Prediction of Key Variables in Wastewater Treatment Plants Using Machine Learning Models

Rodrigo Salles*, Jérôme Mendes*, Rui Araújo*, Carlos Melo*, and Pedro Moura*
* *University of Coimbra, Institute of Systems and Robotics,*
Department of Electrical and Computer Engineering, Pólo II, PT-3030-290 Coimbra, Portugal
Email: rodrigo.salles@isr.uc.pt, jermendes@isr.uc.pt, rui@isr.uc.pt, carlos.melo@isr.uc.pt, pmoura@isr.uc.pt

Abstract—**Prediction of key variables is an important part of the monitoring, control, and optimization of industrial processes, since it is important to anticipate certain behaviors so that the correct actions can be taken. To assess which algorithm is best suited to the prediction of a number of key variables at various stages of wastewater treatment plants (WWTP), five computational algorithms were researched: Artificial Neural Network, Long Short-Term Memory, deep learning Transformer model, Adaptive Neuro-Fuzzy Inference System, and Gaussian Mixture Model. With these models, techniques already well established in the state-of-the-art are evaluated, as well as more recent methods that have been exhibiting good performance in variable prediction regression problems. These algorithms were evaluated in four WWTP case studies, in which the objective is to predict the following key variables: total suspended solids, nitrate and nitrite, ammonia and ammonium, and biochemical oxygen demand. The learning process of each algorithm was performed using extensive tests in order to select the input variables, and define the topologies and hyper-parameters of the presented models by cross-validation. The results indicate that it is possible to adequately predict the four variables, and the best results were achieved by the Transformer algorithm, which presents the lower error values in the considered metrics.**

Index Terms—**Wastewater treatment plant, LSTM, ANFIS, Transformer, Gaussian Mixture Model, key variable prediction.**

1. Introduction

Wastewater treatment plants (WWTPs) are structures of great importance to society, the environment, and on the circular economy and energy sustainability domains [1]. Its role is to remove polluting agents from the waters that result from human activities, and which could pose risks to the environment. In the United States of America there are about 16,000 public administration WWTPs [2], while the European Union has more than 18,000 WWTPs, serving a population of over 450 million people in its member states [3]. WWTPs are complex structures, with non-linear characteristics, which carry out the water treatment in several complementary steps, in order to accelerate the process that would occur in nature. Due to their characteristics, most WWTPs are inefficient, and use more resources than necessary [4]. In [5], it was found that the size of the plant, the type of aeration of the bioreactors and the amount of organic matter varies significantly, helping to explain the different levels of energy efficiency. [6] showed that the eco-efficiency of WWTPs is affected by low efficiency related to energy consumption, global warming effects and economic integration.

The WWTPs optimization techniques often involve predicting variables related to their water treatment steps, being very important to know the behavior of key variables in the next few minutes or hours, to implement control, optimization, and failure prediction techniques. For example, it is possible to anticipate the energy demand, chemical reagents, or the quality of the water leaving the station. In [7], the authors propose the use of a Long Short-Term Memory (LSTM) network to identify collective faults in WWTP applications. The LSTM is compared with autoregressive integrated moving average (ARIMA), principal component analysis (PCA), and support vector machine (SVM) models, presenting a superior performance, reaching a fault detection rate (recall) of over 92% for failures in the sensors responsible for measuring ammonia levels. Models derived from LSTM and gated recurrent unit (GRU) are used as soft-sensors in [8] to predict key variables in WWTPs. [8] have used real data to validate the models, in which, LSTM obtained better results. In [9] an artificial neural network (ANN) is used to forecast the effluent stream, in terms of biological oxygen demand (BOD), chemical oxygen demand (COD), and total suspended solids (TSS). In [10], a random forest (RF) is proposed for the prediction of the daily wastewater inflow in a real plant. The RF model uses regression trees to capture the nonlinear relationship between wastewater inflow and various influencing variables. Comparisons were made with multilayer perceptron neural networks (MLP) and ARIMA, and the results show that the RF model performs well in predicting inflow. To predict the energy consumption values, an ANN, a Gradient Boosting Machine (GBM) with two decision trees, and RF were used, being the best prediction performance achieved

This research was supported by the ERDF and national funds through the project SYNAPPS (CENTRO-01-0247-FEDER-046978), and by Fundação para a Ciência e a Tecnologia (FCT) under the project UIDB/00048/2020.

with GBM. Ensemble methods have also been studied. In [11] four methods were analyzed and their results were combined to make predictions. Feed forward neural network (FFNN), adaptive neuro fuzzy inference system (ANFIS), support vector regression (SVR), and a classical multi-linear regression (MLR) method were applied for predicting the performance of effluent biological oxygen demand (BOD_{eff}), chemical oxygen demand (COD_{eff}), and total nitrogen (TN_{eff}) in a WWTP. The results showed that in the prediction of BOD, the ensemble models of simple averaging ensemble, weighted averaging ensemble, and neural network ensemble, increased the prediction performance up to 14%, 20% and 24%, respectively. In all previously mentioned works, it is possible to verify the use of several different techniques, with different architectures, in order to predict key variables on the WWTPs.

With the aim of efficiently predicting the key variables on WWTPs, algorithms must capture the patterns present in the time series that represent the various stages of treatment, and the same algorithm may behave differently in different series. Therefore, an individual evaluation and performance comparison is necessary to determine which is the best model for each stage of treatment. That is the purpose of this paper, where ANN, LSTM, deep learning Transformer model, ANFIS, and Gaussian Mixture Model (GMM) algorithms are evaluated in the prediction of four key variables: total suspended solids, nitrate and nitrite, ammonia and ammonium, and biochemical oxygen demand. Methods already established in the state-of-the-art, such as ANN, ANFIS and GMM, performed well, but more recent methods, such as Transformer and LSTM, which take into account the context of the information, have presented superior performance, with great potential for predicting variables, which represents an important contribution for the prediction of variables in these complex WWTPs.

The remainder of the paper is structured as follows. Section 2 presents the WWTP model and the case studies. In Section 3, the machine learning models used to predict the key variables are described. Section 4 presents the experimental results. Finally, the conclusions are presented in Section 5.

2. Benchmark Simulation Model No 2: Case Study

WWTPs are structures subject to many variations, presenting highly non-linear characteristics [12], and which nevertheless need to function within the strict limits imposed by environmental legislation [13]. Benchmark Simulation Model 2 (BSM2) [14] is a simulation environment, in which are defined the plant layout, the simulation model, influence loads, test procedures and evaluation criteria, where for each of these items, compromises were pursued to combine plainness with realism and accepted standards. BSM2, presented in Figure 1, describes a structure responsible for the treatment of wastewater, consisting of biological and sludge treatment, being mainly composed of primary settling, activated sludge system (including anoxic and aerobic reactors, and a secondary clarifier), anaerobic digester, sludge thickener and dewatering, and a sludge storage unit.

During the various stages of the treatment process, many variables can be assessed. The main ones, and the most important to the prediction case-studies investigated in this paper, are: inert soluble material, S_I [gCOD/m^3], readily biodegradable substrate, S_S [gCOD/m^3], inert particulate material, X_I [gCOD/m^3], slowly biodegradable substrate, XS [gCOD/m^3], heterotrophic biomass, $X_{B,H}$ [gCOD/m^3], autotrophic biomass, $X_{B,A}$ [gCOD/m^3], inert particulate material from biomass decay, X_P [gCOD/m^3], dissolved oxygen, S_O [gCOD/m^3], nitrate and nitrite, S_{NO} [gN/m^3], ammonia and ammonium, S_{NH} [gN/m^3], soluble organic nitrogen associated with SS, S_{ND} [gN/m^3], particulate organic nitrogen associated with XS, X_{ND} [gN/m^3], alkalinity, S_ALK, total suspended solids, TSS [gSS/m^3] , flow rate [m$^3/d$], and temperature [^{o}C].

The variables analyzed, from simulations of BSM2, which compose the dataset used in the present work, were collected, with a sampling frequency of 15 minutes, along the so-called water line (primary, secondary and tertiary treatment), over 10 points, labeled from **P1** to **P10**, as presented in Figure 1, which goes from the entrance of the primary settler to the exit of the station, passing through the activated sludge tank. Univariate and bivariate statistical analyzes were carried out, and the variables with the highest correlations to the variables to be predicted were selected. Sections 2.1–2.4 briefly overview the case studies, i.e. the target variables to be predicted.

2.1. Total Suspended Solids

Total suspended solids (TSS) are considered one of the major pollutants that contribute to the deterioration of water quality, contributing to higher costs of water treatment, decreases in fish resources, and the general aesthetics of the water [15]. TSS is an important parameter, because an excess of TSS depletes the dissolved oxygen (DO) in the effluent water. The target is to predict the TSS value at the output of the primary clarifier, labeled in Figure 1 as **P3**, TSS^*, (see Figure 1) by using the input data of this clarifier, labeled in the same figure as point **P2**. To predict $TSS^*(k+1)$ the following input variables were selected: $TSS(k)$, $S_I(k)$, $SNH(k)$, $S_{ND}(k)$, and $X_{ND}(k)$.

2.2. Ammonia and Ammonium

Human activities at municipal, industrial, agricultural and domestic levels generate many nitrogenous compounds, with ammonia being one of these compounds. The excessive accumulation of ammonium that is discharged into the water can cause serious ecological problems, such as: the accelerated eutrophication of lakes and rivers, the depletion of dissolved oxygen, and toxicity in fish and other aquatic animals in the water body [15]. To avoid it, the nitrogen compounds are removed in the biological reactor. The target is to predict the NH_4 values at the output of the aerobic

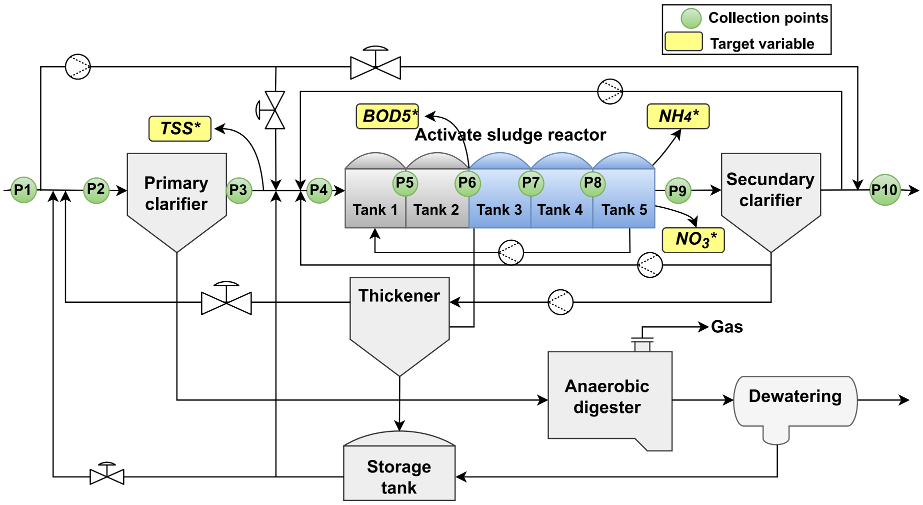


Figure 1: Layout of BSM2.

tank, NH_4^* (see Figure 1). Therefore, the data were collected between tanks 2 and 3, and at the exit of tank 5, at points **P6** and **P9**, respectively. The input variables chosen to compose the models to predict $NH_4^*(k+1)$ where: $S_S(k)$, $X_S(k)$, $S_O(k)$, $S_NO(k)$, $S_{NH}(k)$, $S_{ND}(k)$, $X_{ND}(k)$, $S_{alk}(k)$, and $TSS(k)$.

2.3. Nitrate and Nitrite

Nutrients present in water benefit plant growth, but an excess of these components can cause water eutrophication. Together with phosphorus, nitrates in excess amounts can accelerate eutrophication. Aquatic ecosystems are harmed by the excess of nitrogenous substances, which seriously affect aquatic fauna and flora. Therefore, it is necessary to reduce the wastewater nitrogen pollution loads. Nitrogen substances in wastewater are traditionally removed by nitrification and denitrification. When nitrites and nitrates are used by heterotrophic bacteria as an oxidant of organic matter, the removal of nitrogen as a gas takes place [15]. The target is to predict the value of nitrate and nitrite at the output of the aerobic tank, NO_3^*, by using data collected between tanks 2 and 3, and at the output of tank 5, at points **P6** and **P9**, respectively. The input variables chosen to predict $NO_3^*(k+1)$ are: $X_S(k)$, $S_{NO}(k)$, $S_{NH}(k)$, $S_{ND}(k)$, $X_{ND}(k)$, and $S_{alk}(k)$.

2.4. Biological Oxygen Demand (BOD5)

The BOD5 indicates the amount of oxygen that bacteria and other micro-organisms consume in a water sample during the period of five days at a temperature of 20 °C to degrade the water contents aerobically [15]. BOD5 is thus an indirect measure of the sum of all biodegradable organic substances in the water. The BOD5 indicates how much dissolved oxygen is needed in a given time for the biological degradation of the organic wastewater constituents, and is given by (1).

$$BOD_5 = 0.25(S_S + X_S + (1 - f_P)(X_{B,H} + X_{B,A})), \quad (1)$$

where $f_P = 0.08$ is a stoichiometric parameter. The objective is to predict the $BOD5^*$ value at the input of the aerobic tank, labeled in Figure 1 as point **P6**, by using data collected between tanks 2 and 3. The input variables chosen to predict $BOD5^*(k+1)$ are: $X_{B,H}(k)$, $X_p(k)$, $S_O(k)$, and $S_{N,D}(k)$.

3. Machine Learning Models for Prediction of Key Variables in WWTPs

In order to predict the key variables identified in Section 2, TSS, NH_3, S_{NO}, and BOD_5, for the WWTP, this section presents a brief description of the following methods that were applied: Artificial neural networks (ANN), Long short-term memory (LSTM), Adaptive neuro-fuzzy inference system (ANFIS), Transformers, and Gaussian Mixture Models (GMM).

3.1. Artificial Neural Networks

ANN is a massively parallel combination of simple and interconnected processing units which can acquire knowledge from the environment through a learning process and store the knowledge in its connections [16]. In the next definitions, the notation $n(\cdot)$ refers to the dimension of some layer, as example, $n(l)$ denotes the dimension of the layer l.

The mathematical description of an ANN with L layers can be formulated by equations (2) to (4).

$$\begin{aligned}
\mathbf{x}_l &= [\mathbf{y}_{l-1}^T, 1]^T, \; 1 \le l \le L, && (2)\\
\mathbf{v}_l &= \mathbf{W}_l \mathbf{x}_l + b_l, \; 1 \le l \le L, && (3)\\
\mathbf{y}_l &= \boldsymbol{\psi}_l(\mathbf{v}_l), \; 1 \le l \le L, && (4)
\end{aligned}$$

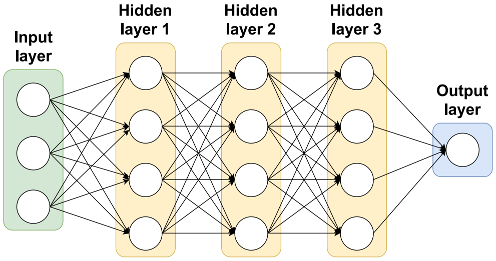


Figure 2: General deep neural network model.

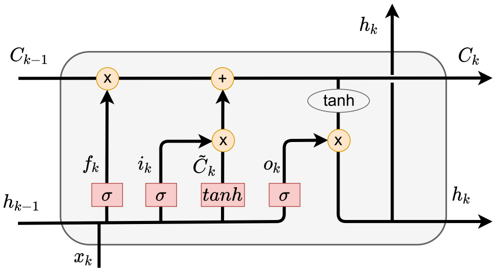


Figure 3: Generic structure of the LSTM cell.

where $\mathbf{y}_0 = \mathbf{x}$, $\mathbf{x} = [x_1, \ldots, x_n]^T$ is the input vector containing n input variables, $\mathbf{x}_l = [x_{l,1}, \ldots, x_{l,n_{(l-1)}+1}]^T$ is input vector of layer l, $\mathbf{v}_l = [v_{l,1}, \ldots, v_{l,n(l)}]^T$ is the synaptic operation output vector of layer l, $\mathbf{y}_l = [y_{l,1}, \ldots, y_{l,n(l)}]^T$ is the output vector of layer l, $\mathbf{W}_l = [W_{l,1}, \ldots, W_{l,n(l)}]$ and b_l are the synaptic weight coefficients matrix and the bias of the layer l, respectively; $\boldsymbol{\psi}_l(\mathbf{v}_l) = [\psi_{l,1}(\mathbf{v}_{l,1}), \ldots, \psi_{l,n(l)}(\mathbf{v}_{l,n(l)})]^T$ is the element-wise activation functions vector of layer l. All elements of matrices $\mathbf{W}_l$ are concatenated into one vector of parameters by (5), where $vec(\cdot)$ is a linear transformation which converts the matrix into a column vector [17].

$$\boldsymbol{\Theta} = [vec(\mathbf{W}_1)^T, \ldots, vec(\mathbf{W}_L)^T]^T \times 1. \tag{5}$$

For the training of the deep neural network, in this paper, the Backpropagation algorithm with the Mean Squared Error (MSE) cost function was used.

3.2. Long Short-Term Memory

LSTM is a recurrent neural network (RNN) that has shown good performances in time series prediction. Proposed in [18], LSTM networks make their predictions based on the historical context of occurrences, using memory cells. A generic structure of a LSTM is presented in Figure 3. In an LSTM cell there are input, forget, memory and output gates, that are used in order to decide which signals are going to be forwarded to another node. Considering $\mathbf{x}_k$ the cell input signal, the behavior of all gates in the LSTM cell is described by equations (6) to (11).

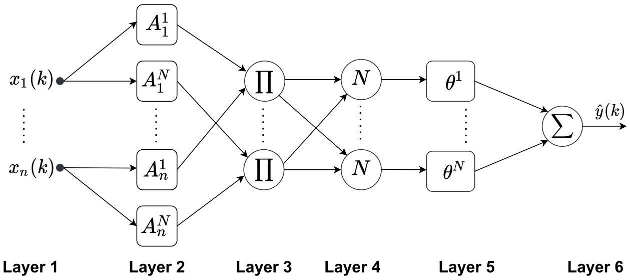


Figure 4: An ANFIS architecture.

$$\begin{aligned}
\mathbf{i}_k &= \sigma(\mathbf{x}_k\mathbf{U}^i + \mathbf{h}_{k-1}\mathbf{W}^i), && (6)\\
\mathbf{f}_k &= \sigma(\mathbf{x}_k\mathbf{U}^f + \mathbf{h}_{k-1}\mathbf{W}^f), && (7)\\
\mathbf{o}_k &= \sigma(\mathbf{x}_k\mathbf{U}^o + \mathbf{h}_{k-1}\mathbf{W}^o), && (8)\\
\tilde{\mathbf{C}}_k &= \tanh(\mathbf{x}_k\mathbf{U}^g + \mathbf{h}_{k-1}\mathbf{W}^g), && (9)\\
\mathbf{C}_k &= \sigma(\mathbf{f}_k * \mathbf{C}_{k-1} + \mathbf{i}_k * \tilde{\mathbf{C}}_k), && (10)\\
\mathbf{h}_k &= \tanh(\mathbf{C}_k) * \mathbf{o}_k, && (11)
\end{aligned}$$

where $\mathbf{W}^i$, $\mathbf{W}^f$, $\mathbf{W}^o$ and $\mathbf{W}^g$ are the recurrent connection between the previous hidden layer $l-1$ and current hidden layer l, $\mathbf{U}^i$, $\mathbf{U}^f$, $\mathbf{U}^o$ and $\mathbf{U}^g$ are the weight matrix that connects the inputs to the hidden layer. The indices i, f, o and g are related to the input, forget, output and memory gates, respectively. $\tilde{\mathbf{C}}$ is a candidate hidden state that is computed based on the current input and the previous hidden state, C is the internal memory of the unit, which is a combination of the previous memory, multiplied by the forget gate $\mathbf{f}_k$, and the newly computed hidden state, multiplied by the input gate. The update of the LSTM network weights was done through the Adam optimizer, and with the MSE loss function.

3.3. Adaptive Neuro-Fuzzy Inference System

ANFIS (Figure 4) is a combination of neural networks and fuzzy systems. The ANFIS architecture works with six layers, with a Takagi-Sugeno (T-S) fuzzy rules structure, which in this paper is a zero order T-S structure [19], defined by:

$$\begin{aligned}
R_i :\ &\text{IF } x_1(k) \text{ is } A_1^i, \text{ and} \ldots \text{and } x_n(k) \text{ is } A_n^i\\
&\text{THEN } y_i(k) = \theta^i,
\end{aligned}$$

where R_i $(i = 1, \ldots, N)$ represents the i-th fuzzy rule, $\mathbf{x}(k) = [x_1(k), \ldots, x_n(k)]^T$ is the vector of input variables, and A_j^i $(j = 1, \ldots, n)$ are linguistic terms characterized by fuzzy membership functions $\mu_{A_j^i}(k)$. ANFIS is composed of 6 layers [20]: Layer 1 presents the input variables, and its nodes represent linguistic variables; Layer 2 contains the membership functions A_j^i and acts as a unit of memory; Layer 3 computes a product of the membership functions, according to (12).

$$\bar{\mu}_i(k) = \prod_{j=1}^{n} \mu_{A_j^i}(k); \tag{12}$$

Layer 4 is responsible for the normalization of the activation values of the fuzzy rules, i.e. computing $\bar{\mu}_i(k)/\sum_{i=1}^{N}\bar{\mu}_i(k)$. Layer 5 presents the consequents of the fuzzy rules (adaptive nodes); and Layer 6 computes the overall output, i.e. sum of all inputs from the precedent layer. The ANFIS model using the rules structure (12) is given by (13).

$$y(k) \quad = \quad \sum_{i=1}^{N} \bar{\omega}^i[\mathbf{x}(k)]\boldsymbol{\theta}^i = \boldsymbol{\Theta}^T\mathbf{W}(k), \qquad (13)$$

where

$$\begin{aligned} \bar{\omega}^i[\mathbf{x}(k)] &= \frac{\bar{\mu}_i(k)}{\sum_{i=1}^{N}\bar{\mu}_i(k)}, & (14)\\ \boldsymbol{\Theta} &= \left[\theta^1,\ldots,\theta^N\right]^T, & (15)\\ \mathbf{W}(k) &= \left[\bar{\omega}^1[\mathbf{x}(k)],\ldots,\bar{\omega}^N[\mathbf{x}(k)]\right]^T. & (16)\end{aligned}$$

The learning algorithm uses a combination of the least-squares and backpropagation gradient descent methods.

3.4. Transformers

Transformer is a deep learning model that adopts a mechanism called self-attention in its predictions. Self-attention is a mechanism that relates different positions of elements in a sequence to calculate their representation. The Transformer's architecture is composed of stacked self-attention and fully connected layers for encoder and decoder purposes. The model adopted for the predictions of key variables is similar to the original proposed in [21].

According to Figure 5, Transformer consists of an encoder and decoder. The encoder is composed of an **Input layer**, a **Positional encoding** layer, and a stack of two identical **Encoder layers**. The input layer maps the time series $(x_1,\ldots,x_n)$ to a n-dimensional vector. A positional encoding layer with sine and cosine functions is used to encode sequential information in the time series data by element-wise addition of the input vector with a positional encoding vector. The resulting vector is fed into the encoder layers. Each encoder layer consists of four sub-layers: a **Self-attention** sub-layer and a fully-connected **Feedforward** sub-layer. Each sub-layer (Self-attention and Feed-forward) is followed by a normalization sub-layer. The encoder produces a n-dimensional vector $\mathbf{z} = [z_1,\ldots,z_n]$ to feed the decoder. The decoder is composed of a **Input layer**, **Decoder layers**, and **Linear mapping layers**. The input decoder receives the last value of the input encoder time series. The input layer maps the decoder input to a n-dimensional vector. In addition to the defined two sub-layers presented in each encoder layer, the decoder layers have a third sub-layer, the **Enconder-Decoder attention**, to apply self-attention mechanisms over the encoder output. Finally, there is a linear mapping layer that maps the output of the last decoder layer to the time series representing the forecast of the key variables of the case studies [22].

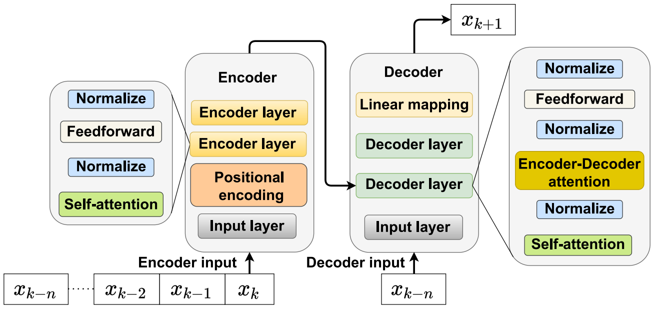


Figure 5: The Encoder-Decoder Structure of the Transformer.

3.5. Gaussian Mixture Models

A Gaussian Mixture Model (GMM) is an effective prediction technique that has origins in statistics and has been largely adopted by the machine learning community [23]. When it comes to modeling real datasets, a simple Gaussian distribution can be limited to capture the structure of the data, while a superposition of two or more Gaussians can do a better characterization of the dataset [24]. This superposition can be composed of probabilistic models which are parameterized by: a Gaussian component $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu},\boldsymbol{\Sigma})$ with means, $\boldsymbol{\mu} \equiv \{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K\}$, and covariances, $\boldsymbol{\Sigma} \equiv \{\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_K\}$, and the component weights $\boldsymbol{\pi} \equiv \{\pi^1,\ldots,\pi^K\}$, where K is the number of Gaussian components.

During the training phase, the joint probability distribution was learned $p(\mathbf{x}_k, y_k)$ through Expectation-Maximization (EM) technique [25]. In this phase, the data points are assigned to their Gaussians. The joint probability distribution is given by (17).

$$p(\mathbf{x}_k, y_k) = \sum_{i=1}^{K} \pi^i \mathcal{N}_i(\mathbf{x}_k, y_k|\boldsymbol{\mu}^i_{\mathbf{x}y}, \boldsymbol{\Sigma}^i_{\mathbf{x}y}), \qquad (17)$$

where i-th component $\mathcal{N}^i(\cdot)$ has a mean of $\boldsymbol{\mu}^i_{\mathbf{x}y}$, a variance of $\boldsymbol{\Sigma}^i_{\mathbf{x}y}$, and the component weight π^k with the constraint $\sum_{i=1}^{K}\pi^i = 1$.

During the prediction phase, it was used Gaussian Mixture for Regression (GMR) algorithm [25], that computes the conditional distribution $p(y|\mathbf{x})$ for each test observation. The conditional distribution is computed by (18).

$$p(y|\mathbf{x}) = \sum_{i=1}^{K} \pi^i_{y|\mathbf{x}} \mathcal{N}_i(y|\boldsymbol{\mu}^i_{y|\mathbf{x}}, \boldsymbol{\Sigma}^i_{y|\mathbf{x}}), \qquad (18)$$

with the component weight $\pi^i_{y|\mathbf{x}}$ computed by (19).

$$\pi^i_{y|\mathbf{x}} \frac{\mathcal{N}_i(\mathbf{x}_k|\boldsymbol{\mu}^i_{\mathbf{x}}, \boldsymbol{\Sigma}^i_{\mathbf{x}})}{\sum_{l=1}^{K}\mathcal{N}_l(\mathbf{x}_k|\boldsymbol{\mu}^l_{\mathbf{x}}, \boldsymbol{\Sigma}^l_{\mathbf{x}})}, \qquad (19)$$

where the component of each i-th Gaussian $\mathcal{N}_i(\mathbf{x}_k, y|\boldsymbol{\mu}_{\mathbf{x}y}, \boldsymbol{\Sigma}_{\mathbf{x}y})$ has means $\boldsymbol{\mu}_{\mathbf{x}y} = (\boldsymbol{\mu}_x, \mu_y)^T$ and covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}y}$, guiving by (20).

$$\boldsymbol{\Sigma}_{\mathbf{x}y} \begin{pmatrix} \boldsymbol{\Sigma}_{\mathbf{xx}} & \boldsymbol{\Sigma}_{\mathbf{x}y} \\ \boldsymbol{\Sigma}_{y\mathbf{x}} & \boldsymbol{\Sigma}_{yy} \end{pmatrix} \tag{20}$$

$\boldsymbol{\Sigma}_{\mathbf{xx}}$ is the variance of $\mathbf{x}$, $\boldsymbol{\Sigma}_{yy}$ the variance of y, $\boldsymbol{\Sigma}_{y\mathbf{x}}$ the covariance between y and $\mathbf{x}$, and $\boldsymbol{\Sigma}_{\mathbf{x}y}$ the covariance between $\mathbf{x}$ and y. Then, the means and covariance of y given x, (y|**x**), for each test observation, were computed by (21) and (22) [25].

$$\mu_{y|\mathbf{x}} = \mu_y + \boldsymbol{\Sigma}_{y\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}, \tag{21}$$

$$\boldsymbol{\Sigma}_{y|\mathbf{x}} = \boldsymbol{\Sigma}_{yy} - \boldsymbol{\Sigma}_{y\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{xx}}^{-1}\boldsymbol{\Sigma}_{\mathbf{x}y}. \tag{22}$$

4. Experimental Results

The algorithms presented in Section 3, ANN, LSTM, ANFIS, Transformer and GMM, were applied to the case studies introduced in Section 2, in which, the objective is to predict the variables TSS, NH_4, NO_3, and $BOD5$.

In order to obtain the best model for each tested algorithm, several tests were performed by the combination of the respective main parameters, and the best were those who obtained the lowest errors according to the metrics considered, namely, MSE (23), Root MSE (RMSE) (24), Mean Absolute Error (MAE) (25), and Mean Absolute Percentage Error (MAPE) (26).

$$\text{MSE} = \frac{1}{S}\sum_{k=1}^{S}(y_k - \hat{y}_k)^2, \tag{23}$$

$$\text{RMSE} = \sqrt{\sum_{k=1}^{S}\frac{(\hat{y}_k - y_k)^2}{S}}, \tag{24}$$

$$\text{MAE} = \frac{1}{S}\sum_{k=1}^{S}|y_k - \hat{y}_k|, \tag{25}$$

$$\text{MAPE} = \frac{1}{S}\sum_{k=1}^{S}\left|\frac{y_k - \hat{y}_k}{y_k}\right| 100\%, \tag{26}$$

where y_k and $\hat{y}_k$ are the real and estimated target values at the instant of time k, and S is the number of samples used to validation.

The best models were found through grid search. Considering the number of hidden layers (Nl), number of neurons per layer (Nn), number of training cycles ($Epochs$), number of batch size (Bs), the membership function type (MF) and the number of LSTM cells for the input layer (Nc_i) and for the hidden layers (Nc_h), the best models result in the combinatorial analysis of the following hyperparameters:

- ANN:
  - $Nl = [1, 2, 3, 4, 5]$,
  - $Nn = [100, 200, 300, 400, 500]$,
  - $Epochs = [10, 20, 30, 40, 50]$.
- LSTM:
  - $Nl = [1, 2, 3, 4, 5]$;
  - $Nc_i = [16, 32, 64, 128, 256]$;
  - $Nc_h = [16, 32, 64, 128, 256]$;
  - $Bs = [8, 16, 32, 64, 128]$;
  - $Epochs = [4, 8, 12, 16, 32]$.
- ANFIS:
  - MF = {triangular, trapezoidal, Gaussian, generalized bell-shaped};
  - $Epochs = [40, 60, 80, 100, 120]$.
- Transformer:
  - $Nn = [16, 32, 64, 128]$;
  - $Bs = [8, 16, 32, 64]$;
  - $Epochs = [10, 20, 40, 60]$.
- GMM:
  - $K = [2, 3, 4, 5, 10, 15, 20, 25, 30]$.

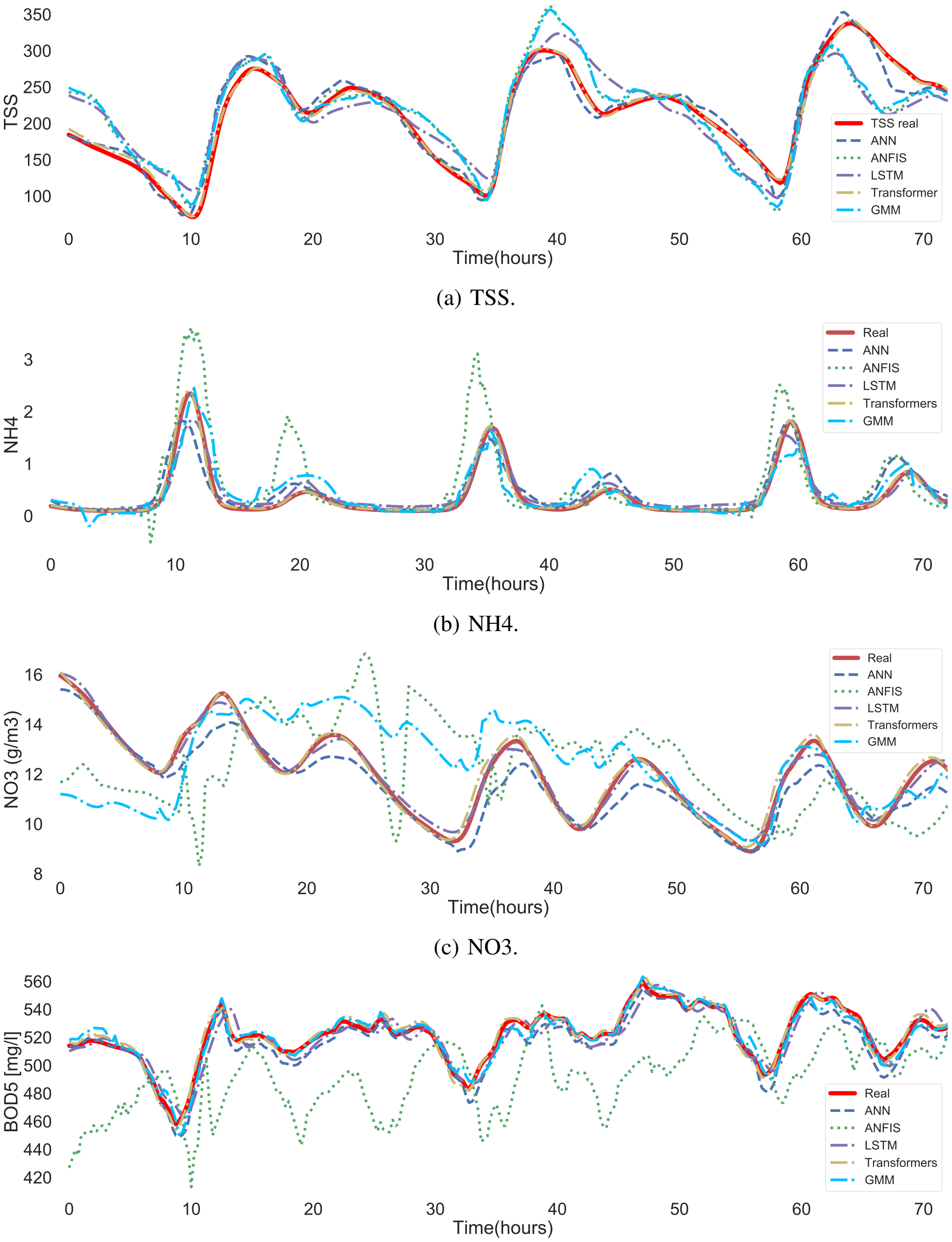
According to the predefined metrics, the best models obtained were:

- ANN: $Nl = 3$, $Nn = 100$, and $Epochs = 20$ for TSS, NO_3 and NH_4, and $Nl = 3$, $Nn = 200$, and $Epochs = 20$ for $BOD5$;
- LSTM: $Nl = 2$, $Nc_i = 64$, $Nc_h = 32$, $Bs = 16$ and $Epochs = 16$ for TSS, NO_3 and NH_4; and $Nl = 2$, $Nc_i = 64$, $Nc_h = 32$, $Bs = 32$ and $Epochs = 32$ for $BOD5$;
- ANFIS: MF = {triangular} and $Epochs = 100$ for TSS, NO_3, NH_4, and $BOD5$;
- Transformer: $Nn = 64$, $Bs = 32$ and $Epochs = 40$ for TSS, NO_3, NH_4, and $BOD5$;
- GMM: $K = 25$ for TSS and $BOD5$, and $K = 30$ for $NO3$ and $NH4$.

The datasets that represent each key variable of the case studies are composed of samples that correspond to 190 days of operation. The sampling frequency is 15 minutes, totaling 18240 samples. For training and hyperparameter validation tests, the equivalent of 180 days (17280 samples) was used, 70% for training and 30% for validation tests, and samples representing the last 10 days (960 samples) were used for final performance tests. All algorithms used the same datasets for training, testing and validation, according to the variables chosen for each case study.

4.1. Results

The results obtained by the algorithms are represented in Table 1 for the test dataset. The evaluation was performed according to the prediction of the key variables for a period corresponding to 10 days of operation. In order to facilitate the perception of the performance of the algorithms for each case study, Figure 6 presents graphically the prediction of the first three days of each target for each tested algorithm. As it can be seen from Table 1, the best performance for predicting all key variables was obtained by the Transformer



(a) TSS.

(b) NH4.

(c) NO3.

(d) BOD5.

Figure 6: Forecast results, for the first 72 hours, of case study variables: (a) TSS, (b) NH_4, (c) NO_3, and (d) $BOD5$.

algorithm with the lowest value of RMSE, MSE, MAE and MAPE. The second best performance when predicting TSS and NO_3 was obtained by the LSTM algorithm, when predicting NH_4 was achieved by ANN, and for $BOD5$ prediction was obtained by the GMM algorithm.

For the TSS variable, Transformer obtained a RMSE of 5.525 while the second best performance was obtained by LSMT with 8.461, that is, 55.06% higher than the Transformer. For the $NH4$ variable, Transformer obtained an MSE error very close to zero. As the data used had never been presented to the algorithm, the possibility of overfitting is excluded, indicating that the algorithm was able to model the patterns present in the time series. For $NH4$, the worst performance was obtained by the GMM with RMSE value 1478% higher than that reached by Transformer. When predicting the value of variable $BOD5$, the GMM obtained the second performance with RMSE of 4.988, being 21.8% higher than the error obtained by Transformer. For the

TABLE 1: Algorithm performances predicting key variables of the case study for the test dataset. According to assessment metrics, the best results are presented in bold.

Target	Algorithms	Assessment metrics			
		RMSE	*MSE*	*MAE*	*MAPE*
TSS	LSTM	8.461	71.589	7.452	5.329
	ANN	11.913	141.929	9.201	5.764
	ANFIS	12.894	166.268	9.711	5.820
	Transformer	**5.425**	**29.430**	**3.905**	**3.301**
	GMM	12.220	149.329	9.238	5.580
NH_4	LSTM	0.616	0.379	0.273	61.945
	ANN	0.495	0.245	0.271	61.260
	ANFIS	0.550	0.302	0.268	46.214
	Transformer	**0.051**	**0.003**	**0.030**	**8.895**
	GMM	0.754	0.568	0.465	184.624
NO_3	LSTM	0.320	0.102	0.249	2.247
	ANN	0.600	0.360	0.457	3.788
	ANFIS	1.140	1.301	0.786	6.483
	Transformer	**0.179**	**0.032**	**0.129**	**1.143**
	GMM	0.560	0.314	0.430	3.557
$BOD5$	LSTM	7.632	58.255	5.989	1.131
	ANN	5.506	30.315	4.458	0.851
	ANFIS	16.236	263.608	11.069	2.123
	Transformer	**4.095**	**16.769**	**3.166**	**0.606**
	GMM	4.988	24.879	4.093	0.782

variable $NO3$, the second position was occupied by the LSTM with MAE equal to 0.249, which is 93% higher than that obtained by the Transformer.

Despite presenting better results, the Transformer algorithm also has disadvantages in relation to the other methods analyzed in this paper. The algorithms ANN, LSTM, Transformer and GMM were implemented in Python, trained and evaluated on the same datasets, and for training the average of the computational time for the best model of each variable was for Transformer 12607 [s], LSTM 721 [s], ANN 125 [s], and GMM 71.4 [s] to complete the same task. The ANFIS was developed in MATLAB and ended the training process in 231 [s]. Thus, the greater need for computational resources is a limitation for Transformer, and could represent a limitation for real applications.

5. Conclusions

The paper proposed the application of machine learning algorithms to predict key variables in the wastewater treatment process. The ANN, LSTM, Transformer, ANFIS and GMM algorithms were used to approximate the values of the TSS, $BOD5$, NH_4, and NO_3 variables. The studied algorithms had their hyperparameters chosen through a grid search, and were evaluated using the RMSE, MSE, MAE, and MAPE prediction performance metrics. The Transformer algorithm presented better performance according to the considered metrics. The second best performance was presented by the ANN when predicting the values of NH_4, and by the LSTM when predicting the values of TSS and NO_3. The GMM had the second best performance in predicting the values of $BOD5$, demonstrating that each algorithm captures the patterns present in the time series in a different way, with the context of the information being an important feature for the quality of forecasts. As it is necessary to seek quality with the feasibility of predictions, it is important to emphasize that the Transformer algorithm requires greater computational resources and demands more time for training and predictions. ANN and LSTM presented good forecasts, according to the adopted metrics, and require less computational resources, with less training and forecasting time.

References

[1] S. Guerra-Rodríguez, P. Oulego, E. Rodríguez, D. N. Singh, and J. Rodríguez-Chueca, "Towards the implementation of circular economy in the wastewater sector: Challenges and opportunities," *Water*, vol. 12, no. 5, p. 1431, 2020.

[2] America's Infrastructure Report Card, "Wastewater," tech. rep., ASCE, USA, 2021.

[3] The European Federation of National Associations of Water Services, "Europe's water in figures - An overview of the European drinking water and waste water sectors," tech. rep., EurEAU, 2017.

[4] S. Alizadeh, H. Zafari-Koloukhi, F. Rostami, M. Rouhbakhsh, and A. Avami, "The eco-efficiency assessment of wastewater treatment plants in the city of mashhad using emergy and life cycle analyses," *Journal of Cleaner Production*, vol. 249, p. 119327, 2020.

[5] F. Hernández-Sancho, M. Molinos-Senante, and R. Sala-Garrido, "Energy efficiency in spanish wastewater treatment plants: A non-radial dea approach," *Science of the Total Environment*, vol. 409, no. 14, pp. 2693–2699, 2011.

[6] X. Dong, X. Zhang, and S. Zeng, "Measuring and explaining eco-efficiencies of wastewater treatment plants in china: an uncertainty analysis perspective," *Water Research*, vol. 112, pp. 195–207, 2017.

[7] B. Mamandipoor, M. Majd, S. Sheikhalishahi, C. Modena, and V. Osmani, "Monitoring and detecting faults in wastewater treatment plants using deep learning," *Environmental monitoring and assessment*, vol. 192, no. 2, pp. 1–12, 2020.

[8] T. Cheng, F. Harrou, F. Kadri, Y. Sun, and T. Leiknes, "Forecasting of wastewater treatment plant key features using deep learning-based models: A case study," *IEEE Access*, vol. 8, pp. 184475–184485, 2020.

[9] F. S. Mjalli, S. Al-Asheh, and H. Alfadala, "Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance," *Journal of Environmental Management*, vol. 83, no. 3, pp. 329–338, 2007.

[10] P. Zhou, Z. Li, S. Snowling, B. W. Baetz, D. Na, and G. Boyd, "A random forest model for inflow prediction at wastewater treatment plants," *Stochastic Environmental Research and Risk Assessment*, vol. 33, no. 10, pp. 1781–1792, 2019.

[11] V. Nourani, G. Elkiran, and S. Abba, "Wastewater treatment plant performance analysis using artificial intelligence–an ensemble approach," *Water Science and Technology*, vol. 78, no. 10, pp. 2064–2076, 2018.

[12] J. Mendes, R. Araújo, T. Matias, R. Seco, and C. Belchior, "Automatic extraction of the fuzzy control system by a hierarchical genetic algorithm," *Engineering Applications of Artificial Intelligence*, vol. 29, pp. 70–78, March 2014.

[13] M. Preisner, E. Neverova-Dziopak, and Z. Kowalewski, “An analytical review of different approaches to wastewater discharge standards with particular emphasis on nutrients,” *Environmental Management*, vol. 66, no. 4, pp. 694–708, 2020.

[14] U. Jeppsson, M.-N. Pons, I. Nopens, J. Alex, J. Copp, K. Gernaey, C. Rosén, J.-P. Steyer, and P. Vanrolleghem, “Benchmark simulation model no 2: general protocol and exploratory case studies,” *Water Science and Technology*, vol. 56, no. 8, pp. 67–78, 2007.

[15] F. R. Spellman, *Handbook of water and wastewater treatment plant operations*. CRC press, 2003.

[16] M. Kubat, “Neural networks: a comprehensive foundation by simon haykin, macmillan, 1994, isbn 0-02-352781-7.,” *The Knowledge Engineering Review*, vol. 13, no. 4, pp. 409–412, 1999.

[17] M. Dalto, J. Matuško, and M. Vašak, “Deep neural networks for ultra-short-term wind forecasting,” in *2015 IEEE international conference on industrial technology (ICIT)*, pp. 1657–1663, IEEE, 2015.

[18] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] J. Mendes, R. Maia, R. Araújo, and F. A. A. Souza, “Self-Evolving Fuzzy Controller Composed of Univariate Fuzzy Control Rules,” *Applied Sciences*, vol. 10, p. 5836, August 2020.

[20] J. Mendes, N. Sousa, and R. Araújo, “Adaptive predictive control with recurrent fuzzy neural network for industrial processes,” in *Proc. 16th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2011)*, (Toulouse, France), pp. 1–8, IEEE, September 5-9 2011.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[22] N. Wu, B. Green, X. Ben, and S. O’Banion, “Deep transformer models for time series forecasting: The influenza prevalence case,” *arXiv preprint arXiv:2001.08317*, 2020.

[23] Z. Ghahramani and M. I. Jordan, “Supervised learning from incomplete data via an em approach,” in *Advances in neural information processing systems*, pp. 120–127, 1994.

[24] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[25] A. Fabisch, “gmr: Gaussian mixture regression,” *Journal of Open Source Software*, vol. 6, no. 62, p. 3054, 2021.