



Forecasting wastewater influent parameters using neural networks (Case Study)

Abdellah CHAOUI^{1*}, *Wafae EL KHOUMSI*¹, *Mohamed LAAOUAN*²,
*Rqia BOURZIZA*¹, *Karima SEBARI*¹

¹IAV Hassan II, Rabat, Morocco.

²International Institute for Water and Sanitation, Rabat, Morocco.

Abstract

Machine learning forecasting has been widely used in order to increase WWTPs performance. Forecasting wastewater influent quality parameters is not only beneficial for operators and the plant itself, as it is also important environmentally, economically, and socially. In this paper, Zaio's WWTP BOD5 variable will be used as a case example. The current paper applies two time series forecasting models that are, based on machine learning, Recurring Neural Networks (RNN) and Elman Neural Networks (ENN) in order to provide more or less accurate predictions that will provide support to WWTPs management. The models used are statistically correct. The models were compared with error measures terms like MAPE, RMSE and R-squared. The ENN model has the highest error measures compared to the RNN trained models. Thus, the comparison concludes that the second RNN trained model is the one with the best fit, as it holds the least error terms. The second RNN model has low error terms and high coefficients of determination, with an MAPE of 0.19, RMSE of 66,55 and an R-squared of 0,68486. The objective of this paper is to find the best wastewater influent quality parameter forecasting model.

Keywords: Wastewater; WWTPs; BOD5; Forecasting; Machine learning; ENN; RNN

Case Study *Corresponding Author, e-mail: abdellahchaoui2000@gmail.com

1. Introduction

This paper introduces two forecasting models that are based on machine learning. These models are Recurring Neural Networks (RNN) and Elman Neural Networks (ENN). Machine learning forecasting has been widely used in order to increase WWTPs performance. In the contribution of Al-Asheh et al. [1], the authors applied different forecasting models that include Artificial Neural Networks (ANN), Adaptive Linear Neuron Networks (ADALINE), and Multi-Layer Feed-forward neural networks (ML-FF) to forecast influent quality parameters (BOD, COD, and TSS). Empirical findings indicate that the ML-FF model resulted in lower error terms. But concerning the contribution of Kriger and Tzoneva [2], authors used the ML-FF and Elman Recurring neural networks in order to forecast wastewater treatment plant influent disturbances. The current contribution will add to the existing literature by applying the two neural networks cited above (RNN and ENN). For this, the following section will introduce the methods that will be used, followed by the results sections. Finally, this paper will conclude with a discussion as well as the limitations of this work.

A. CHAOUI et al., 2023

2. Materials and Methods

It is essential to note that in the case of this contribution, the only used forecasted models are time series. By definition, time series variables refer to variables that are measured over time over successive times with a unified frequency such as daily, monthly, quarterly, annually, and whatnot. Forecasting models, as summarized in this following [3]–[15] :

- Extrapolative methods with simple moving average, exponential smoothing, such as the Holt-Winters method, and autoregressive moving average (ARMA) - aka Box-Jenkins.
- Explanatory variable methods with regression analysis, predictive modelling, artificial neural networks and econometric modelling.
- Simulation modeling methods with cell-based modelling, system dynamics simulation and multi-agent simulation.
- Judgmental methods.
- Composite methods with bayesian forecasting and others

like the combination of different forecasting methods and models is a type of composite methods.

Also, with regards to the contribution of Kim et al. [16], the artificial neural network (ANN) and the *ASM3 + Bio - P* models were used in order to forecast the influent and effluent in order to forecast the $NH_4^+ - N$. Furthermore, this contribution forecasted the control of the effluent water quality in the A^2/O (Anaerobic/Anoxic/Oxic) process for 1 day in advance. Empirical findings indicate that by applying this model, the $NH_4^+ - N$ concentration decreased by 68% by only increasing the air flow rate by 9%, as suggested by the model. For Nadiri, Shokri, Tsai, and Moghaddam [17], the application of forecasting models consisted of applying a proposed ensemble supervised committee fuzzy logic model to predict various wastewater influent parameters that are BOD, COD, and TSS. The used model is based on artificial neural networks and uses the MAPE as an error comparison tool. Results indicate that the proposed model performs better than individual fuzzy logic models.

With regards to the publication of Yu et al. [18], it has used a combined Kernel Principal Component Analysis (KPCA) and Extreme Machine Learning (EML) to forecast the inlet water quality of sewage treatment. This same contribution was compared to a back propagation neural network (BPNN) model and many other models. These models were compared based on the mean absolute error (MAE), the mean absolute percentage error (MAPE), and the root mean square error (RMSE). Findings indicate that forecasts resulted from the KPCA model were more accurate than any other model included in the study. Concerning the contribution of Lotfi et al. [19], many wastewater parameters were forecasted and include BOD, COD, TSS, and total dissolved solids (TDS). This contribution suggested a new methodology that consists of a combination of ARIMA, which is a stochastic model, and a nonlinear outlier robust extreme machine learning technique (ORELM). More than 144×8 linear models were considered (using ARIMA) in addition to more than 48 nonlinear models (using ORELM) and 48 hybrid models (using ARIMA-ORELM). Results show that the different obtained correlations between the observed values and the predicted ones could reach a value of 0.99.

For Abba, Nourani, and Elkiran [20], different variables such as PH and SS were forecasted using different models. These latter include general regression neural network (GRNN), Hammerstein-wiener (HW) and non-linear autoregressive with exogenous (NARX) neural network, and the least square support vector machine (LSSVM), besides many other nonlinear ensemble techniques. Findings indicate that each model was suitable for a specific variable. For instance, HW model showed better accuracy in forecasting SS while GRNN-E model was better in predicting pH and SS. Neural networks are modern computational systems (either software or hardware) that simulate biological systems via using a large number of interconnected artificial neurons. A neural net consists of neurons that enable establishing a connection between them while maintaining a direct communication link between each of them. In addition to that, this established communication links have different weights and represent information using in the neural net in order to

solve a given problem. This is given by the following formula [21]:

$$O_j = f \left(\sum_{i=1}^n w_{ij} I_i + b_j \right) \tag{1}$$

Where:

- O_j : The output of the jth neuron.
- f : represents the transfer function used or the activation of the neuron.
- b_j : represents the bias of the jth neuron.
- w_{ij} : represents the synaptic weight of the ith synapse of the jth neuron.
- I_i : represents the input signal of the jth neuron.
- n : represents the total number of the input signals to the jth neuron.

The equation is represented in figure 1 [21]. The logarithmic sigmoid, hyperbolic tangent sigmoid, and linear functions are the most common used activation or transfer functions used in neural networks.

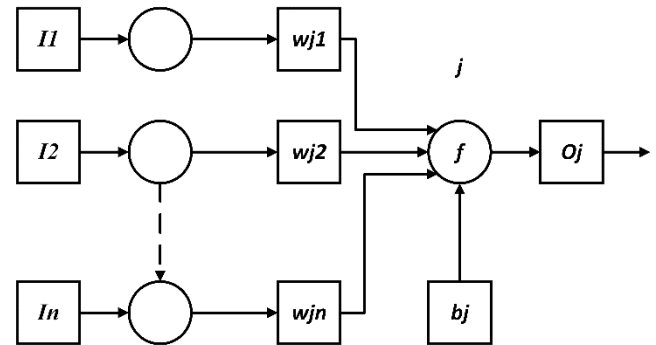


Figure 1 : Schematic diagram of neuron j and its link

2.1. Zaio WWTP description

The Zaio WWTP (Figure 2) is designed for a capacity of 2,454.00 m³ per day and allows the treatment of suspended solids, chemical oxygen demand, biological oxygen demand. This is throughout a pre-treatment stage and a biological treatment of the trickling filter type.

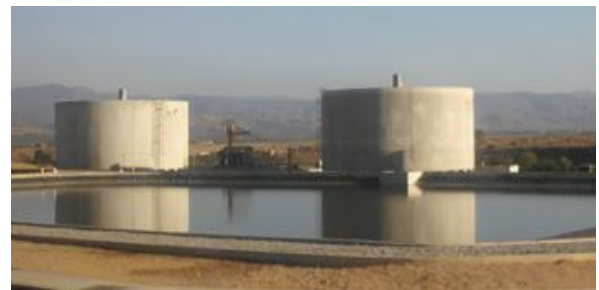


Figure 2: Zaio WWTP

The raw wastewater flow rates used are as follows (Table 1):

Table 1 : Raw wastewater flow rates

Parameters	Units	Values
Nature of raw water	Domestic	
Average daily flow	m ³ /j	2.454
Daily BOD5	Kg/j	1.364
Suspended Solids	Kg/j	955
Daily COD	Kg/j	2.455

The average daily concentrations of raw wastewater are as follows (Table 2):

Table 2: Average daily concentrations of raw wastewater

Parameter	Unit	Value
BOD5	mg/l	556
COD	mg/l	1001
TSS	mg/l	389

The following section will introduce all the treatment process included in this wastewater treatment plan. This is divided into water and sludge sectors. The water queue is made up of two treatment queues, one of which is backup (in the case of pre-treatment), to give the station flexibility and security in maintenance operations. It consists of the following stages:

- Pre-treatment: The pre-treatment structures are designed for the reception of peak flow in rainy weather 355,5 m³ / h;
 - Screening: this stage is made up of two channels, one of which is emergency with 10 mm air gap (mechanic) and the other channel is provided with an inclined screen with automatic cleaning of 06 mm air gap.
 - Flow measurement: The flow measurement of the wastewater is done using an Parshall canal flow meter.
- Biological treatment: Biological treatment is provided by two anaerobic ponds and two trickling filters.
 - Anaerobic ponds: These ponds are of rectangular type with a unit volume of 4.366 m³ /u and 3,75 m of depth.
 - Trickling filters: These filters are of circular type with a unit volume of 891,7m³/u and 14,50 m of diameter and 5,40 m of height with natural aeration 31 windows de 1 m² (1m*1m) and a sprinkler with 0.03 rpm of velocity.
 - Secondary settling tank: secondary decantation is ensured by one secondary clarifiers. This clarifier is lamellar type with a unit volume of 400 m³.
 - Flow measurement: The flow measurement of treated water is done using an Parshall canal flow meter.

The sludges queue is made up of two treatments:

- Extraction secondary sludge: The sludge is pumped by two pumps, one of which is an emergency backup

installed of 17,20 m³/h per line, towards the anaerobic ponds.

- Extraction primary sludge: The sludge is pumped by one pump installed of 27 m³/h, towards the anaerobic ponds.
- Drying beds: Sludge drying is done using sixteen (16) drying beds, each one is of rectangular type with a unit surface of 250 m²/u.

2.2. Data

The data used in this contribution was measured in the Zaio WWTP between 02/10/2017 and 01/10/2018. The data set accounts for the wastewater influent and effluent parameters that is BOD5 influent and effluent quality variable. The values of this variable are stated in mg/L and will enable calculating the daily treated BOD5 variable.

2.3. Elman Neural Network Model

The current section introduces the Elman Neural Network model that was first proposed by Elman in 1990, which is considered as a typical dynamic neural network [22]. The standard model accounts for an input layer, a hidden layer, an output layer, and a context layer such as shown in figure 3 [22].

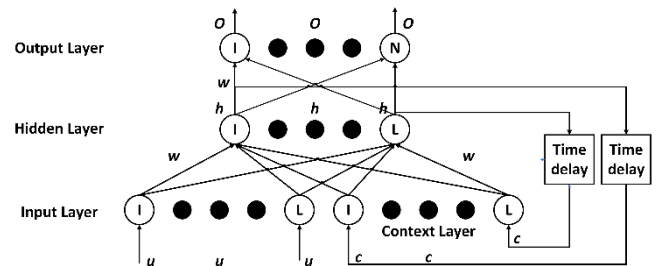


Figure 3: Structure of the Elman Neural Network

Concerning the role of the context layer, it serves as the information record that enables transferring information from the last network iteration to the newest one. According to Chiang et al. [23], this feature makes the model more suitable in forecasting time series variables compared to other time series models. In the contribution of Li et al. [22], the model’s computational process can be expressed by the following formulas:

- The output of the output layer at period t:

$$O_q^{(t)} = f \left(\sum_{j=1}^L h_j^{(t)} \omega_{j,q} \right) \tag{2}$$

- The output of the hidden layer at period t:

$$h_j^{(t)} = f \left(\sum_{i=1}^L u_i^{(t)} \omega_{i,j} + \sum_{k=1}^L c_k^{(t-1)} \omega_{j,k} \right) \tag{3}$$

- The output of the context layer at period t-1:

$$c_k^{(t-1)} = h_j^{(t)} \tag{4}$$

Where f represents the activation function. In the Scase of this contribution, the activation function of the hidden layer is the sigmoid function and is given such as:

$(x) = 1/(1 + e^{-x})$, and the activation function of the output layer is linear and is given such as: $f(x) = x$. Concerning $\omega_{j,q}$, $\omega_{i,j}$, and $\omega_{j,k}$, they represent the connection weights between the layers, respectively.

The following model consists of a learning model that accounts for different steps in its process. With regards to the first step, it consists of a random function that has as a purpose initiating the connection weights between the network's layers in order to determine the allowable error. In the case of this contribution, the error term is represented by the mean square error (MSE) that is given such as [22]:

$$E = \frac{1}{N} \sum_{q=1}^N (O_{eq}^{(t)} - O_q^{(t)})^2 \quad (5)$$

Where $O_{eq}^{(t)}$ represents the expected output at period t of the network, and $O_q^{(t)}$ represents the actual output of the network at the same period. Concerning the second step, it consists of normalizing the input data and calculating the value of the error.

This is followed by updating the connection weights between layers according to the calculated error, and using the momentum gradient descent algorithm. Concerning the normalization of input data and the weight change, they are given such as [22]:

$$z = (z_{max} - z_{min}) \frac{q - q_{min}}{q_{max} - q_{min}} + q_{min} \quad (6)$$

$$\Delta\omega^{(k)} = -\eta \frac{\partial E}{\partial \omega^{(k)}} + \alpha \Delta\omega^{(k-1)} \quad (7)$$

Where q is the principal component sequence, and the minimum value of the sequence and the maximum value of the sequence are q_{min} and q_{max} , respectively. For the normalized data z , it is between 1 and 0 that corresponds to z_{max} and z_{min} , respectively. $\Delta\omega^{(k)}$ represents the change in the network weights in the k th update, and $\Delta\omega^{(k-1)}$ represents the change in the network weights in the $k - 1$ th update. In the case of this contribution, α , which is the momentum constant is given such as $\alpha=0.9$, and η , which is the learning rate, is given such as $\eta = 0.05$. The maximum number of iterative trainings is set in order to stop the Elman Neural Network training process.

2.4. Recurring Neural Networks

This section introduces the RNN model that is used for sequential data. The contribution of Hochreiter and Schmidhuber [24] indicate that this type of models uses a long short-term memory that is considered to be a gated unit for neural networks. RNN model has 3 gates that manage the memory's content. These gates are given by simple logistics functions of weighted sums that are learnt by back propagation [25]. This is illustrated in figure 4 [26].

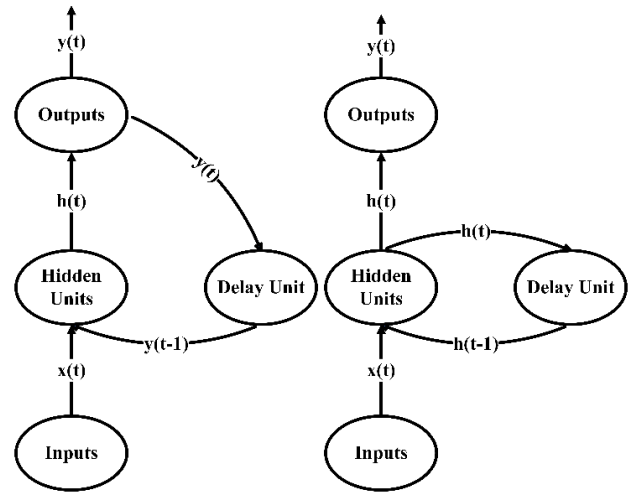


Figure 4: Two simple recurrent network structures

In this process, the cell state is managed by the input gate and the forget gate (or the long-term memory). In addition to that, the input state results in a hidden state. This can be mathematically expressed such as [25]:

- Input gate:

$$i_t = \text{sigmoid}(W_i x_t + U_i h_{t-1} + b_i) \quad (8)$$

- Forget gate:

$$f_t = \text{sigmoid}(W_f x_t + U_f h_{t-1} + b_f) \quad (9)$$

- Cell state:

$$o_t = \text{sigmoid}(W_o x_t + U_o h_{t-1} + b_o) \quad (10)$$

- Output gate:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c x_t + U_c h_{t-1} + b_c) \quad (11)$$

- Hidden state:

$$h_t = o_t \odot \tanh(c_t) \quad (12)$$

RNN models can learn as well as memorize the time series characteristics. However, it is important to encode many features such as trend and seasonality. For this, it is necessary to normalize the values of the dataset such as [25]:

$$\tilde{x} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (13)$$

Or

$$\tilde{x} = \frac{x - \text{mean}(x)}{sd(x)} \quad (14)$$

The main features of RNN models are lags, trend, seasonality, and dummy indicators. Concerning the lag attribute, they are used as the main input for forecasting. Yet, the only drawback in using this feature is losing the first observation. Still, the higher is the number of the observations, the lower the impact of this feature is. But with regards to the trend attribute, it refers to the ability of the model to learn the trend of the time series over time and duplicate it. This is also the case of the seasonal attribute. But for the dummy variable feature, it refers to indexing special or unusual events such as holidays [25].

Finally, it is important to note that forecasting using

RNN models is suitable for only one-step ahead. In order to use a multi-step-ahead forecasting, it is important to use an iterative or recursive manner. This latter step can be explained by looping over the RNN model for each new point forecast, and including each new prediction at the beginning of each new loop iteration [25].

2.5. Forecast performance measures

Error measures comparison help to compare different forecasting models in order to know their performances and apply the most suitable model. There are many error measures, some of them are [27]–[31]:

- The mean forecast error (MFE) referred to as the forecast bias and which measures the average deviation between actual and predicted values. However, since errors can be positive or negative, this error measure cancels out, and does not provide the real error value.
- The mean absolute error (MAE), which measure solves the issues of positive and negative errors that cancel out, and shows the magnitude of the overall error. The lower is this value, the better is the forecast model.
- The mean percentage error (MPE), which measures the average error stated in terms of percentage
- The mean squared error (MSE), which measures the squared deviation of predicted values
- The sum of squared error (SSE), which measures the total squared deviation instead of the average

In this paper and in order to compare between the accuracy of the predictions of the ENN and RNN models, three error measures will be used that are the root mean square error (RMSE), the mean absolute percentage error (MAPE), and the coefficient of determination (R-squared). For RMSE and MAPE, lower values indicate a better prediction, but for the R-squares, the higher the percentage, the better the predictions. These error measures are given such as [32]:

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (\hat{y}_t - y_t)^2}{n}} \quad (15)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{y_t - \hat{y}_t}{y_t} \right| \quad (16)$$

Where:

- \hat{y}_t : predicted values;
- y_t : regression’s dependent variables;
- n: all periods.

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\text{Explained variation}}{\text{Total variation}} \quad (17)$$

3. Results and Discussion

3.1. Elman Neural Network model results

The following section will present the steps as well as the results of the BOD5 forecasting using ENN model.

3.1.1. Exploring and preparing the data

The first step under this section is checking the

decomposition of our time series variable using a multiplicative model given such as:

$$BOD5_t = trend_t \times seasonal_t \times random_t \quad (18)$$

Where random refers to the error component.

This done throughout the “decomp” function in R. The decomposition of the BOD5 variable (Figure 5) shows that the trend component is nonlinear, as it contains periods of increase and period of decrease. In addition to that, the seasonal component is not clear.

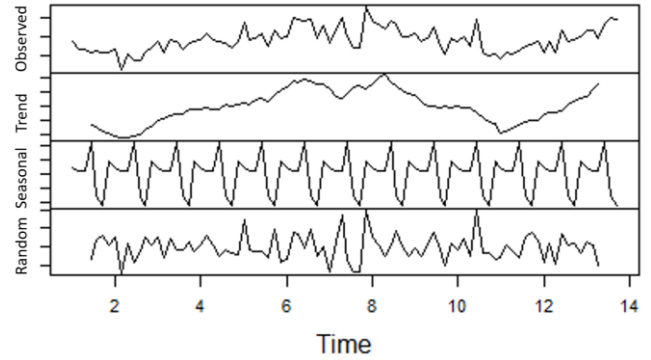


Figure 5: Decomposition of BOD5 variable

The next step relates to checking the distribution of the error component. In the case of this research, figure 6 indicates that the error density has a normal distribution shape, which is further confirmed by the linear line shown in the Q-Q plot in figure 7.

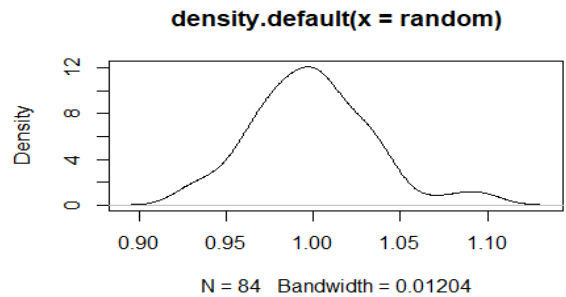


Figure 6: Density plot of the random component of the BOD5 variable

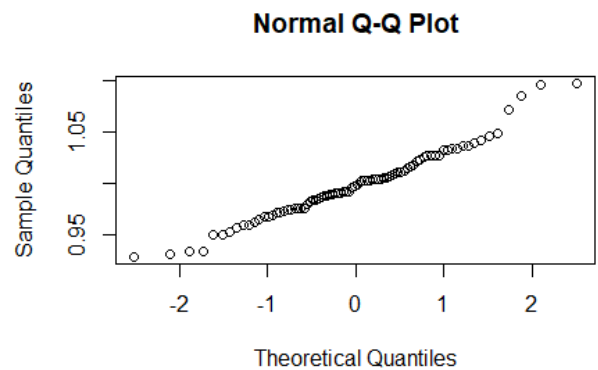


Figure 7: Q-Q plot of the random component of the BOD5 variable

In order to check the stationarity of the data, the next step is to plot the ACF plot of the BOD5 variable. It is clearly

shown in figure 8 that there are numerous spikes outside of the confidence interval, which indicates that there is an obvious autocorrelation between lagged BOD5 values and current ones.

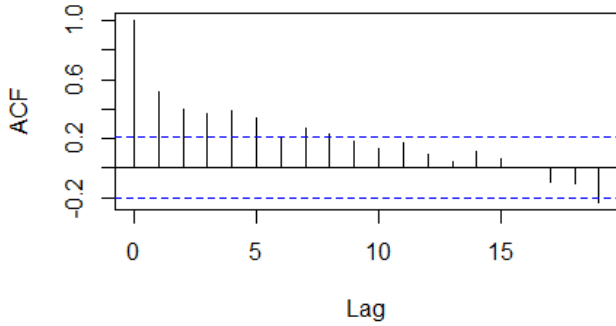


Figure 8: Auto correlations of the BOD5 variables

Concerning the last part of the data exploration and preparation, the data will be normalized, and the lag attributes will be created. Since this contribution uses daily data, 7 lagged attributes will be added in order for the R program to understand this frequency.

3.1.2. Training the model on the data

The following section will train the ENN model using two third of the data sample, while only one third of the data sample will be used in order to test the model. The following part requires the “RSNNS” package in R. The trained ENN model accounts for 2 hidden layers where each one has 2 nodes. This model runs for 100 000 iterations. After training this model, figure 9 shows the error in each iteration. It is clearly seen that that model is able to learn from the historical data. In addition to that, figure 10 represents the fitted versus actual BOD5 values regression curves. The black line shows the linear regression curve of actual values while the red line shows the linear regression curve of fitted values. As both these two lines fit the data, it further confirms that the ENN model was able to train by learning from historical values. Finally, the trained model has resulted in an R-squared of 0.6679.

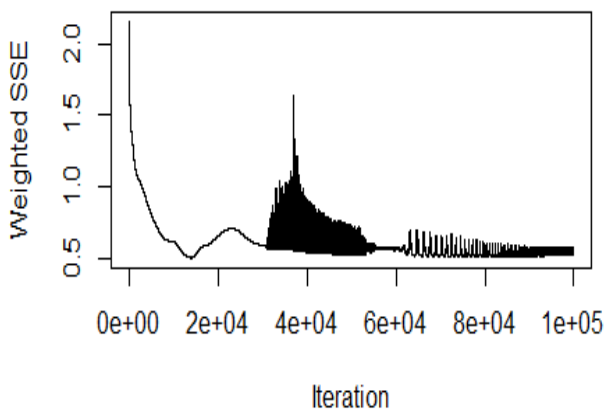


Figure 9: Error by iteration for ENN model

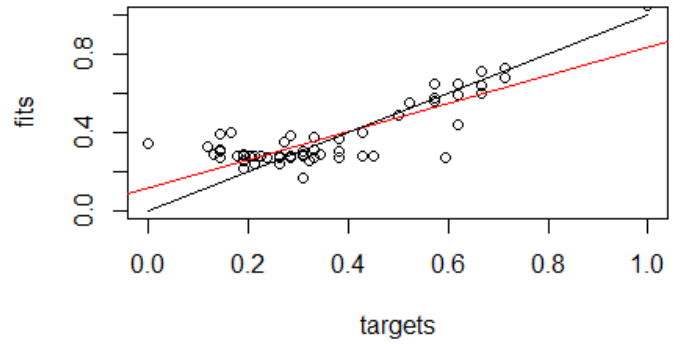


Figure 10: Fitted and actual values regression curve

Figure 11 represents a graphical illustration that represents the actual BOD5 values versus the trained ENN model values. It is noticeable that the trained values follow the same pattern as the actual BOD5 values. In addition to that, the variance or standards deviation are minimal.

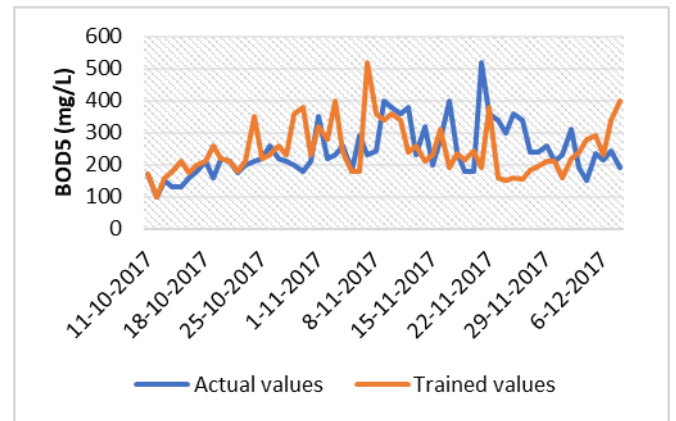


Figure 11: Trained versus actual BOD5 values

3.1.3. Assessing test performance

The final step of this model is to assess the test performance. This step requires predicting BOD5 values in the test sample in order to check its performance. In order to make the data usable, it is necessary to unscale the date in order to get them back to their original form. The following graph (Figure 12) shows the resulted BOD5 values from the ENN test sample. This graph does not clearly indicate whether if the model was able to accurately predict the BOD5 values or not. For this, the error measure tools will be used in a later section to assess the ENN model and compare it with the RNN model.

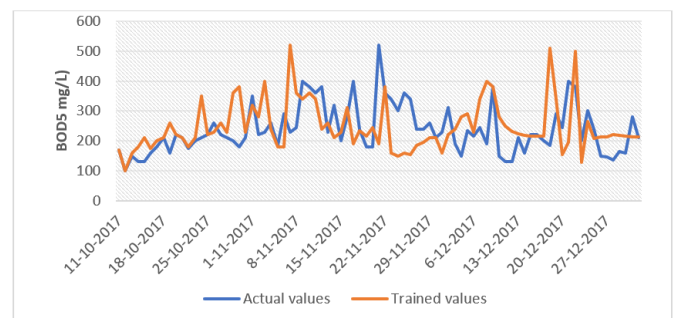


Figure 12: Actual versus tested ENN model BOD5 values

3.2. Recurring Neural Network model results

The following section will present the steps as well as the results of the BOD5 variable predictions using the recurring neural network model. Figure 13 shows the BOD5 time series variable for a period of 90 days. This figure indicates that the time series variable has a lot of fluctuations, which refers to a high variation or a high standard deviation. For this, the next step consists of exploring as well as preparing the data to fit within a recurring neural network.

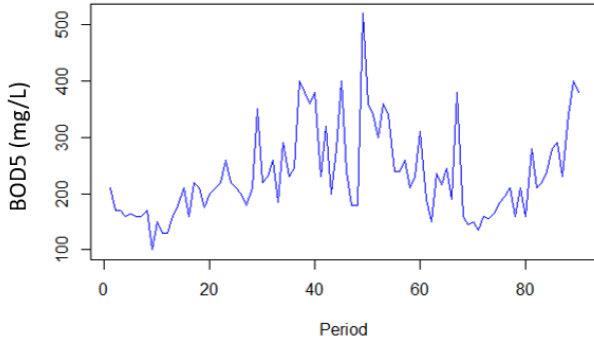


Figure 13: BOD5 in mg/l for 90 consecutive days

3.2.1. Exploring and preparing the data

In building a recurrent neural network, it is of prime important to explore and prepare the data. This model is univariate and consists of the creation of time lag attributes. For this, 4 time lagged attributes will be created such as to have 4 variables in addition to the initial BOD5 variables. Each of the created variables will account for a lag of 1, 2, 3, and 4, respectively. In addition to this step, the BOD5 variable is transformed using the log function. The steps discussed above will result in a matrix that has missing values. For this, it is necessary to remove the first 4 lines of this matrix to remove these missing values. The second part of the data preparation step relates to the normalization of the values as described in the methods section, to become within the range (0,1). Concerning the last step of the data preparation, it relates to the creation of the train and test samples. Since our dataset accounts for 90 daily variables, two third of the BOD5 values will be used to train the model, and one third of the BOD5 values will be used to test the model.

3.2.2. Training the model on the data

The current section will train the RNN model. This part requires the “rnn” package, and uses the “train” function to train the model. This latter function requires a 3 dimensional array to pass the attribute data that are mainly the number of samples, time, and number of variables. This corresponds to the numbers 1, 60, and 4, respectively. In the case of this contribution, the first model that will be trained has 3 nodes in the hidden layer. In addition to that, the model’s learning rate is 0.05, the number of numepochs is 10 000, and the transfer or activation function is sigmoid as described in the results section. After training the model, the error curve decreases rapidly after the first 1000 epochs. This indicates that the model is learning rapidly from the data (Figure 14). The trained model accounts for an error margin of less than 6.6, and this value can be further decreased by increasing the numbers of epochs.

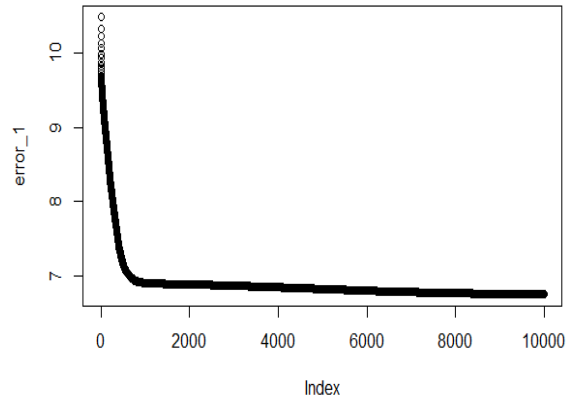


Figure 14: Error curve by epoch for model1

This model has resulted in an R-squared value of 0.6796. In order to improve this model, the same model will be trained. Yet, and similar to model 1, this model will account for 3 nodes exists while 2 hidden layers will be added. Figure 15 shows the error curve by epoch. This model has resulted in an R-squared of 0.68486, which is higher than the previous model.

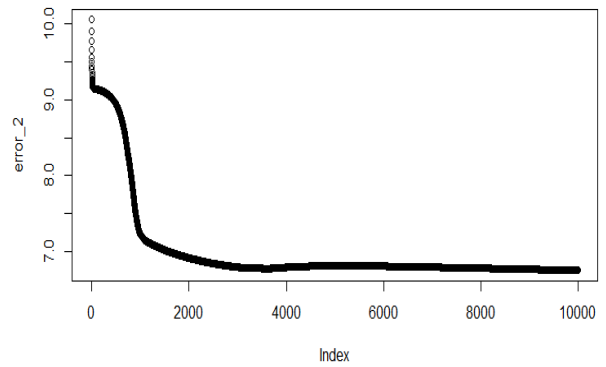


Figure 15: Error curve by epoch for model2

The following two graphs (Figures 16 and 17) represent the actual BOD5 variables and the trained model. Both models indicate that the trained values fit the actual values, as they have the same trends.

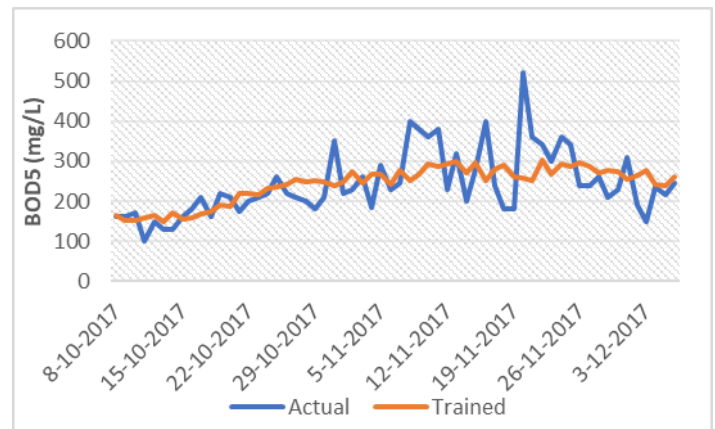


Figure 16: Actual versus trained variables of BOD5 variable – model 1

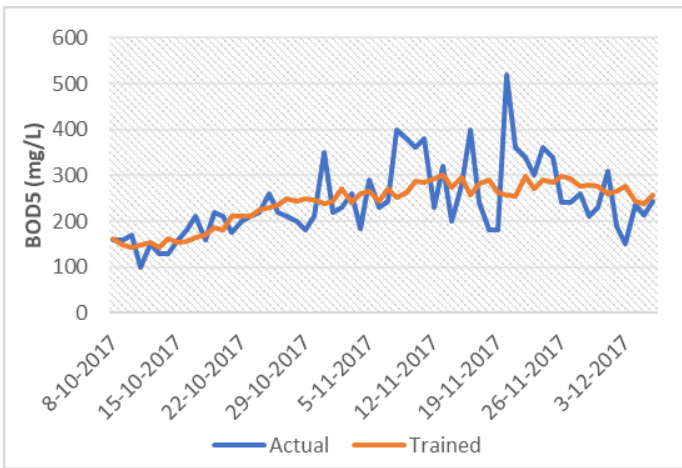


Figure 17: Actual versus trained variables of BOD5 variable – model 2

3.2.3. Preparing the test set data and forecasting using the model

After improving the model, the test sample will be created in order to forecast the BOD5 variable using the RNN model. This step enabled forecasting the BOD5 variable using the trained models. Yet, and in order to make this data ready to use, it is necessary to unscale the data and use the exponential function to make it in its original form. After unscaling the data, and transforming it to its natural state using the exponential function. The model has learned from the historical data and provided good forecasts. Yet, in a later section, the error measures will be calculated in order to compare between the RNN model and the ENN model. The last step of this section consists of making the forecast of 6 periods after the end of the study period. It is necessary to improve the model by re-optimizing the weights as new observations become available. For this, the following part will use a loop that includes all the steps used above. This accounts for opening the loop, preparing the training data, specifying the model, testing the data, and making the prediction. Each iteration of this loop makes a new point forecast. The following line graph (Figure 18) represents the forecasted values of the BOD5 variable for the next 6 days after the study period.

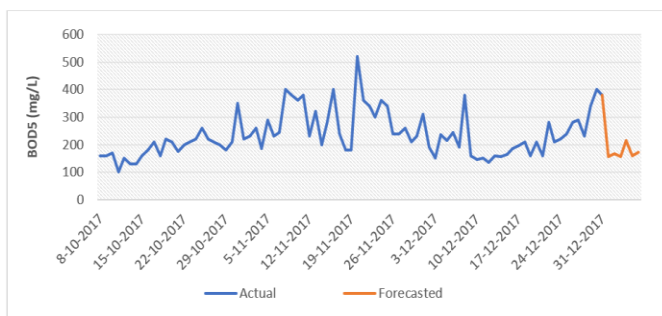


Figure 18: Forecasts of BOD5 variable for 6 days using RNN model

3.3. Models comparison tools

The following section represents the error measures for each model. With regards to the ENN model, it has resulted in a MAPE with a value of 0.33 and an RMSE with a value of 105.64. In addition to that, this latter model resulted

in an R-squared of 0.6679 (Table 3). Comparing to other models, the ENN model has the highest error measures compared to the RNN trained models (both model 1 and model 2). This is because the RNN model has resulted in the values of 0.21 and 0.19 for the MAPE, values of 67.09 and 66.55 for the RMSE, and an R-squared with the values of 0.6796 and 0.68486, for model 1 and model 2, respectively (Table 3). This comparison concludes that the second RNN trained model is the one with the best fit, as it holds the least error terms.

Table 3: Models’ error measures

	MAPE	RMSE	R-squared
ENN	0.33	105.64	0.66790
RNN model 1	0.21	67.09	0.67960
RNN model 2	0.19	66.55	0.68486

4. Conclusions

This paper applied two time series forecasting models that are based on machine learning. These models are Elman Neural Network, and Recurring Neural Network. Analyses indicate that the models used are statistically correct. In addition to that, both models resulted in a significantly low error terms and high coefficients of determination. Furthermore, empirical findings indicate that the Recurrent Neural Network is the model that holds the lowest error terms, which makes it the most suitable. It is important to note that among the main limitations of this study can be summarized such as:

- The RNN model is most suitable for predicting only one-step forward. For this, it is important to use an iterative process that includes new predictions for multi-step forward forecasting.
- These models require lots of statistical as well as programming skills that might be hard to apply among WWTPs professionals. Yet, they can be automated.
- The programs used might take time to run.

References

- [1] S. Al-Asheh, F. S. Mjalli, and H. E. Alfadala, “Forecasting influent-effluent wastewater treatment plant using time series analysis and artificial neural network techniques,” *Chemical Product and Process Modeling*, vol. 2, no. 3, p. 3, May 2007, doi: 10.2202/1934-2659.1063.
- [2] C. Kriger and R. Tzoneva, “Neural networks for prediction of wastewater treatment plant influent disturbances,” *IEEE AFRICON Conference*, 2007, doi: 10.1109/AFRCON.2007.4401646.
- [3] J. X. Che and J. Z. Wang, “Short-term load forecasting using a kernel-based support vector regression combination model,” *Appl Energy*, vol. 132, pp. 602–609, Nov. 2014, doi: 10.1016/J.APENERGY.2014.07.064.
- [4] B. Dyson and N. Bin Chang, “Forecasting municipal solid waste generation in a fast-growing urban region with system dynamics modeling,” *Waste Management*, vol. 25, no. 7, pp. 669–679, Jan. 2005, doi: 10.1016/J.WASMAN.2004.10.005.
- [5] E. Erdem and J. Shi, “ARMA based approaches for forecasting the tuple of wind speed and direction,”

- Appl Energy, vol. 88, no. 4, pp. 1405–1414, Apr. 2011, doi: 10.1016/J.APENERGY.2010.10.031.
- [6] E. W. Frees, R. A. Derrig, and Glenn. Meyers, “Predictive modeling applications in actuarial science”, Cambridge University Press, Vol. 1, 2014,.
- [7] C. F. S. P J Harrison, “Bayesian Forecasting,” Journal of the Royal Statistical Society. Series B (Methodological). Accessed: Apr. 27, 2023. [Online]. Available: <https://www.jstor.org/stable/2984970>
- [8] F. R. Johnston, J. E. Boyland, M. Meadows, and E. Shale, “Some Properties of a Simple Moving Average when Applied to Forecasting a Time Series,” J Oper Res Soc, vol. 50, no. 12, p. 1267, Dec. 1999, doi: 10.2307/3010636.
- [9] A. S. Khwaja, A. Anpalagan, M. Naeem, and B. Venkatesh, “Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting,” Electric Power Systems Research, vol. 179, p. 106080, Feb. 2020, doi: 10.1016/J.EPSR.2019.106080.
- [10] M. Lawrence, P. Goodwin, M. O’Connor, and D. Önkal, “Judgmental forecasting: A review of progress over the last 25 years,” Int J Forecast, vol. 22, no. 3, pp. 493–518, Jan. 2006, doi: 10.1016/j.ijforecast.2006.03.007.
- [11] G. Li, H. Song, and S. F. Witt, “Recent Developments in Econometric Modeling and Forecasting,” Journal of Travel Research, vol. 44, no. 1, pp. 82–99, Aug. 2005, doi: 10.1177/0047287505276594.
- [12] J. D. Melo, E. M. Carreno, and A. Padilha-Feltrin, “Multi-agent simulation of urban social dynamics for spatial load forecasting,” IEEE Transactions on Power Systems, vol. 27, no. 4, pp. 1870–1878, 2012, doi: 10.1109/TPWRS.2012.2190109.
- [13] E. Ostertagová and O. Ostertag, “Forecasting using simple exponential smoothing method,” Acta Electrotechnica et Informatica, vol. 12, no. 3, Jan. 2012, doi: 10.2478/v10198-012-0034-2.
- [14] T. Schmidberger, C. Posch, A. Sasse, C. Gülch, and R. Huber, “Progress toward forecasting product quality and quantity of mammalian cell culture processes by performance-based modeling,” Biotechnol Prog, vol. 31, no. 4, pp. 1119–1127, Jul. 2015, doi: 10.1002/BTPR.2105.
- [15] G. C. S. Wang and C. L. Jain, “Regression analysis : modeling & forecasting,” p. 293, 2003, Accessed: May 02, 2023. [Online]. Available: https://books.google.com/books/about/Regression_Analysis.html?hl=fr&id=dRQAkWHHmtwC
- [16] H. S. Kim, Y. J. Kim, S. P. Cheon, G. D. Baek, S. S. Kim, and C. W. Kim, “Evaluation of model-based control strategy based on generated setpoint schedules for NH₄-N removal in a pilot-scale A2/O process,” Chemical Engineering Journal, vol. 203, pp. 387–397, Sep. 2012, doi: 10.1016/J.CEJ.2012.07.067.
- [17] A. A. Nadiri, S. Shokri, F. T. C. Tsai, and A. Asghari Moghaddam, “Prediction of effluent quality parameters of a wastewater treatment plant using a supervised committee fuzzy logic model,” J Clean Prod, vol. 180, pp. 539–549, Apr. 2018, doi: 10.1016/J.JCLEPRO.2018.01.139.
- [18] T. Yu, S. Yang, Y. Bai, X. Gao, and C. Li, “Inlet Water Quality Forecasting of Wastewater Treatment Based on Kernel Principal Component Analysis and an Extreme Learning Machine,” Water 2018, Vol. 10, Page 873, vol. 10, no. 7, p. 873, Jun. 2018, doi: 10.3390/W10070873.
- [19] K. Lotfi, H. Bonakdari, I. Ebtehaj, F. S. Mjalli, M. Zeynoddin, R. Delatolla, and B. Gharabaghi, “Predicting wastewater treatment plant quality parameters using a novel hybrid linear-nonlinear methodology,” J Environ Manage, vol. 240, pp. 463–474, Jun. 2019, doi: 10.1016/J.JENVMAN.2019.03.137.
- [20] S. I. Abba, V. Nourani, and G. Elkiran, “Multi-parametric modeling of water treatment plant using AI-based non-linear ensemble,” Journal of Water Supply: Research and Technology-Aqua, vol. 68, no. 7, pp. 547–561, Nov. 2019, doi: 10.2166/AQUA.2019.078.
- [21] B. ZareNezhad and A. Aminian, “A multi-layer feed forward neural network model for accurate prediction of flue gas sulfuric acid dew points in process industries,” Appl Therm Eng, vol. 30, no. 6–7, pp. 692–696, May 2010, doi: 10.1016/J.APPLTHERMALENG.2009.11.017.
- [22] C. Li et al., “Runoff Prediction Method Based on Adaptive Elman Neural Network,” Water 2019, Vol. 11, Page 1113, vol. 11, no. 6, p. 1113, May 2019, doi: 10.3390/W11061113.
- [23] Y. M. Chiang, L. C. Chang, and F. J. Chang, “Comparison of static-feedforward and dynamic-feedback neural networks for rainfall–runoff modeling,” J Hydrol (Amst), vol. 290, no. 3–4, pp. 297–311, May 2004, doi: 10.1016/J.JHYDROL.2003.12.033.
- [24] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” Neural Comput, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/NECO.1997.9.8.1735.
- [25] G. Petneházi, “Recurrent Neural Networks for Time Series Forecasting,” arXiv preprint, 2019, <https://arxiv.org/abs/1901.00069v1>
- [26] N. D. C. Lewis, “Neural networks for time series forecasting with R : an intuitive step by step blueprint for beginners,” AusCov, p. 220, 2017.
- [27] R. Adhikari and R. K. Agrawal, “An Introductory Study on Time Series Modeling and Forecasting,” LAP Lambert Academic Publishing, Germany, Feb. 2013, Accessed: Apr. 26, 2023. [Online]. Available: <https://arxiv.org/abs/1302.6613v1>
- [28] L. J. Cao and F. E. H. Tay, “Support vector machine with adaptive parameters in financial time series forecasting,” IEEE Trans Neural Netw, vol. 14, no. 6, pp. 1506–1518, Nov. 2003, doi: 10.1109/TNN.2003.820556.
- [29] C. Hamzaçebi, “Improving artificial neural networks’ performance in seasonal time series forecasting,” Inf Sci (N Y), vol. 178, no. 23, pp. 4550–4559, Dec. 2008, doi: 10.1016/J.INS.2008.07.024.
- [30] H. Park, “Forecasting Three-Month Treasury Bills Using ARIMA and GARCH Models,” Econ 930,

Department of Economics, Kansas State University.,
1999.

- [31] S. Radha and M. Thenmozhi, “Forecasting Short Term Interest Rates Using Arma, Arma-Garch and Arma-Egarch Models,” In Indian Institute of Capital Markets 9th Capital Markets Conference Paper, 2006, doi: 10.2139/SSRN.876556.
- [32] G. G. C. Ç. N. Tirkeş, “Demand forecasting: a comparison between the Holt-Winters, trend analysis and decomposition models,” *Tehnicki vjesnik - Technical Gazette*, vol. 24, no. Supplement 2, pp. 503–509, Sep. 2017, doi: 10.17559/TV-20160615204011.