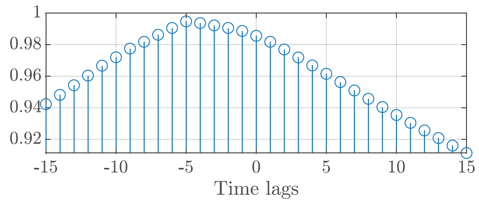


Fig. 9. Cross-correlation between target value and estimate for Model 2.

The period shown in Fig. 10(b) appears to be well represented in the training data, as the two models show high prediction performance for this region. The only exception is at the time period 06:00-08:00, where both models fail to predict the high concentration. In the same period, Model 2 predicts slower dynamics than Model 1 and the real process, hence indicating that Model 2 does not truly capture the effect of the chemical precipitant, and depends heavily on the previous phosphorus measurement.

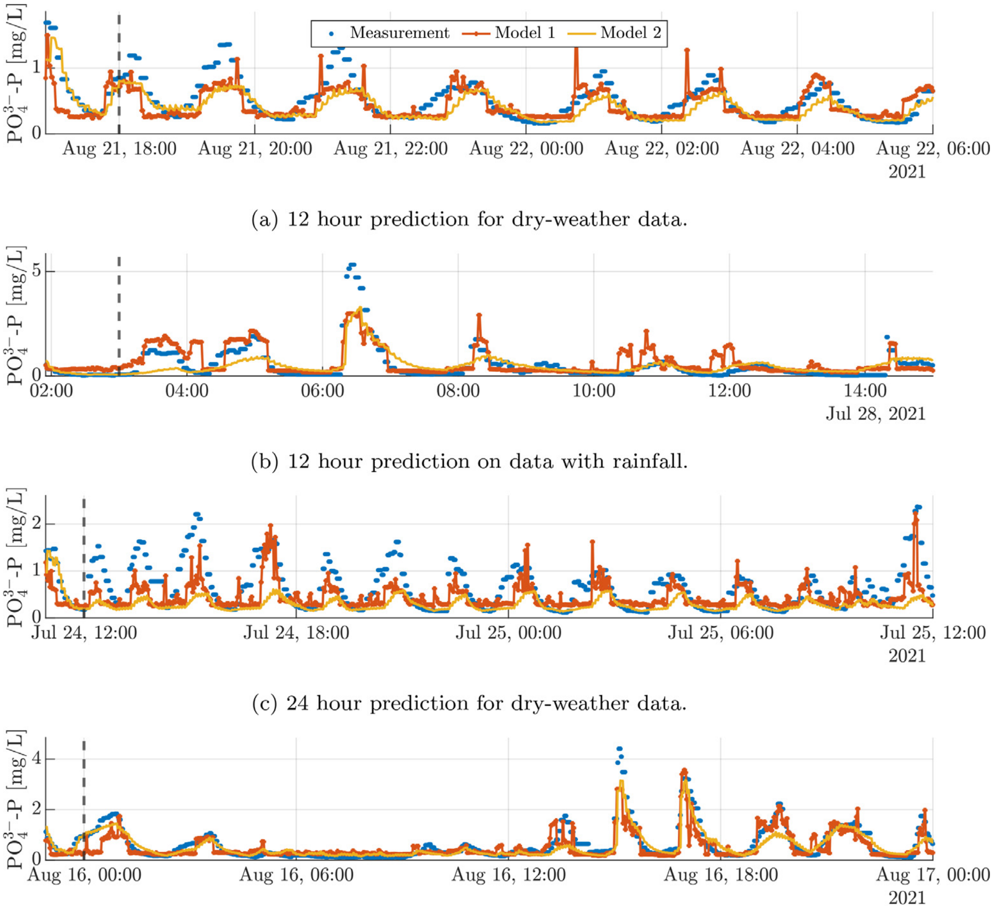
In Fig. 10(c), both models show poor prediction performance (the worst of the four periods shown in Fig. 10). The figure indicates that the two models are triggered by the same conditions, as we observe concentration increase and decrease simultaneously. Contrary to the evaluation of period (a), Model 1 seem to estimate the amplitude better than Model 2 in period (c).

A 24 h period where both models show great performance is presented in Fig. 10(d). Similar to period (b), we see high performance metrics in Table 6. However, as observed in period (b), the predictions of Model 2 show slower dynamics compared to the observed dynamics and estimates using Model 1.

From Fig. 10 we see that the model performances depends mainly on the true system behavior in the period it is used to predict, thus is less affected by the duration of the prediction horizon or even the model structure. Table 6 supports this statement, as there is no model which is clearly superior to the other over the four different regions.

The regions presented in Fig. 10(a) and Fig. 10(c) contains dynamic behavior which is unmodeled, possibly caused by an unusually large load of phosphorus in the influent from industry or due to the majority of the data set representing low phosphors concentration as a result of the control module. Regardless, this indicates that the training data is not representative for the test data set.

The two models presented in this work perform similarly on the test data set; however, a major advantage of Model 1 is that



(a) 12 hour prediction for dry-weather data.

(b) 12 hour prediction on data with rainfall.

(c) 24 hour prediction for dry-weather data.

(d) 24 hour prediction on data with rainfall.

Fig. 10. Predicting phosphorus concentration 12 hours (720 time steps) and 24 hours (1440 time steps) into the future using Model 2. The stippled line indicates the transition from estimation to prediction using Model 2.

the phosphorus sensor is not needed to initiate the model. Hence, Model 1 has the highest prospects of usage seen from a plant operation perspective, as this model can be used to estimate phosphorus concentrations in tanks, where there are no sensor installed yet. As a result, Model 1 can decrease the expense related to phosphorus modeling and control and provide a method to monitor all ASP tanks in the plant.

A whole year of data is used for training in this work, thus indicating that a single cycle of data is not adequate to model a cyclostationary system while assuring consistently good model performance. It is assumed that at least 2–3 years of training data is required to achieve high model performance as obtained in Figs. 10(b) and 10(d).

6. Conclusion

Two models are proposed in this work; one which estimates the phosphorus concentration solely by use of data describing environmental conditions and process operation, and one which utilizes past phosphorus measurements along with information about environmental conditions and process operation. The LSTM hyperparameters are optimized using Bayesian optimization, yielding two different model structures. The resulting LSTM models are suitable for estimation and predictions of phosphorus concentrations up to 24 hours into the future with $R^2 = 0.79$ for dynamics well represented in the training data set.

Using the same set of test data, the two models are evaluated based on performance metrics such as MSE, GoF and R^2. Model 1, which serves as an estimator, captures the dynamics of the system but struggles to estimate the amplitude of the system. For the task of estimation, Model 1 is outperformed by Model 2, which utilizes past phosphorus measurements in the model. Results show that the two models have similar predictive performance, where the two models both show high performance for certain periods. Some periods with low model performance can be related to the dynamics not being represented in the training data set - suggesting a simple but necessary improvement strategy for the task of WWTP modeling: include more data in the training procedure. Utilizing 2–3 years of data is expected to solve this problem.

Although we present two equally applicable dynamic models, the first choice for modeling the process seen from a plant operative point of view, is Model 1. This model has the benefit of not including phosphorus measurements to predict the future phosphorus concentrations. Hence, it is not affected by sensor noise or bad measurement quality, and expenses concerning the phosphorus sensor can be saved.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Laura Debel Hansen: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Mikkel Stokholm-Bjerregaard:** Investigation, Resources, Writing – review & editing. **Petar Durdevic:** Conceptualization, Methodology, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgment

The authors thank Krüger-Veolia Collegues for their technical and supervisory support. This project was supported by the Danish Innovation Foundation, grant number: 1044-00031B.

References

Aguado, D., Ribes, J., Montoya, T., Ferrer, J., Seco, A., 2009. A methodology for sequencing batch reactor identification with artificial neural networks: a case study. Comput. Chem. Eng. 33 (2), 465–472. doi:10.1016/j.compchemeng.2008.10.018.

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for hyper-parameter optimization. Adv Neural Inf Process Syst 24.

Bergstra, J., Bengio, Y., 2012. Random search for hyper-parameter optimization. Journal of machine learning research 13 (2).

Besharati Fard, M., Mirbagheri, S.A., Pendashteh, A., Alavi, J., 2020. Estimation of effluent parameters of slaughterhouse wastewater treatment with artificial neural network and B-spline quasi interpolation. International Journal of Environmental Research 14 (5), 527–539. doi:10.1007/s41742-020-00274-1.

Blue Kolding, 2021. Blue Kolding Home Page. Accessed: 2021-08-24 https://bluekolding.dk/.

Bongards, M., 1999. Controlling Biological Wastewater Treatment Plants Using Fuzzy Control and Neural Networks. In: Reusch, B. (Ed.), Computational Intelligence. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 142–150.

Bunce, J. T., Ndam, E., Ofiteru, I. D., Moore, A., Graham, D. W., 2018. A review of phosphorus removal technologies and their applicability to small-scale domestic wastewater treatment systems. 10.3389/fenvs.2018.00008

Calderon, G., Draye, J.P., Pavisic, D., Teran, R., Libert, G., 1996. Nonlinear dynamic system identification with dynamic recurrent neural networks. Proceedings of International Workshop on Neural Networks for Identification, Control, Robotics, and Signal/Image Processing, NICROSP 49–54. doi:10.1109/nicrsp.1996.542744.

Cao, W., Yang, Q., 2018. Prediction Based on Online Extreme Learning Machine in WWTP Application. In: Cheng, L., Leung, A.C.S., Ozawa, S. (Eds.), Neural Information Processing. Springer International Publishing, Cham, pp. 184–195.

Côté, M., Grandjean, B.P., Lessard, P., Thibault, J., 1995. Dynamic modelling of the activated sludge process: improving prediction using neural networks. Water Res. 29, 995–1004. doi:10.1016/0043-1354(95)93250-W.

Dias, A.M.A., Ferreira, E.C., 2009. Computational Intelligence Techniques for Supervision and Diagnosis of Biological Wastewater Treatment Systems. In: do Carmo Nicoletti, M., Jain, L.C. (Eds.), Computational Intelligence Techniques for Bioprocess Modelling, Supervision and Control. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 127–162. doi:10.1007/978-3-642-01888-6_5.

Dürrenmatt, D., Gujer, W., 2012. Data-driven modeling approaches to support wastewater treatment plant operation. Environ. Modell. Software 30, 47–56. doi:10.1016/j.envsoft.2011.11.007.

Feldman, H., Flores-Alsina, X., Ramin, P., Kjellberg, K., Jeppsson, U., Batstone, D.J., Gernaey, K.V., 2017. Modelling an industrial anaerobic granular reactor using a multi-scale approach. Water Res. 126, 488–500. doi:10.1016/j.watres.2017.09.033.

Flores-Alsina, X., Ramin, E., Ikumi, D., Harding, T., Batstone, D., Brouckaert, C., Sotemann, S., Gernaey, K.V., 2021. Assessment of sludge management strategies in wastewater treatment systems using a plant-wide approach. Water Res. 190, 116714. doi:10.1016/j.watres.2020.116714.

Garikiparthy, P.S.N., Lee, S.C., Liu, H., Kolluri, S.S., Esfahani, I.J., Yoo, C.K., 2016. Evaluation of multiloop chemical dosage control strategies for total phosphorus removal of enhanced biological nutrient removal process. Korean J. Chem. Eng. 33 (1), 14–24. doi:10.1007/s11814-015-0132-9.

Gaya, M.S., Wahab, N.A., Sam, Y.M., Samsuddin, S.I., 2013. Feed-Forward Neural Network Approximation Applied to Activated Sludge System. In: Tan, G., Yeo, G.K., Turner, S.J., Teo, Y.M. (Eds.), AsiaSim 2013. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 587–598.

Gernaey, K.V., Jeppsson, U., 2014. Benchmarking of control strategies for wastewater treatment plants. IWA publishing.

Gernaey, K.V., Van Loosdrecht, M.C.M., Henze, M., Lind, M., Jørgensen, S.B., 2004. Activated sludge wastewater treatment plant modelling and simulation: state of the art. Environ. Modell. Software 19 (9), 763–783.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT Press.

Han, H., Zhu, S., Qiao, J., Guo, M., 2018. Data-driven intelligent monitoring system for key variables in wastewater treatment process. Chin. J. Chem. Eng. 26 (10), 2093–2101. doi:10.1016/j.cjche.2018.03.027.

Hansen, L.D., Veng, M., Durdevic, P., 2021. Compressor scheduling and pressure control for an alternating aeration activated sludge process–A simulation study validated on plant data. Water (Basel) 13 (8), 1037. doi:10.3390/w13081037.

Henze, M., Gujer, W., Mino, T., van Loosdrecht, M., 2000. Activated Sludge Models ASM1, ASM2, ASM2d and ASM3. Technical Report.

Henze, M., Harremoës, P., Jansen, J., Arvin, E., 2002. Wastewater treatment: Biological and chemical processes, 3rd Springer.

Hochreiter, S., Schmidhuber, J., 1997. Long short-Term memory. Neural Comput 9 (8), 1735–1780. doi:10.1162/neco.1997.9.8.1735.

Husin, M.H., Rahmat, M.F., Wahab, N.A., Sabri, M.F.M., 2021. Neural Network Ammonia-Based Aeration Control for Activated Sludge Process Wastewater Treatment Plant. In: Md Zain, Z., Ahmad, H., Pebrianti, D., Mustafa, M., Abdullah, N.R.H., Samad, R., Mat Noh, M. (Eds.), Proceedings of the 11th National Technical Seminar on Unmanned System Technology 2019. Springer Singapore, Singapore, pp. 471–487.

Hwangbo, S., Al, R., Chen, X., Sin, G., 2021. Integrated model for understanding N2O emissions from wastewater treatment plants: A Deep learning approach. Environ. Sci. Technol. 55 (3), 2143–2151. doi:10.1021/acs.est.0c05231.

Hwangbo, S., Al, R., Sin, G., 2020. An integrated framework for plant data-driven process modeling using deep-learning with monte-Carlo simulations. Comput. Chem. Eng. 143. doi:10.1016/j.compchemeng.2020.107071.

Ingildsen, P., Rosen, C., Gernaey, K.V., Nielsen, M.K., Gulldal, T., Jacobsen, B.N., 2006. Modelling and control strategy testing of biological and chemical phosphorus removal at avedøre WWTP. Water Sci. Technol. 53 (4–5), 105–113. doi:10.2166/wst.2006.115.

Kazadi Mbamba, C., Flores-Alsina, X., John Batstone, D., Tait, S., 2016. Validation of a plant-wide phosphorus modelling approach with minerals precipitation in a full-scale WWTP. Water Res. 100, 169–183. doi:10.1016/j.watres.2016.05.003.

Kazadi Mbamba, C., Lindblom, E., Flores-Alsina, X., Tait, S., Anderson, S., Saagi, R., Batstone, D.J., Gernaey, K.V., Jeppsson, U., 2019. Plant-wide model-based analysis of iron dosage strategies for chemical phosphorus removal in wastewater treatment systems. Water Res. 155, 12–25. doi:10.1016/j.watres.2019.01.048.

Keskitalo, J., Leiviskä, K., 2015. Artificial Neural Network Ensembles in Hybrid Modelling of Activated Sludge Plant. In: Angelov, P., Atanassov, K., Doukovska, L., Hadjiski, M., Jotsov, V., Kacprzyk, J., Kasabov, N., Sotirov, S., Szmidt, E., Zadrożny, S. (Eds.), Intelligent Systems'2014. Springer International Publishing, Cham, pp. 683–694.

Kingma, D.P., Lei Ba, J., 2017. Adam: a method for stochastic optimization. arXiv preprint.

Krüger, 2021. Hubgrade Performance Plant. Accessed: 2021-10-15 https://www.kruger.dk/english/hubgrade-advanced-online-control.

Lanzetti, N., Lian, Y.Z., Cortinovis, A., Dominguez, L., Mercangoz, M., Jones, C., 2019. Recurrent neural network based MPC for process industries. 2019 18th European Control Conference, ECC 2019 1005–1010. doi:10.23919/ECC.2019.8795809.

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444. doi:10.1038/nature14539.

MathWorks, a. Deep Learning Toolbox - MATLAB. Accessed: 2021-05-28 mathworks.com/products/deep-learning.

MathWorks, b. Statistics and Machine Learning Toolbox - MATLAB. Accessed: 2021-05-28, mathworks.com/products/statistics.

Meng, X., Zhang, Y., Qiao, J., 2021. An adaptive task-oriented RBF network for key water quality parameters prediction in wastewater treatment process. Neural Computing and Applications 0123456789 (3). doi:10.1007/s00521-020-05659-z.

Narendra, K.S., Parthasarathy, K., 1990. Identification and control of dynamical systems using neural networks. IEEE Trans. Neural Networks 1 (1), 4–27. doi:10.1109/72.80202.

Nelles, O., 2020. Nonlinear system identification: From classical approaches to neural networks, fuzzy models, and Gaussian processes, 2nd Springer International Publishing doi:10.1007/978-3-030-47439-3.

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

Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural networks. In: International Conference on Machine Learning. PMLR, pp. 1310–1318.

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

Pisa, I., Vilanova, R., Santín, I., Vicario, J.L., Morell, A., 2019. Artificial Neural Networks Application to Support Plant Operation in the Wastewater Industry. In: Camarinha-Matos, L.M., Almeida, R., Oliveira, J. (Eds.), Technological Innovation for Industry and Service Systems. Springer International Publishing, Cham, pp. 257–265.

Reimers, N., Gurevych, I., 2017. Optimal hyperparameters for deep LSTM-networks for sequence labeling tasks. https://arxiv.org/abs/1707.09861.

Sinha, N.K., Gupta, M.M., Rao, D.H., 2000. Dynamic neural networks: an overview. Proceedings of the IEEE International Conference on Industrial Technology 1, 491–496. doi:10.1109/icit.2000.854201.

Smagulova, K., James, A.P., 2019. A survey on LSTM memristive neural network architectures and applications. Eur. Phys. J. Special Topics 228, 2313–2324. doi:10.1140/epjst/e2019-900046-x.

Snoek, J., Larochelle, H., Adams, R.P., 2012. Practical bayesian optimization of machine learning algorithms. Adv Neural Inf Process Syst 25.

Solomatine, D., See, L., Abrahart, R., 2008. Data-Driven Modelling: Concepts, Approaches and Experiences. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 17–30.

Solon, K., Flores-Alsina, X., Kazadi Mbamba, C., Ikumi, D., Volcke, E.I., Vaneeckhaute, C., Ekama, G., Vanrolleghem, P.A., Batstone, D.J., Gernaey, K.V., Jeppsson, U., 2017. Plant-wide modelling of phosphorus transformations in wastewater treatment systems: impacts of control and operational strategies. Water Res. 113, 97–110. doi:10.1016/j.watres.2017.02.007.

Stentoft, P.A., Munk-Nielsen, T., Vezzaro, L., Madsen, H., Møller, J.K., Mikkelsen, P.S., 2019. Towards model predictive control: online predictions of ammonium and nitrate removal by using a stochastic ASM. Water Sci. Technol. 79 (1), 51–62. doi:10.2166/wst.2018.527.

Victoria, A.H., Maragatham, G., 2021. Automatic tuning of hyperparameters using bayesian optimization. Evolving Systems 12 (1), 217–223. doi:10.1007/s12530-020-09345-2.

Vrecko, D., Gernaey, K.V., Rosén, C., Jeppsson, U., 2006. Benchmark simulation model No 2 in matlab-Simulink: towards plant-wide WWTP control strategy evaluation. Water Sci. Technol. 54 (8), 65–72.

Wang, Y., 2017. A new concept using LSTM Neural Networks for dynamic system identification. In: Proceedings of the American Control Conference. Institute of Electrical and Electronics Engineers Inc., pp. 5324–5329. doi:10.23919/ACC.2017.7963782.

Wilfert, P., Kumar, P.S., Korving, L., Witkamp, G.J., Van Loosdrecht, M.C., 2015. The relevance of phosphorus and iron chemistry to the recovery of phosphorus from wastewater: A Review. Environ. Sci. Technol. 49 (16), 9400–9414. doi:10.1021/acs.est.5b00150.

Wunsch, A., Liesch, T., Broda, S., 2018. Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). J Hydrol (Amst) 567, 743–758. doi:10.1016/j.jhydrol.2018.01.045.

Zhao, L., Dai, T., Qiao, Z., Sun, P., Hao, J., Yang, Y., 2020. Application of artificial intelligence to wastewater treatment: a bibliometric analysis and systematic review of technology, economy, management, and wastewater reuse. Process Safety and Environmental Protection 133 (92), 169–182. doi:10.1016/j.psep.2019.11.014.