

Article

Predicting Effluent Quality in Full-Scale Wastewater Treatment Plants Using Shallow and Deep Artificial Neural Networks

Raed Jafar ^{1,*}, Adel Awad ², Kamel Jafar ³ and Isam Shahrour ⁴ ¹ Engineering Faculty, Manara University, Lattakia, Syria² Environmental Engineering Department, Tishreen University, Lattakia P.O. Box 1385, Syria³ Information Technology Department, Syrian Virtual University, Damascus P.O. Box 35329, Syria⁴ Laboratory of Civil Engineering and Geo-Environment (LGCgE), University of Science and Technology of Lille, 59650 Villeneuve-d'Ascq, France

* Correspondence: raed.jafar@manara.edu.sy or raedjafar@yahoo.fr; Tel.: +963-948309911

Abstract: This research focuses on applying artificial neural networks with nonlinear transformation (ANNs) models to predict the performance of wastewater treatment plant (WWTP) processes. The paper presents a novel machine learning (ML)-based approach for predicting effluent quality in WWTPs through explaining the relationships between the multiple influent and effluent pollution variables of an existing WWTP. We developed AI models such as feed-forward neural network (FFNN) and random forest (RF) as well as deep learning methods such as convolutional neural network (CNN), recurrent neural network (RNN), and pre-train stacked auto-encoder (SAE) in order to avoid various shortcomings of conventional mechanistic models. The developed models focus on providing an adaptive, functional, and alternative methodology for modeling the performance of the WWTP. They are based on pollution data collected over three years. It includes chemical oxygen demand (COD), biochemical oxygen demand (BOD₅), phosphates (PO_4^{-3}), and nitrates (NO_3^{-}), as well as auxiliary indicators including the temperature (T), degree of acidity or alkalinity (pH), electric conductivity (EC), and the total dissolved solids (TDS). The paper presents the results of using SNN- and DNN-based models to predict the effluent concentrations. Our results show that SNN can predict plant performance with a correlation coefficient (R) up to 88%, 90%, 93%, and 96% for the single models COD, BOD₅, NO_3^{-} , and PO_4^{-3} , respectively, and up to 88%, 96%, and 93% for the ensemble models (BOD₅ and COD), (PO_4^{-3} and NO_3^{-}), and (COD, BOD₅, NO_3^{-} , PO_4^{-3}), respectively. The results also show that the two-hidden-layers model outperforms the one-hidden-layer model (SNN). Moreover, increasing the input parameters improves the performance of models with one and two hidden layers. We applied DNN (CNN, RNN, SAE) with three, four, and five hidden layers for WWTP modeling, but due to the small datasets, it gave a low performance and accuracy. In sum, this paper shows that SNN (one and two hidden layers) and the random forest (RF) machine learning technique provide effective modeling of the WWTP process and could be used in the WWTP management.

Keywords: shallow neural networks; deep neural networks; modeling; statistical analysis; wastewater treatment plant; random forest; quality prediction



Citation: Jafar, R.; Awad, A.; Jafar, K.; Shahrour, I. Predicting Effluent Quality in Full-Scale Wastewater Treatment Plants Using Shallow and Deep Artificial Neural Networks. *Sustainability* **2022**, *14*, 15598. <https://doi.org/10.3390/su142315598>

Academic Editor: Silvia Fiore

Received: 12 October 2022

Accepted: 14 November 2022

Published: 23 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The issue of wastewater disposal has become a worldwide major problem because of its high impact on health and the environment. Treatment plants have a major role in wastewater management; consequently, they should be managed in an optimal way [1,2]. The total system has to be regarded as an entity; therefore, to establish an adequate, statistical basis for evaluation of performance, it is required to examine and treat years of data [3]. ANN black-box models can be used to predict the WWTP performance using important pollution variables. Certain key parameters in a WWTP can be used to evaluate plant performance. These parameters could contain biological oxygen demand (BOD), chemical oxygen demand (COD), and suspended solid (SS) [4].

However, the strategy for monitoring and observing the influent and effluent of the plant requires an understanding of the plant's performance and the factors affecting the water specifications, such as time, season, and people's lifestyle. Measuring wastewater pollutants at the inlet and outlet of treatment plants enables managers to predict the quality of the water rejected into the water resource.

The operational control process of a biological wastewater treatment plant (WWTP) is complex due to changes in the composition of raw wastewater, different flow rates, and the complex nature of the treatment process [5]. In addition, the lack of permanent monitoring of the pollution variables reduces the effective control of the quality of the wastewater discharge. Conventional bioprocess modeling methods are based on equilibrium equations with rate equations for bacterial growth and substrate consumption [6], and micro-organisms grow by converting environmental nutrients into biomass, primarily proteins and other macromolecules. This conversion is accomplished through networks of biochemical reactions that span cellular functions such as metabolism, gene expression, transport, and signaling. Furthermore, because microbial reactions in conjunction with environmental interactions are nonlinear, time-variable, and complex, traditional deterministic and empirical modeling have limitations. Predicting plant operational parameters using traditional experimental techniques is also time consuming and a hindrance to the efficient control of such processes. Hong et al. [7] analyzed multidimensional process data and diagnosed the inter-relationship of process variables in a real activated sludge WWTP using an unsupervised neural network as an efficient tool for discovering complex dependencies between process variables and diagnosing the municipal WWTP's system behavior. Since microbial interactions were combined with environmental reactions, these equations are nonlinear, time-dependent, and of a crucial nature [8]. Côte et al. [9] developed a two-step procedure to improve the accuracy of a mechanistic model of the activated sludge process. First, they optimized the numerous model parameters using the downhill simplex technique to minimize the sum of the squares of the errors between the calculated and observed values of related parameters. Second, neural network models were successfully used to predict the optimized mechanistic model's remaining errors. Hamed et al. [10] developed two ANN-based models to predict BOD and SS effluent concentrations at a main WWTP in Cairo. Over a 10-month period, the developed models were trained and tested on daily sets of BOD and SS measurements. The BOD and SS models provided good estimates. The ANN models were a reliable prediction tool due to the prediction error, which varied slightly and smoothly across the range of data sizes used in training and testing. However, their data were limited, and additional parameters measured (e.g., pH, temperature, etc.) would improve the predictive capability of the neural network. Given all of this, predicting the operational parameters of the processing unit using traditional experimental methods is time-consuming. It constitutes an impediment to the effective, better management of the WWTP process.

Methods based on artificial intelligence technologies have been applied in many fields, especially environmental issues. The artificial neural network (ANN) method was employed to model the wastewater treatment plants (WWTP) in order to improve the prediction of treatment process [11]. Some key variables can be used to evaluate the performance of the plant, including biological oxygen demand (BOD), chemical oxygen demand (COD), phosphates (PO_4^{-3}), and nitrates (NO_3^{-}). The work in Ref. [12] showed that the ANN models can provide an effective tool for modeling the complex processes in the treatment units. Several other studies have employed ANN in this domain as well. For example, Hamoda et al. [5] evaluated the performance of the Al-Ardiya wastewater treatment plant in Kuwait City using the ANN backpropagations method. The results demonstrated that neural networks provide a flexible tool for modeling the wastewater treatment plants. Rene and Saidutta [13] predicted the BOD₅ and COD values of wastewater from a petrochemical factory treatment plant in India using multilayer neural networks. Vyas et al. [14] used the ANN approach to predict the parameters of cotreatment plants (sewage and industrial wastewater). Two models were constructed using ANN with a forward-feeding method-

ology (three layers) and a backpropagation algorithm to predict the BOD₅ concentration in and out of the Govindpura sewage treatment plant in Bhopal. They found that the plant's efficiency for removing BOD₅ is 80%. Jami et al. [15] used the multi-input ANN to predict the performance of a wastewater treatment plant. The highest correlation coefficient between the calculated and measured values was equal to $R = 0.6$ when predicting the COD pollution. Pakrou et al. [16] used the artificial neural networks to predict the treatment efficiency and the effect of input parameters on predicting the wastewater treatment plant in Tabriz. They concluded that the best model was obtained by combining the input variables of Q_{inf} , TSS_{eff} , and MLSS. It gave $R = 0.898$ and $RSME = 0.443$. Nourani et al. [17] studied the Nicosia WWTP performance by artificial intelligence using three variables: COD_{eff} , BOD_{5eff} , and TN_{eff} and three various artificial intelligence (AI) adopted nonlinear models, feed-forward neural network (FFNN), adaptive neuro-fuzzy inference system (ANFIS), support vector machine (SVM) methods, and a traditional multilinear regression (MLR) for WWTP performance forecasting. The results showed that the neural network ensemble (NNE) model was robust for predicting the WWTP performance.

Wang et al. [18] used machine learning (ML) methods to model waste water treatment plant processes in order to prevent the weaknesses of traditional mechanistic models. They presented an original ML context based on RF, DNN, variable importance measure (VIM), and partial dependence plot (PDP) to improve effluent quality control in WWTPs. The suggested ML framework seems to have the possible improvement of effluent-quality management approaches at Sweden's Umeå WWTP.

It is worth noting that data mining and knowledge recognition through machine learning (ML) have recently found use in environmental cleanup, particularly the investigation for the multifactorial method such as hexavalent chromium [Cr(VI)] elimination from industrial wastewater [19] and adsorption of the antibiotics as emerging component pollutants from the wastewater [20].

Alsulaili and Refaie [21] investigated the use of the ANN in forecasting the influent BOD₅ and the WWTPs' performance. The performance of the WWTPs were determined in relation to the effluent concentrations of COD, BOD₅, and TSS. The best forecasting model for the inlet BOD₅ achieved a value of $R = 0.87$.

Finally, WWTPs are difficult, nonlinear processes with great variations in stream rate, chemical environment, pollution load, and hydraulic situations. Modeling WWTP processes is challenging due to these complexities and uncertainties [18]. Indeed, deterministic models such as activated sludge models and other mechanistic models have been commonly applied for modeling WWTP methods and to forecast the comportment of specific parameters [22]. However, because many hypotheses and simplifications are required to make mechanistic models controllable and calculable, they have several restrictions. ASMs, for example, are only acceptable in specific alkalinity extents, pH, and temperature [18,23].

Many of these limitations are avoided by machine learning (ML) models, since they are specially focused on capturing relations concerning the input and output data that facilitate decisions and allow forecasts [24]. Analysis of the above research on the use of ANN in wastewater treatment plant modeling shows the following limitations: (i) the use of a low number of variables in the ANN models and (ii) the moderate performance achieved by these models ($R = 0.70 - 0.89$).

This work overcomes the aforementioned literature limitations by using a new ML-adopted framework, which is planned to predict WWTP effluent quality by explaining the relationships between the multi-influent auxiliary and effluent pollution variables.

We developed AI-based models such as the feed-forward neural network (FFNN) and random forest (RF) as well as deep learning methods such as the convolutional neural network (CNN), recurrent neural network (RNN), and pre-train stacked auto-encoder (SAE), which are relatively new methods for the assessment of WWTP methods with the goal of preventing the many shortcomings of traditional mechanistic models. Through the application of these developed models, we can decrease the number of laboratory measurements and shorten the time, effort, and cost. It takes five days to measure the

effluent BOD₅ in the laboratory, for example, while we can calculate its value by applying these models at any time and with a very high accuracy, thus reducing the cost of laboratory materials and the time required to conduct these experiments. All this ultimately serves to stabilize the ecological balance and reduce pollution. The paper presents, firstly, the models developed in this research, followed by the methodology and its application to the Khirbet al-Mu'azah full-scale wastewater treatment plant located in the southeast of Tartous city in Syria.

2. Theoretical Background

2.1. Artificial Neural Networks Theory

This study is based on creating multiple artificial neural networks (ANNs) models. These networks are defined as an inspired data-processing system that simulates how human data are processed, such as the biological nervous system and the human brain. The goal of the neural network is to calculate the output from the input values by certain inner calculations [25]. Feed-forward neural networks generally consist of a system of neurons organized in numerous layers, input layer, output layer, and at least one hidden layer, representing the second generation of neural networks or the shallow neural network (SNN). Each neuron in each layer is associated with each neuron in the next layer with an initial weight and then modified and adjusted during the training and learning process (Figure A1).

In our study, feed-forward neural networks (FFNNs) of multiple ANN models were developed. Forward feeding of networks means the spread of data entering the network in the forward direction, always from the input towards the output layer. This type of network is called the error backpropagation network, because the real output of the network is compared with the target output, and the difference between these values is called the error that the network propagates starting from the output layer [26]. To mathematically define the mechanism of error backpropagation, the mechanism of forward feeding must first be clarified, as demonstrated in the equations below [27].

The first stage in the feed-forward phase, in which the output Y_i^{k-1} of the neuron (i) in layer (k - 1) of the forward feeder network is associated with the input from the j neuron in the posterior layer k by a true weight factor (W_{ji}^k).

Where:

k: index of (k = I, II) layer;

i: neuron index of the (k - 1) layer;

j: neuron index of the (k) layer.

To compute the output Y_j^k , the neuron j of the k layer [k = I, II] performs the following calculation:

$$Y_j^k = f^k \left[\sum_{i=1}^N (W_{ji}^k \cdot Y_i^{k-1}) + b_i \right] \quad (1)$$

where:

N: the number of neurons in the k - 1 layer;

f^k : transfer function.

The bias (bi) vector is considered as the constant term in the polynomial mathematical equations that helps in solving these equations more easily and quickly.

The second stage is the error backpropagation step, in which the mean square error (MSE) and the error correction factor (δ) are determined in the output unit using the following equations:

$$MSE = err = \frac{1}{2 \cdot q} \sum_{i=1}^{i=q} (y_i - a_2)^2 \quad (2)$$

$$\delta = \frac{\partial err}{\partial a_{(2)}} \quad (3)$$

where:

err: error square rate in output unit; y : output target; (δ) : error correction factor; a_2 : calculated output.

The third stage is the updating weights phase, which involves updating the weights and bias factor as follows:

$$W_{(New)} = W_{old} + \Delta w \quad (4)$$

$$b_{(New)} = b_{old} + \Delta b \quad (5)$$

where: Δw : weight correction factor; Δb : bias correction factor.

There are two different methods (increment and batch input method) of updating the weights of an artificial neural network, assuming that the network inputs are in the form of a mathematical matrix consisting of rows and columns. Each row represents a vector that contains all the variables to be entered into the network [28].

The process of updating weights can be repeated thousands of times in familiar practical applications, and training usually stops when an acceptable error level is reached or when the number of iterations (epoch) specified by the trainer is reached.

2.2. Deep Neural Networks DNN

A deep neural network (DNN) is an artificial neural network (ANN) that contains numerous layers between the input and output layers, which are regular feed-forward networks in which data flows from the input to the output layer without returning back [29]. DNN can be thought of as an upgrade of SNN, which shows major improvement over SNN for its obvious enhancement of prediction accuracy on the unseen or testing dataset [30]. These layers are the input, hidden, and output, each of which is composed of several neurons; more than three layers (together with input and output layer) qualify as “deep” learning. Generally, when there is more than one hidden layer, a feed-forward ANN can be referred to as a deep neural network (DNN). The most essential theories in a FF-DNN are weights, biases, nonlinear activation, and backpropagation [18].

The DNN constructs a network of simulated neurons and assigns random numbers “weights”, to interconnections between them. Then, the weights and inputs are multiplied and return an output between 0 and 1. If the network failed to recognize a specific pattern accurately, an algorithm would modify the weights. The algorithm can increase the influence of specific parameters until it determines the correct mathematical treatment to fully process the data and turn the input into the output, whether it is a linear or a nonlinear relation.

Subsequently, deep learning has progressively become successful, and certain types of deep neural networks, such as convolution neural networks (CNN) and recurrence neural networks (RNN), have achieved surprising accomplishments in image, voice recognition, and natural language processing (NLP) [31]. Currently, deep learning has become the major flow in machine learning (ML). However, the implementations of DNNs in environmental issues are still restricted. This is mainly due to the problems associated with DNN training and collecting big datasets in wastewater treatment process science. DNNs for wastewater issues commonly have no more than 100 input variables (including characteristics, physical, chemical, organic, microbiology, processing, and property variables), and lower parameters need to be defined. Therefore, small DNNs (few hidden layers and a small neurons number in every layer) could be sufficient for almost all wastewater treatment plants issues.

The possibilities of using DNN with small datasets in wastewater are obvious: extensive regression obstacles already solved by conventional artificial intelligence (AI) such as SNN [32] with the small dataset can be treated by DNN with greater reliability and superior generalization accomplishment.

In this paper, we use pollution parameters prediction as a case to compare among SNN (traditional shallow NN), random forest (RF), and DNN (CNN, RNN, and SAE pre-train stacked auto-encoder DNN), and we show the performance of each method with a small dataset.

3. Materials and Methods

3.1. Plant Description and Data Used

The Khirbet al-Mu'azah wastewater treatment plant is located southeast of Tartous city in Syria. It is placed beside the main Tartous–Safita road, about 17 km from Tartous and 13 km from Safita city. Figure 1 shows the schematic of the WWTP process.

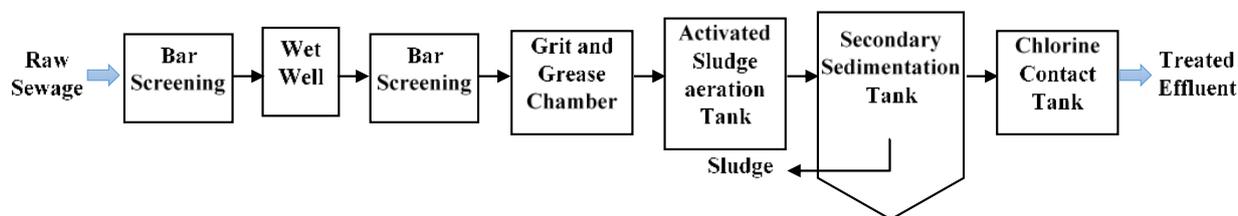


Figure 1. Schematic of the WWTP process.

The Khirbet al-Mu'azah treatment plant is based on the activated sludge treatment with the extended aeration technique. It was planned to serve a group of villages with 10,000 people. The average inflow is 42 m³/h.

This research used measurements of samples taken from the influent and effluent of the Khirbet al-Mu'azah wastewater treatment plant for 2018–2020. These measurements of the selected variables cover all the seasonal variations. Moreover, it includes several sets of input and output parameters. The database consists of 198 domestic wastewater samples taken from the inlet and the same number from the outlet over three years; eight pollution inlet and outlet variables were measured. This database was utilized to build the models, including the influent of

COD_{inf}, biochemical oxygen demand BOD_{inf}, phosphates PO_{4inf}, nitrates NO_{3inf}, temperature T_{inf}, degree of acidity or alkalinity pH_{inf}, electric conductivity EC_{inf} and the total dissolved solids TDS_{inf}, which were used as the inputs of models, and the effluent measurements of the same parameters COD_{eff}, BOD_{eff}, PO_{4eff}, NO_{3eff}, T_{eff}, pH_{eff}, EC_{eff}, and TDS_{eff}, which were used as targets of the models.

In the WWTP, the COD/BOD ratio is normal (equal to 2), indicating that a considerable portion of organic matter will easily reduce biologically. The measurements of BOD₅ showed that the pollution's load is generally originated from the households, with only a minor involvement from the industrial area. In addition, the variations and components of the WWTP are involved by the quantity of domestic organic waste.

Figure 2a illustrates descriptive statistics for the selected treated parameters as a boxplot. Figure 2b presents the observed influent (inf) and treated effluent (eff). Figure 2c illustrates the concentrations of COD, BOD₅, NO₃, PO₄, T, pH, EC, and TDS at the entrance and outlet of the plant.

3.2. Models Development

The creation of the artificial neural networks model depends mainly on the available databases of the studied phenomenon factors. Therefore, the data of these factors (inputs and outputs) that we collected through the research period were statistically examined by the one-way variance of the ANOVA1 method (Figure A2a,b), which is included in the environment work of the MATLAB software.

This ANOVA1 analysis was applied before developing ANN models to reject and exclude the raw data with anomalous and inaccurate values in the database. The plot signifies the degree of anomalies, effectiveness, and the range of each treatment plant variable. After completing the statistical analysis phase, artificial neural network models were created using Matlab, which provides an important platform for applying the ANN modeling and simulation process. The software includes a special toolbox that contains several functions that help manage and analyze historical data.

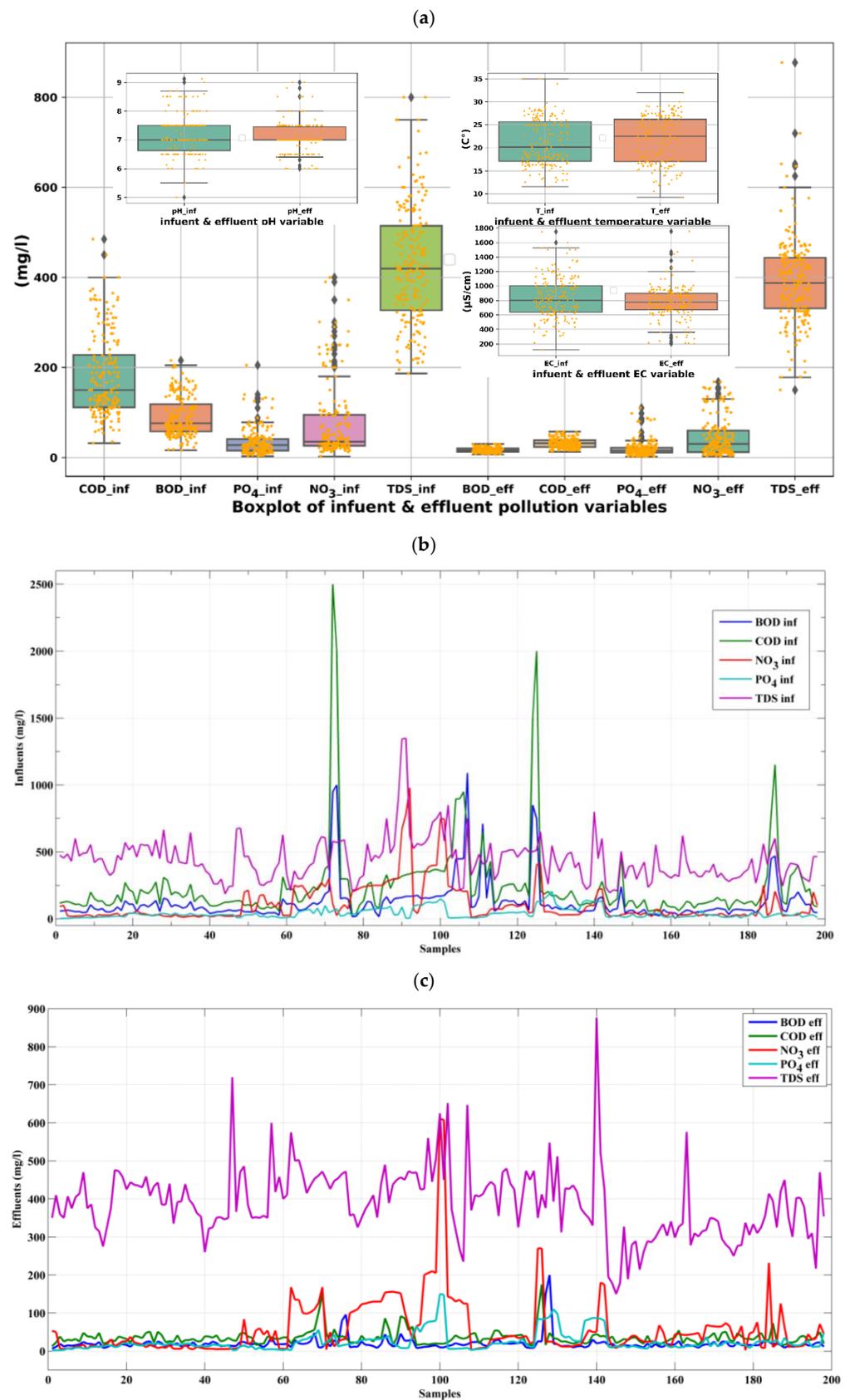


Figure 2. (a) Boxplot of treated influent and effluent parameters The orange dots represent the data points. (b) BOD₅, COD, NO₃, PO₄, and TDS concentrations of the WWTP at the inlet (influent) and (c) outlet (effluent).

These ANOVA1 graphs summarize each variable through the four elements as follows: the centerline in each box indicates the sample mean, which refers to the central tendency or location; square box to represent the variance around this central tendency (the box's edges reflect the 25th and 75th percentiles); whiskers around a box to represent the variable range. All measurements that exceeded the filament longitudinal length are marked with a sign (+) if their value is greater than 1.5 times the range quarterly away from the top or bottom of the box.

Related to the dataset division, we start with 80% of the data in the training set, 10% in the validation set, and 10% in the testing set, and then we continue with 70% of the data in the training set, 15% in the validation set, and 15% in the testing set that we adopted. The optimum split of the training, validation, and testing set depends upon factors such as the use case, the model structure, and data dimension, etc.

Two artificial neural network main scenarios (S1 and S2) were developed. According to the regular rules in the neural network process, input variables and target variables must be normalized before use in the network [33]. Thus, at the primary phase before the training of the model, input and output data were standardized (e.g., in the range of 0 and 1) as: [34,35]

$$x_i = \frac{x_u - x_{(min)}}{x_{(max)} - x_{(min)}} \quad (6)$$

where: X_i is the standardized data value, x_u is observed data, $x_{(min)}$ is the minimum, and $x_{(max)}$ is the maximum value of the measured dataset. The statistical analysis of the input-output variables is vital in artificial neural modeling because this type of analysis determines the nature and strength of the relationships between inputs and outputs.

It is obvious that for all types of data-based methods (e.g., artificial intelligence methods), if the amount of dispersion (standard deviation) of data is low (indicating the closeness of the data to the mean), lower biased outputs from the models are predictable. The correlation coefficient (R; a widely used measure) was calculated in the descriptive statistical analysis to determine the force and amount of linear relationship between two variables, which can be used as an initial indication of a potential linear relationship among a group of parameters. Table 1 presents the results of the Pearson correlation matrix between influent and effluent parameters.

Nevertheless, the drawback of the calculated R coefficient demonstrates that the use of traditional linear methods to process complex nonlinear relations is not preferred, and there is a significant requirement to add additional nonlinear solid techniques.

As a result, unlike previous studies that used linear correlation coefficients between input and output parameters to select the dominant inputs of nonlinear models, this study examines different combinations of input parameters using the ANN method.

The main shallow artificial neural network scenarios aimed to achieve optimum performance using the NNFTool box library included in the Matlab environment and the Levenberg–Marquardt (LM) network-training algorithm. The first ANN scenario (S1) consists of the input layer, one hidden layer, and the output layer. The Tansig function was used as a transfer function for the hidden layer, and the linear transfer function was used as the transfer function for the output layer. As for the second scenario (S2), it includes an input layer, two hidden layers, and an output layer, where the sigmoid function was used as a transfer function for the first hidden layer, the Tansig function as a transfer function for the second hidden layer, and the linear function as a transfer function for the output layer. The mean square error (MSE) and the (R) correlation coefficient were used to assess the network effectiveness in the two scenarios.

Table 1. Pearson correlation matrix between influent and effluent parameters.

Parameters	COD_inf	BOD_inf	PO ₄ _inf	NO ₃ _inf	T_inf	pH_inf	EC_inf	TDS_inf	COD_eff	BOD_eff	PO ₄ _eff	NO ₃ _eff	T_eff	pH_eff	EC_eff	TDS_eff
COD_inf	1															
BOD_inf	0.901 **	1														
PO ₄ _inf	0.130	0.069	1													
NO ₃ _inf	0.194 **	0.169 *	0.346 **	1												
T_inf	0.142 *	0.128	−0.180 *	−0.232 **	1											
pH_inf	0.031	−0.005	−0.080	−0.259 **	−0.130	1										
EC_inf	0.251 **	0.270 **	0.235 **	0.536 **	−0.030	−0.316 **	1									
TDS_inf	0.248 **	0.269 **	0.205 **	0.527 **	−0.055	−0.260 **	0.952 **	1								
COD_eff	0.128	0.105	0.167 *	0.287 **	0.040	−0.070	0.296 **	0.295 **	1							
BOD_eff	0.016	0.015	0.306 **	0.030	0.120	0.034	0.010	0.020	0.187 **	1						
PO ₄ _eff	0.130	0.076	0.888 **	0.396 **	−0.138	−0.052	0.214 **	0.180 *	0.087	0.207 **	1					
NO ₃ _eff	0.196 **	0.168 *	0.457 **	0.774 **	−0.231 **	−0.138	0.350 **	0.344 **	0.174 *	−0.050	0.599 **	1				
T_eff	0.102	0.099	−0.162 *	−0.203 **	0.892 **	−0.178 *	−0.059	−0.095	0.072	0.151 *	−0.129	−0.220 **	1			
pH_eff	0.053	0.004	0.047	0.002	−0.037	0.635 **	−0.012	0.013	0.114	0.129	0.029	0.008	−0.048	1		
EC_eff	0.139	0.217 **	0.329 **	0.332 **	0.141 *	−0.284 **	0.631 **	0.575 **	0.123	0.148 *	0.299 **	0.266 **	0.148 *	0.008	1	
TDS_eff	0.137	0.203 **	0.314 **	0.327 **	0.146 *	−0.268 **	0.616 **	0.581 **	0.124	0.161 *	0.287 **	0.260 **	0.157 *	−0.010	0.949 **	1

** . Correlation is significant at the 0.01 level (2-tailed). * . Correlation is significant at the 0.05 level (2-tailed).

In addition, we applied random forest (RF), which is one of the traditional machine learning (ML) techniques. It can perform both regression and classification problems. It is a supervised type of ML applied in pattern recognition. The random forest (RF) regression is an ensemble machine learning algorithm that combines multiple decision trees and that was first developed by Breiman [36]. A regression tree is a set of conditions or restrictions that are organized hierarchically and applied sequentially from the tree's root to its leaf. RF is based on the assumption that different independent predictors predict incorrectly in different areas, and that by combining the prediction results of the independent predictors, the overall prediction accuracy can be improved. When the training data vary slightly, the structures of regression trees in RF show significant differences. Independent predictors can be created by combining this characteristic with bagging (bootstrap aggregation) and random feature selection to construct a random decision tree.

The RF starts with a large number of bootstrap samples taken at random from the original training dataset. Each bootstrap sample is fitted with a regression tree. For binary segmentation, a small set of input variables chosen at random from the total set is considered for each node per tree. In this study, the random forest regression model was imported from the Sklearn package as "sklearn.ensemble.RandomForestRegressor". The RF algorithm needs to define the number of random features, trees, and stop criteria. Averaging over all trees gives the predicted value of an observation. The number of regression trees ('ntree'; default value is 100 trees) and the number of input variables per node ('mtry'; equate to 1) should be optimized in the RF. When "mtry" equates to one, the split variable is completely random, so all variables get a chance. Given the set of training input–output pairs, the RF regression model was used to model the relationship between the WWTP influent auxiliary pollution parameters and effluent parameters. RF modeling generates training data by sampling and replacing all of the samples for each predictor in the ensemble [36]. We stopped training when the minimum sample in a tree was one sample with a minimum impurity of zero.

Convolutional neural network (CNN), which is a subset of machine learning that uses neural networks with at least three layers, has evolved into one of the most prominent neural networks in the field of deep learning. CNN is a type of feedforward neural network that uses convolution structures to extract features from data, unlike traditional feature extraction methods [37]. CNN needs a convolutional layer but can also include nonlinear, pooling, and fully connected layers to form a deep convolutional neural network [38]. CNN can be useful depending on the application. However, it adds new parameters for training. Convolutional filters are trained in the CNN using the backpropagation method. In the convolutional layer, multiple filters slide over the layer for the given input data. The output of this layer is a sum of an element-by-element multiplication of the filters and receptive field of the input. The weighted summation is added as an element to the following layer.

In this study, related to the COD and BOD₅, their CNN models were imported from the Keras package as "keras.models import Sequential and keras.layers import Dense, Conv1D, Flatten". Sequential is the easiest way to build a model in Keras. It allows building a model layer by layer. Our first layer was Conv1D layer. This is a convolution layer that deals with the input variables, which is seen as a 1-dimensional matrix. We used 32 as the number of nodes in the layer. This number can be adjusted to be higher or lower, depending on the size of the dataset. In our case, 32 worked well. Kernel size is the size of the filter matrix for our convolution. Therefore, a kernel size of 2 means we would have a 2 × 2 filter matrix. We used the ReLU activation function for this and (Dense) layers. This activation function has been proven to work well in neural networks. Our first layer also took in an input shape as 7 input and 1 output. In between the Conv1D layer and the first dense layer, there was a "Flatten" layer. Flatten serves as a connection between the convolution and dense layers. "Dense" is the layer type we used for our third and output layer. Dense is a standard layer type that is used in many cases for neural networks. First dense layer includes 64 nodes and ReLU activation function, while the output dense layer includes one output COD or BOD₅ according to the studied model. Regarding the model compiling, we

used the “Adam” optimizer to control the learning rate. The Adam optimizer adjusts the learning rate throughout training. The learning rate defines in what way and how fast the optimal weights for the model are calculated. A smaller learning rate may drive additional accurate weights (to a certain extent); however, the time required to compute the weights will be longer. We used “MSE” for the loss function; this is the choice for regression issues, a lower value that indicates that the model is performing better. The most widely used evaluation metrics R-squared (R^2) is the proportion of variation in the outcome that is explained by the predictor variables in a regression model, and mean squared error (MSE) is the average error performed by the model in predicting the outcome for an observation.

A recurrent neural network (RNN) is a type of artificial neural network that is most commonly used in speech recognition and natural language processing (NLP). RNN is also used in deep learning and the creation of models that simulate the activity of neurons in the human brain. A recurrent neural network (RNN) is a type of neural network in which the previous step’s output serves as the input to the current step. While all inputs and outputs in traditional neural networks are independent of each other, in some prediction cases, such as predicting the next word of a sentence, knowing the previous words is required, and thus the previous words must be remembered. This is why RNN was created, which solved the problem by using a hidden layer. The hidden state, which remembers some information about the sequence, distinguishes and elevates an RNN [39]. Except for the addition of a hidden layer, a recurrent neural network is similar to a traditional neural network (the memory state to neurons). A simple memory will be included in the computation. A recurrent neural network is a deep learning algorithm that uses a sequential approach. All inputs and outputs in neural networks are always dependent on all other layers. These neural networks perform mathematical calculations sequentially, which is why they are referred to as recurrent neural networks [40].

Similarly (up to a certain point) to the above CNN models, we constructed the RNN model, “keras.models import Sequential and keras.layers import Dense, SampleRNN”, imported from the Keras package. In Keras, the simplest way to build a model is sequential. RNN models have used 1 layer of SampleRNN with 40 neurons, input shape as 7 input, 1 output, and ReLU as activation function. Followed by four layers of Dense type, they contain 30, 10, 5, 1 neurons, respectively, the last dense layer produces the output of the model (COD or BOD₅) and ReLU activation function for all. Concerning the model compiling, we used the “Adam” optimizer to control the learning rate and “MSE” for the loss function. Finally, we used the R-squared (R^2) and MSE for evaluation of the performance of our models.

Pre-training sets up DNN with optimized weights and biases values that are close to the global optimal solution, allowing the subsequent fine-tuning step to avoid the traps of the local optimal solution. It is possible to initialize DNN through stacked auto-encoder (SAE); the auto-encoder is a one-hidden-layer shallow neural network with the same input and output layers. Each auto-encoder has the same number of neurons as the corresponding layer of the DNN. The first auto-encoder receives the DNN’s input and output, and the output of its hidden layer provides the input and output to the second auto-encoder; for example, the output of the previous auto-hidden encoder’s layer provides the input and output to the next auto-encoder. Each trained auto-encoder gives initial weights and biases values to the DNN’s corresponding layer [30].

Moreover, we used RF, CNN, RNN, and SAE pre-train stacked auto-encoder as a DNNs method in Python and Matlab environment to determine its accuracy and performance of COD and BOD₅ models.

4. Results and Discussion

4.1. Single Model (COD Effluent)

We assessed and compared the forecasting effectiveness of the artificial intelligence method on the chemical oxygen demand elimination performance of the full-scale WWTP, after the evaluation data dependency (Figure 3a). The scatter matrix of the dependent

variable (COD_{eff}) and the other seven variables is depicted in Figure 3a. The Pearson correlation coefficient (R) of COD_{eff} as well as the other seven variables are represented in the same figure. The scatter plot can easily reveal any obvious patterns or linear relationships between them. They demonstrate that no noticeable patterns existed in the scatter matrix. The absolute value of the correlation coefficient (R) is well below 0.5, indicating that there is no linear relationship between (COD_{eff}) and the other seven variables; the relationship is nonlinear and cannot be expressed by a simple function. Several artificial neural networks have been constructed in the two main scenarios (S1: one hidden layer, and S2: two hidden layers). Both include one input layer consisting of five subscenarios for the input variables, considering various input combinations (Table 2): (4 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf}), (5 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf}), (6 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf} , PO_{4inf}), (7 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf} , PO_{4inf} , BOD_{5inf}), (8 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf} , PO_{4inf} , BOD_{5inf} , COD_{inf}); all the input values were taken at the WWTP entrance. The output layer was limited to a single neuron, and it represents the values of the COD_{eff} at the outlet of the plant. As for the hidden layer in the two main scenarios, it is worth noting that defining the number of hidden neurons, training epochs, and transfer functions are important elements in planning the FFNN model. Because of its quick learning and high performance precision, Lavenberg–Marquardt was selected and used as the BP training algorithm in this research. It was determined using the experiment (trial) method, in which the number of neurons and sample group division were adjusted to obtain the lowest mean square error for different repetitive epochs.

Table 2. Performance of the COD_{eff} shallow artificial neural network models.

Model No.	Model Input Variables	Model Output Variable(S)	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden Layers	MSE
M1-S1 M1-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf}	COD_{eff}	0.557 0.6637	0.37 0.317	0.49 0.6104	0.504 0.605	67 60-40	137.42 121.26
M2-S1 M2-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$	COD_{eff}	0.65 0.6859	0.703 0.6838	0.806 0.6023	0.676 0.671	65 55-30	39.57 67.34
M3-S1 M3-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$, $PO_4^{-3}_{inf}$	COD_{eff}	0.605 0.696	0.71 0.804	0.393 0.769	0.59 0.719	65 60-40	169.21 46.61
M4-S1 M4-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$, $PO_4^{-3}_{inf}$, BOD_{5inf}	COD_{eff}	0.79 0.891	0.752 0.912	0.872 0.833	0.798 0.888	55 50-30	48.937 8.5135
M5-S1 M5-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$, $PO_4^{-3}_{inf}$, BOD_{5inf} , COD_{inf}	COD_{eff}	0.79 0.826	0.67 0.78	0.658 0.895	0.754 0.841	60 50-30	44.963 40.09

Each subscenario arrangement was trained more than 100 times with different random seeds before selecting the best SNN.

Matlab was used for all calculations, as well as its statistics, ML, and ANN toolboxes. Pearson correlation coefficients of target values and SNN prediction values were investigated as a training, validation, and testing accuracy metric. Table 2 shows the best performance of COD_{eff} models selected from several studied architectures. Figure 3b,c represent the best architecture of the two scenarios (M4-S1 and M4-S2, respectively) through a trial–error process and the correlation coefficients for all training, validation, and testing groups, in addition to the set of all the data. The FFNN model that includes seven inputs and one output neurons was discovered as the best for all simulated parameters, as shown in Table 2 and illustrated in Figure 3d for the first scenario (M4-S1) and Figure 3e for the second scenario (M4-S2). Figure 3f shows the results' predicted values by ANN as the samples' sequence plots for COD_{eff} compared to the measured values. It could be observed that the results of FFNN are acceptable for forecasting the performance of the WWTP (Table 2).

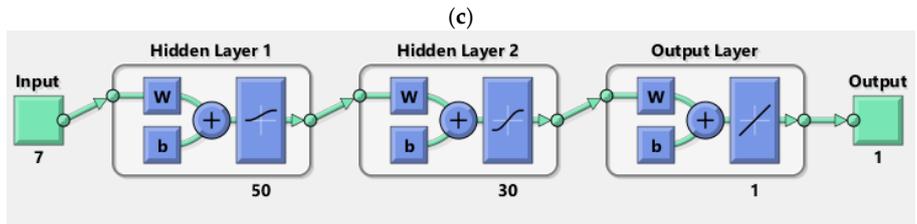
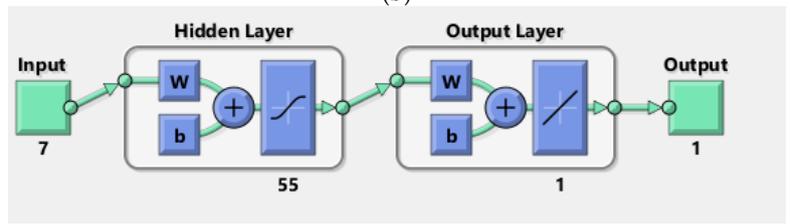
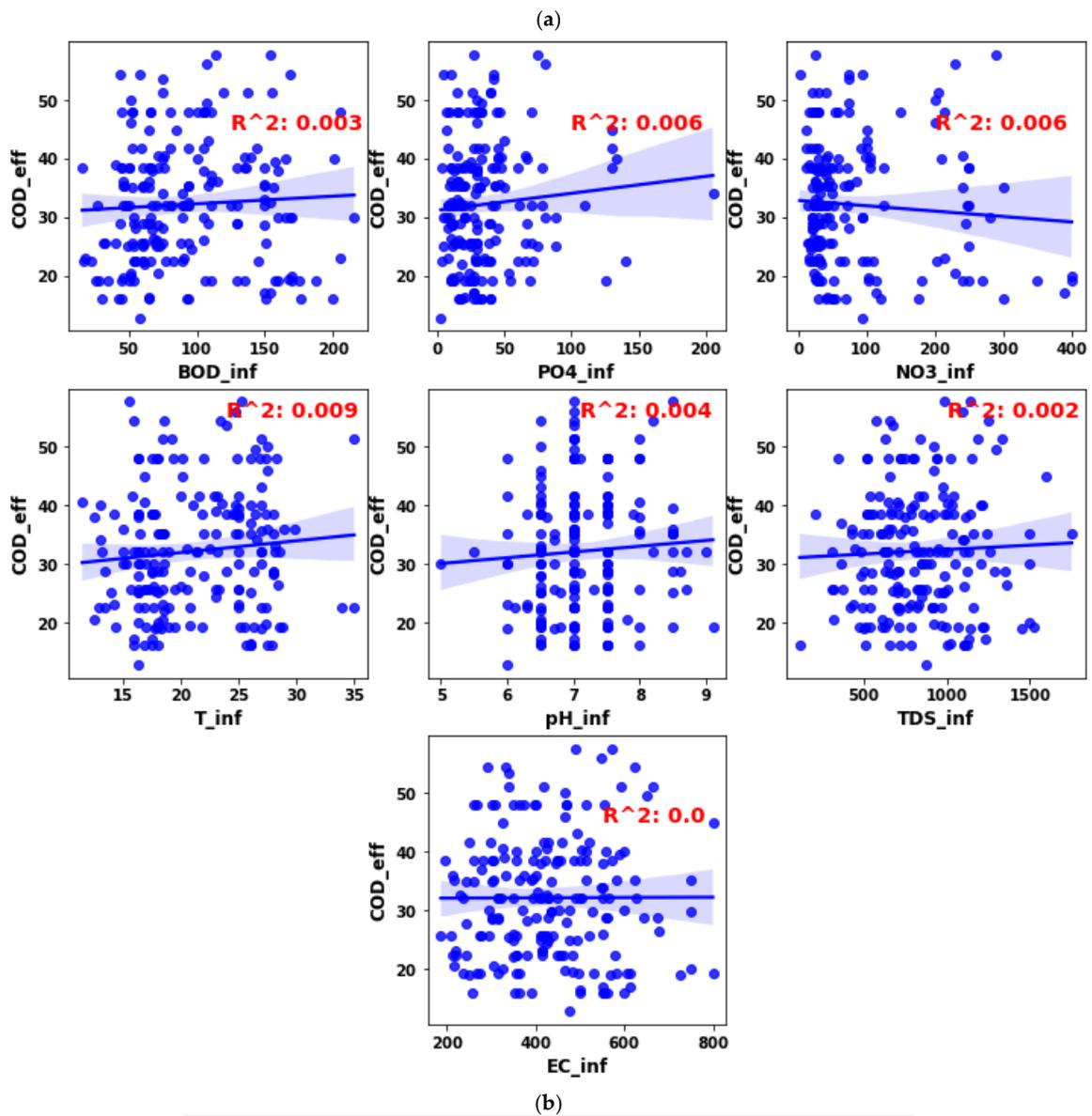


Figure 3. Cont.

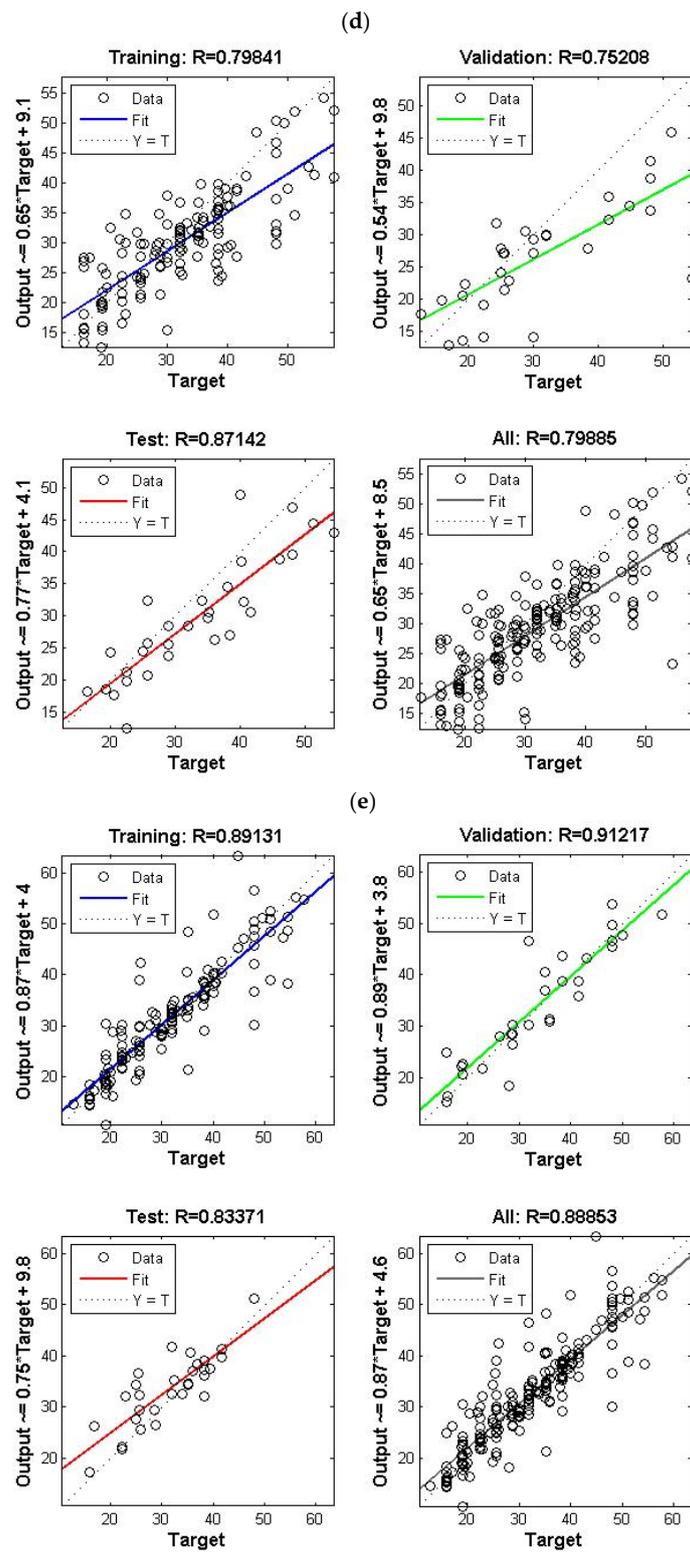


Figure 3. Cont.

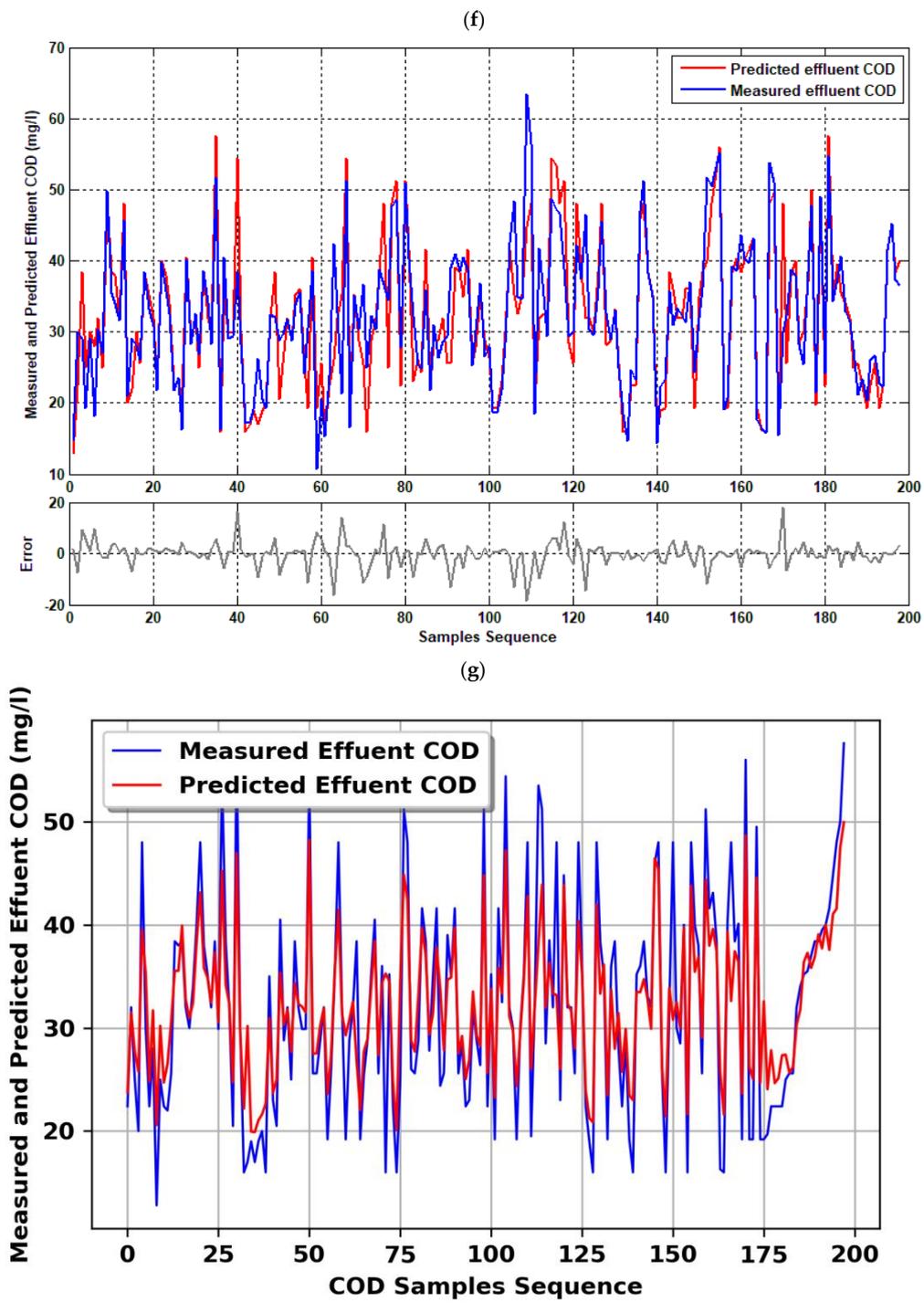


Figure 3. Cont.

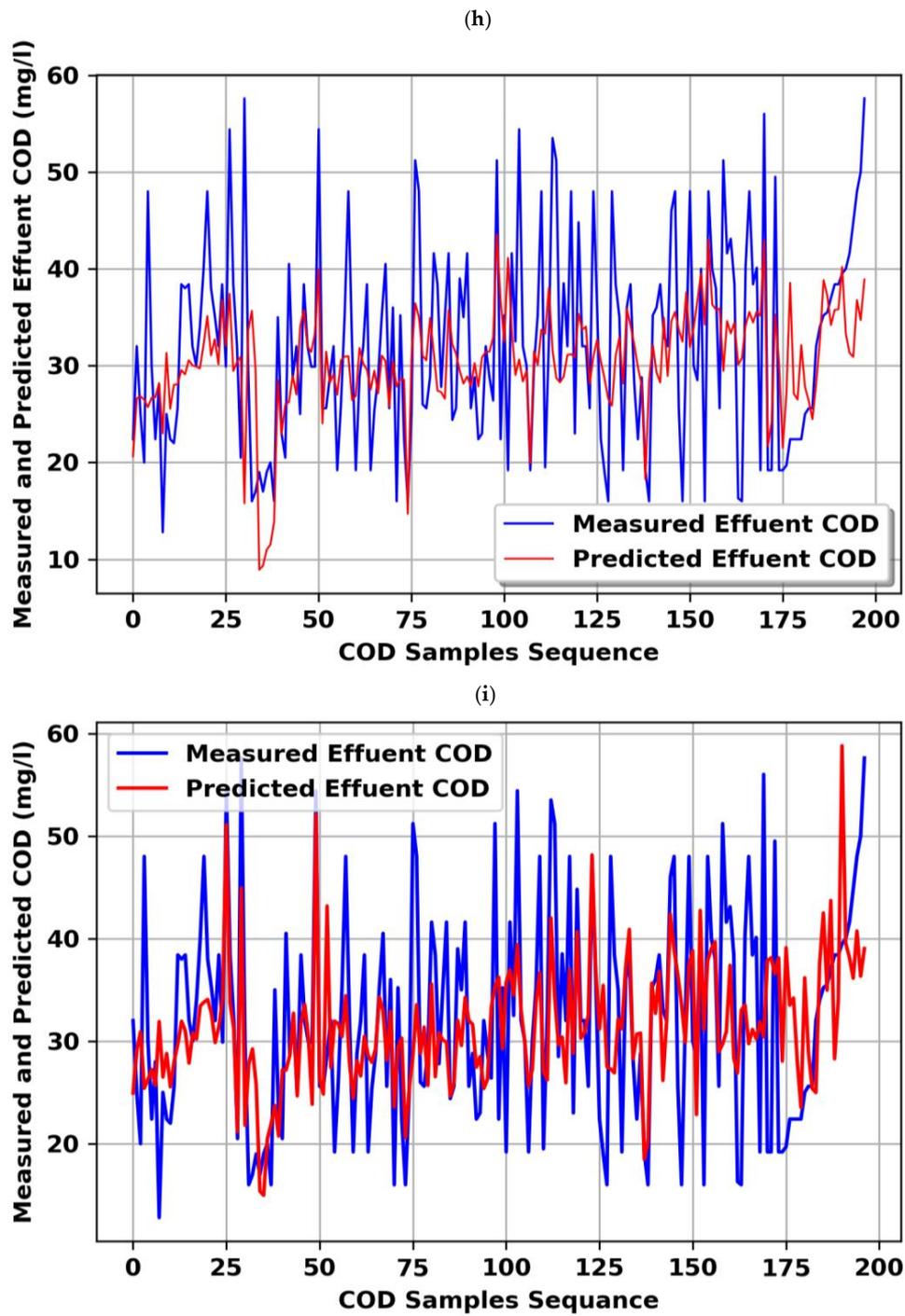


Figure 3. Cont.

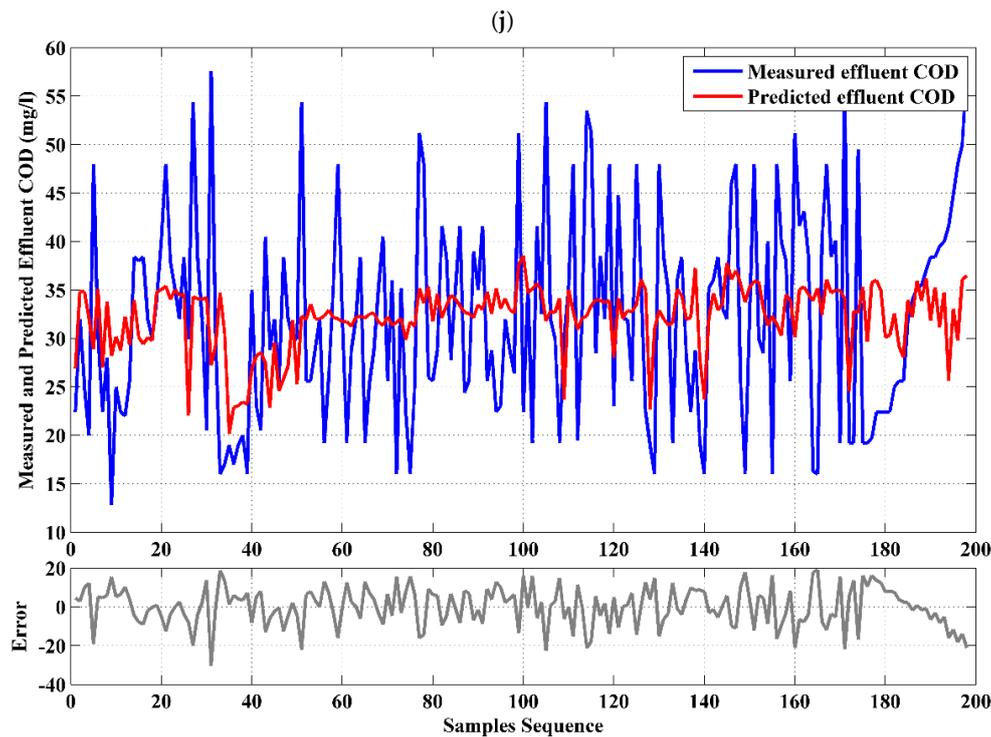


Figure 3. (a) Scatter matrix of (COD_{eff}) and other seven input variables, the values of Pearson correlation coefficients (R_s) are also shown in the figure. (b) The best artificial neural network architecture for Model M4-S1 in the first scenario. (c) The best artificial neural network architecture for Model M4-S2 in the second scenario. (d) The values of correlation coefficient (R) in all stages of Model M4-S1 in the first scenario. (e) The values of correlation coefficient (R) in all stages of Model M4-S2 in the second scenario. (f) Measured vs. predicted samples sequence obtained by best single COD effluent models M4-S2. (g). Measured vs. predicted samples sequence achieved by best single COD effluent RF models M4-S2. (h). Measured vs. predicted samples sequence achieved by best single COD effluent CNN models M4-S2. (i). Measured vs. predicted samples sequence achieved by best single COD effluent RNN models M4-S2. (j). Measured vs. predicted samples sequence achieved by best single COD effluent SAE models M4-S2.

The aim of comparing the two-hidden-layer models to one-hidden-layer models was to integrate a set of models, decrease the disadvantages of every individual model (one hidden layer), and build an enhanced and more reliable model with high accuracy using two hidden layers.

The results of random forest (RF) of the COD model were presented in Figure 3g, with R equal to 0.933 and MSE equal to 24.3.

We applied the deep neural network (DNN) with the (LBFGS) solver to the data of the best model within the first and second scenarios (M4-S2). It consists of three hidden layers according to the following neurons architecture: 50 30 10. It gave slightly better results as it achieved an R value equal to 0.929, but the MSE value was 54.71.

Figure 3h shows the results achieved by the CNN model, with R equal to 0.42 and MSE equal to 152.81.

Figure 3i presents the result of the RNN model, with R equal to 0.51 and MSE equal to 62.16.

We also tried the stacked auto-encoder technique within three hidden layers which make up fully connected deep neural network of the 7-(4-3-2)-1 and 7-(5-4-3)-1 structure, four hidden layers deep neural networks of the 7-(5-4-3-3)-1 and 7-(6-5-4-3)-1 structure, and five hidden layers deep neural networks of the 7-(6-5-4-3-3)-1 and 7-(5-4-3-3-3)-1 structure. Figure 3j illustrates the best results achieved by four hidden DNNs layers with 7-(6-5-4-3)-1 neurons architecture. We have a 0.34 value for the (R) correlation coefficient and MSE equal to 0.174.

4.2. Single Model (BOD_5 Effluent)

We similarly built several artificial neural networks in the two main scenarios that include a single input layer consisting of five subscenarios for the input variables: (4, 5, 6, 7, 8 neurons), as shown in Table A1 (Appendix A). The input values were all taken at the WWTP inlet, while the output layer was limited to a single neuron representing the BOD_{5eff} value at the plant's outlet. Table A1 displays the best performance of BOD_{5eff} models selected from several studied architectures, as well as the correlation coefficients for all training, validation, and testing groups, in addition to the set of the entire data for the two scenarios (one and two hidden layers). The best model performance is illustrated in Figure A3a,b. Measured versus predicted BOD_{5eff} values are described in Figure A3.

We have also applied the deep neural network (DNN) with the (LBFGS) solver to the data of the best BOD_5 effluent model (M9-S2). It consists of three hidden layers according to the following neurons architecture: 24 30 10. It gave much lower results as it achieved an R value equal to 0.294, and the MSE value was 322.

The results of applied random forest (RF) of the BOD_5 model were presented in Figure A3d, with R equal to 0.94 and MSE equal to 6.12.

Figure A3e shows the results achieved by the CNN model, with R equal to 0.53 and MSE equal to 20.55.

Figure A3f presents the result of the RNN model, with R equal to 0.32 and MSE equal to 20.26.

We tried the stacked auto-encoder technique within three hidden layers fully connected to the deep neural network BOD_5 of 7-(4-3-2)-1 and 7-(5-4-3)-1 structure, four hidden layers deep neural networks of 7-(5-4-3-3)-1 and 7-(6-5-4-3)-1 structure, and five hidden layers deep neural networks of 7-(6-5-4-3-3)-1 and 7-(5-4-3-3-3)-1 structure. Figure A3g illustrates the best results achieved by the four hidden DNNs layers with 7-(6-5-4-3)-1 neurons architecture. We have 0.39 value for the (R) correlation coefficient and MSE equal to 0.176.

According to these results of applying three, four and five hidden layers in the previous models (COD effluent and BOD_5 effluent), we were satisfied with two hidden layers for all subsequent models: ($PO_4^{-3} eff$) model, (NO_{3eff}) model, (COD_{eff} and BOD_{5eff}) ensemble model, ($PO_4^{-3} eff$ and NO_{3eff}) ensemble model, and (COD_{eff}, BOD_{5eff} , $PO_4^{-3} eff$, NO_{3eff}) ensemble model.

4.3. Single Model (PO_4^{-3} Effluent)

We constructed several artificial neural networks that include a single input layer consisting of five subscenarios for the input variables: (4, 5, 6, 7, 8 neurons), as shown in Table 3. All the input values were taken at the WWTP inlet, while the output layer was limited to a single neuron representing the value of the $PO_4^{-3} eff$ at the outlet of the plant. Table 3 shows the best performance of the $PO_4^{-3} eff$ models selected from several studied architectures, as well as the correlation coefficients for all training, validation, and test groups, in addition to the set of all data, for the two scenarios (one and two hidden layers). The best model performance is illustrated in Figure 4a,b. Measured versus predicted $PO_4^{-3} eff$ values are presented in Figure 4c.

Table 3. Performance of the $PO_4^{-3} eff$ artificial neural network models.

Model No.	Network Input	Network Output	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden Layers	MSE
M11-S1 M11-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf}	$PO_4^{-3} eff$	0.676 0.681	0.027 0.948	0.021 90.89	0.223 0.930	67 0.81	2999.55 40–60
M12-S1 M12-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^- inf$	$PO_4^{-3} eff$	0.968 0.572	0.136 0.670	0.0526 0.625	0.391 0.58	65 30–55	2833.67 59.22
M13-S1 M13-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^- inf inf$, $PO_4^{-3} inf$	$PO_4^{-3} eff$	0.97 0.976	0.658 0.622	0.569 0.97	0.832 0.963	65 40–60	186.127 10.26
M14-S1 M14-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^- inf$, $PO_4^{-3} inf$, BOD_{5inf}	$PO_4^{-3} eff$	0.869 0.953	0.162 0.939	0.0836 0.989	0.65 0.958	55 30–50	278.46 48.34
M15-S1 M15-S2	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^- inf$, $PO_4^{-3} inf$, BOD_{5inf} , COD_{inf}	$PO_4^{-3} eff$	0.904 0.933	0.789 0.959	0.23 0.937	0.76 0.936	60 30–50	256.2 25.53

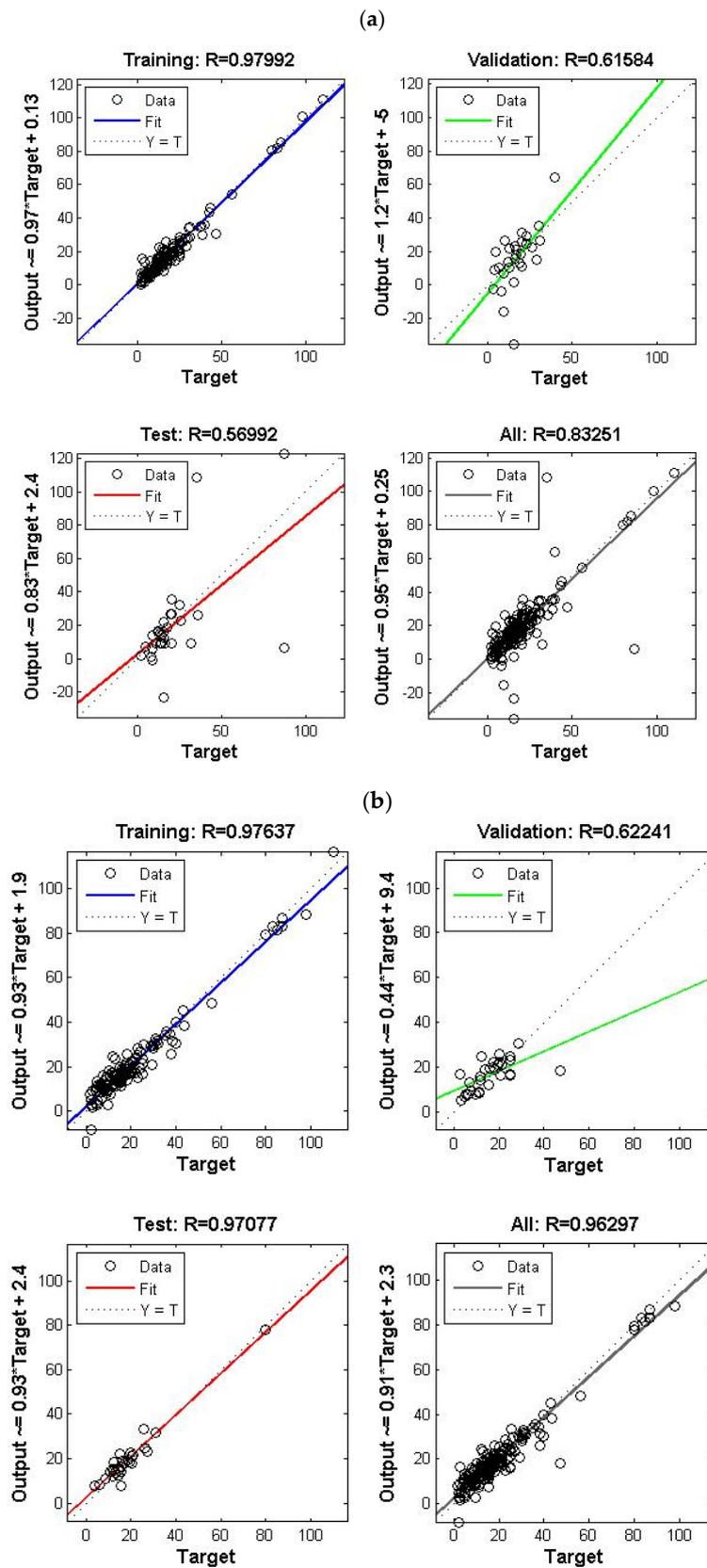


Figure 4. Cont.

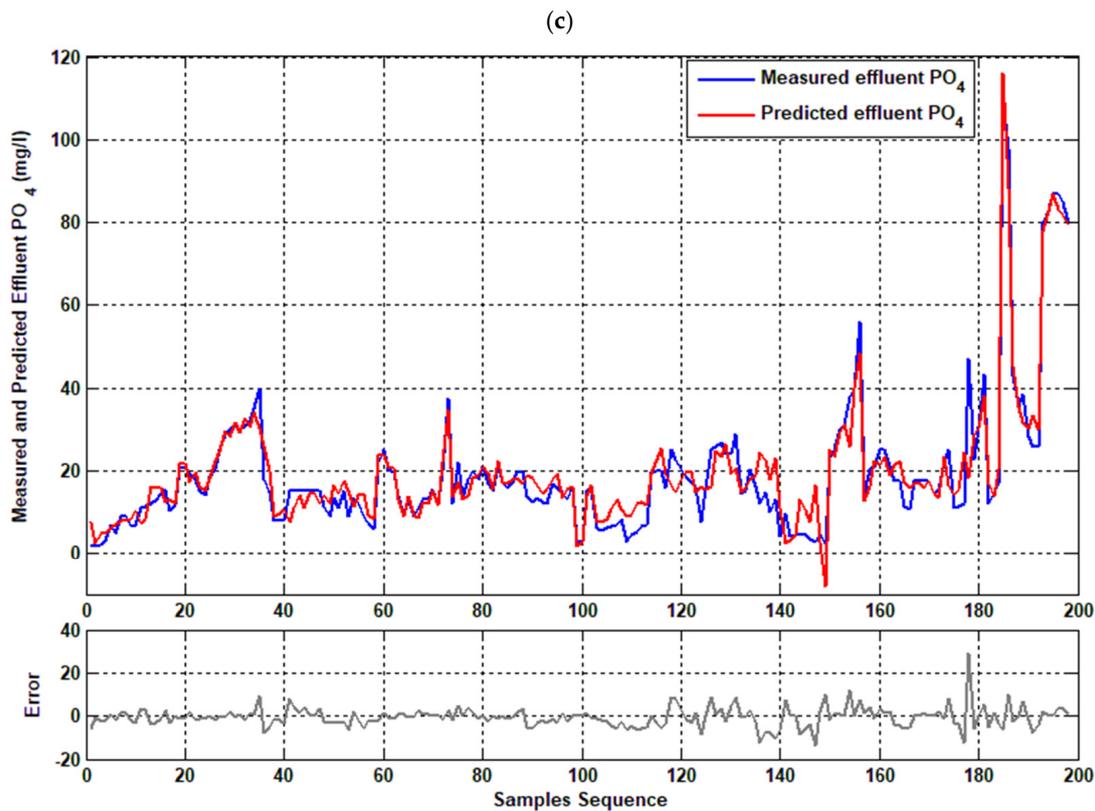


Figure 4. (a) The values of correlation coefficient (R) in all stages of Model M13-S1 in the first scenario. (b) The values of correlation coefficient (R) in all stages of Model M13-S2 in the second scenario. (c) Observed vs. predicted samples sequence obtained by best single PO_4^{-3} effluent models M13-S2.

4.4. Single Model (NO_3^- Effluent)

We built numerous artificial neural networks that included a single input layer containing five subscenarios for the input variables: (4, 5, 6, 7, 8 neurons), as shown in Table A2. The input values were all taken at the WWTP inlet, while the output layer was restricted to a single neuron representing the value of NO_3^- effluent at the plant's outlet. Table A2 shows the best-performance NO_3^- eff models chosen from several studied architectures, as well as the correlation coefficients for all training, validation, and testing groups, in addition to the set of all the data for the two scenarios (one and two hidden layers). The best model performance is illustrated in Figure A4a,b. Observed versus predicted NO_3^- eff values are shown in Figure A4c.

4.5. Ensemble Model (COD and BOD_5 Effluent)

We established several artificial neural networks that included a single input layer consisting of five subscenarios: (4 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf}), (5 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf}), (6 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf} , PO_{4inf}), (7 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf} , PO_{4inf} , BOD_{5inf}), (8 neurons: T_{inf} , pH_{inf} , TDS_{inf} , EC_{inf} , NO_{3inf} , PO_{4inf} , BOD_{5inf} , COD_{inf}). All the input values were taken at the WWTP inlet, whereas the output layer was limited to one layer, but with two neurons representing the values of (COD_{eff} and BOD_{5eff}) at the outlet of the plant. Table 4 shows the best performance of the (COD_{eff} and BOD_{5eff}) models selected from various studied architectures. Figure 5a,b represents the best performance of the first and second scenario, as well as the correlation coefficients for all training, validation, and testing groups, in addition to the set of all data. Observed versus predicted ensemble COD_{eff} and BOD_{5eff} values are presented in Figure 5c.

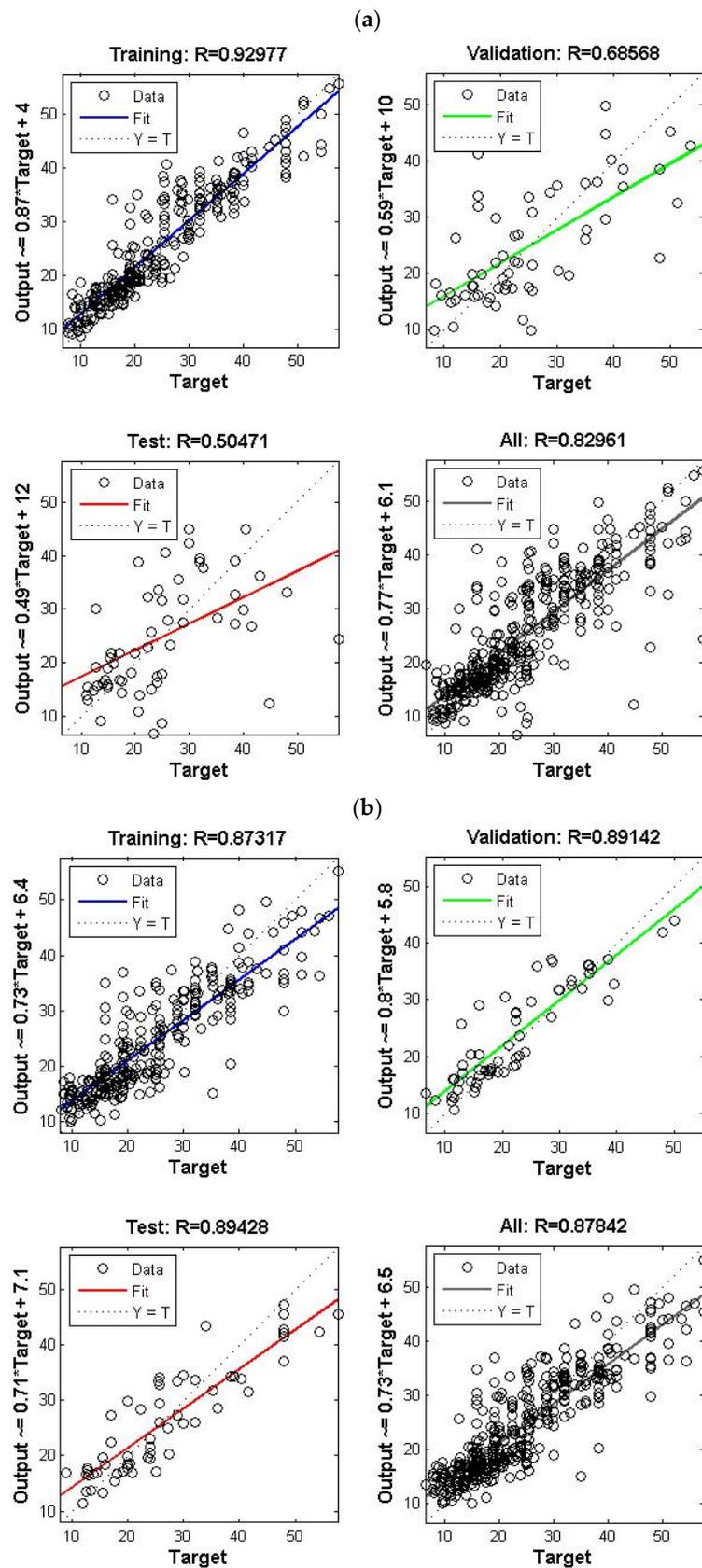


Figure 5. Cont.

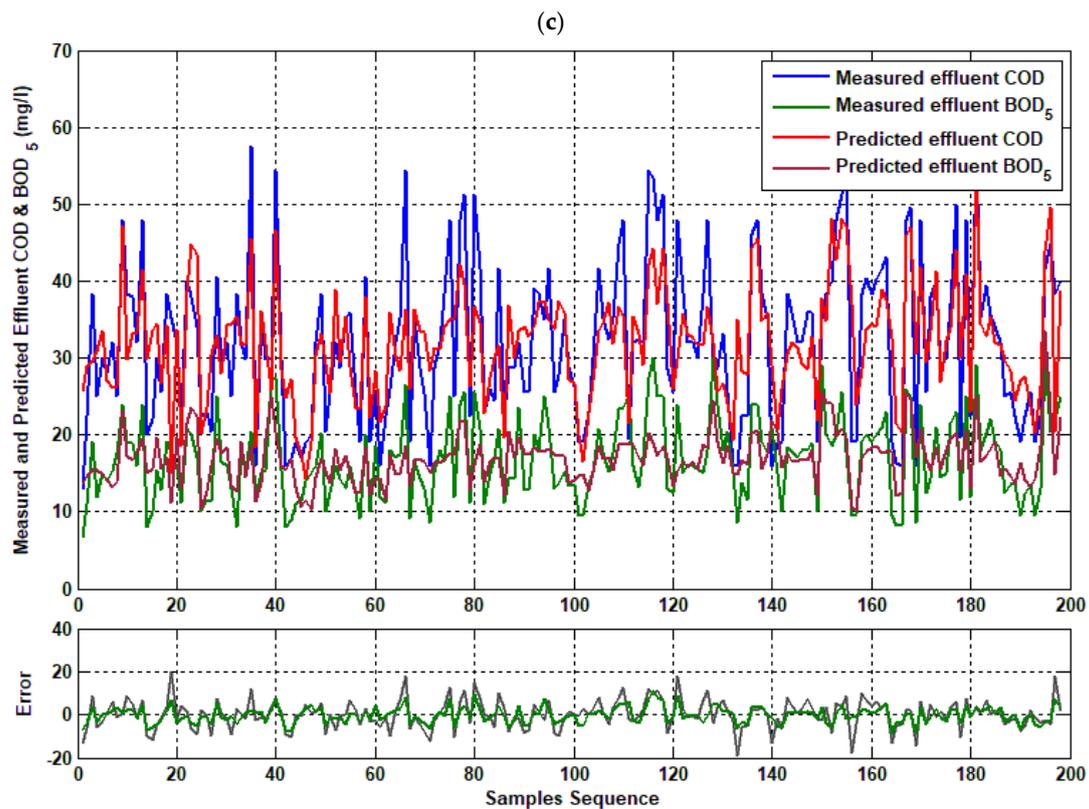


Figure 5. (a) The values of correlation coefficient (R) in all stages of Model M25-S1 in the first scenario. (b) The values of correlation coefficient (R) in all stages of Model M25-S2 in the second scenario. (c) Measured vs. predicted samples sequence obtained by best ensemble (COD and BOD₅) effluent models M25-S2.

Table 4. Performance of the (COD_{eff} and BOD_{5eff}) artificial neural network models.

Model No.	Network Input	Network Output	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden Layers	MSE
M21-S1 M21-S2	$T_{inf}, pH_{inf}, EC_{inf}, TDS_{inf}$	COD _{eff} & BOD _{5eff}	0.806 0.804	0.351 0.850	0.411 0.833	0.62 0.811	67 40–60	242.09 34.22
M22-S1 M22-S2	$T_{inf}, pH_{inf}, EC_{inf}, TDS_{inf}, NO_3^-_{inf}$	COD _{eff} & BOD _{5eff}	0.53 0.74	0.34 0.756	0.52 0.837	0.5 0.756	65 30–55	212.37 46.54
M23-S1 M23-S2	$T_{inf}, pH_{inf}, EC_{inf}, TDS_{inf}, NO_3^-_{inf}, PO_4^{3-}_{inf}$	COD _{eff} & BOD _{5eff}	0.866 0.749	0.46 0.772	0.57 0.828	0.57 0.761	65 40–60	138.95 60.04
M24-S1 M24-S2	$T_{inf}, pH_{inf}, EC_{inf}, TDS_{inf}, NO_3^-_{inf}, PO_4^{3-}_{inf}, BOD_{5inf}$	COD _{eff} & BOD _{5eff}	0.921 0.818	0.432 0.891	0.446 0.87	0.75 0.834	55 30–50	126.998 22.59
M25-S1 M25-S2	$T_{inf}, pH_{inf}, EC_{inf}, TDS_{inf}, NO_3^-_{inf}, PO_4^{3-}_{inf}, BOD_{5inf}, COD_{inf}$	COD _{eff} & BOD _{5eff}	0.929 0.873	0.685 0.891	0.504 0.894	0.829 0.878	60 30–50	75 34.34

4.6. Ensemble Model (PO_4^{3-} and NO_3^- Effluent)

We constructed several artificial neural networks that included a single input layer consisting of five scenarios in the same way as the previous models. The input neurons values were taken at the plant's inlet, whereas the output layer was restricted to one layer, but with two neurons representing the values of (PO_4^{3-} eff and NO_3^- eff) at the outlet of the plant. Table 5 shows the best performance of (PO_4^{3-} eff and NO_3^- eff) models selected from several studied architectures. Figure 6a,b represent the best models' performance, as well as the correlation coefficients for all training, validation, and testing groups, in addition to the set of all the data for the two scenarios (one and two hidden layers). Observed versus predicted (PO_4^{3-} eff and NO_3^- eff) values are illustrated in Figure 6c.

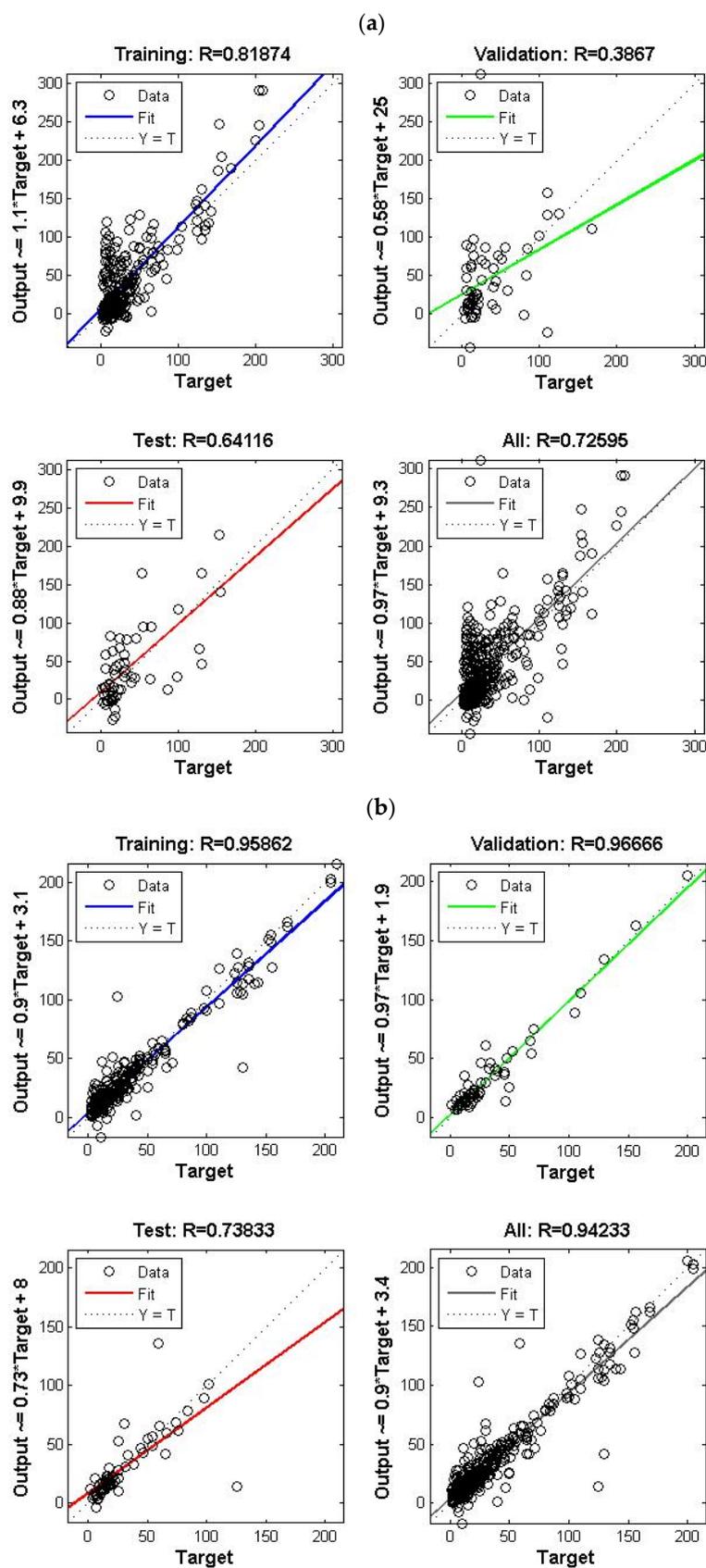


Figure 6. Cont.

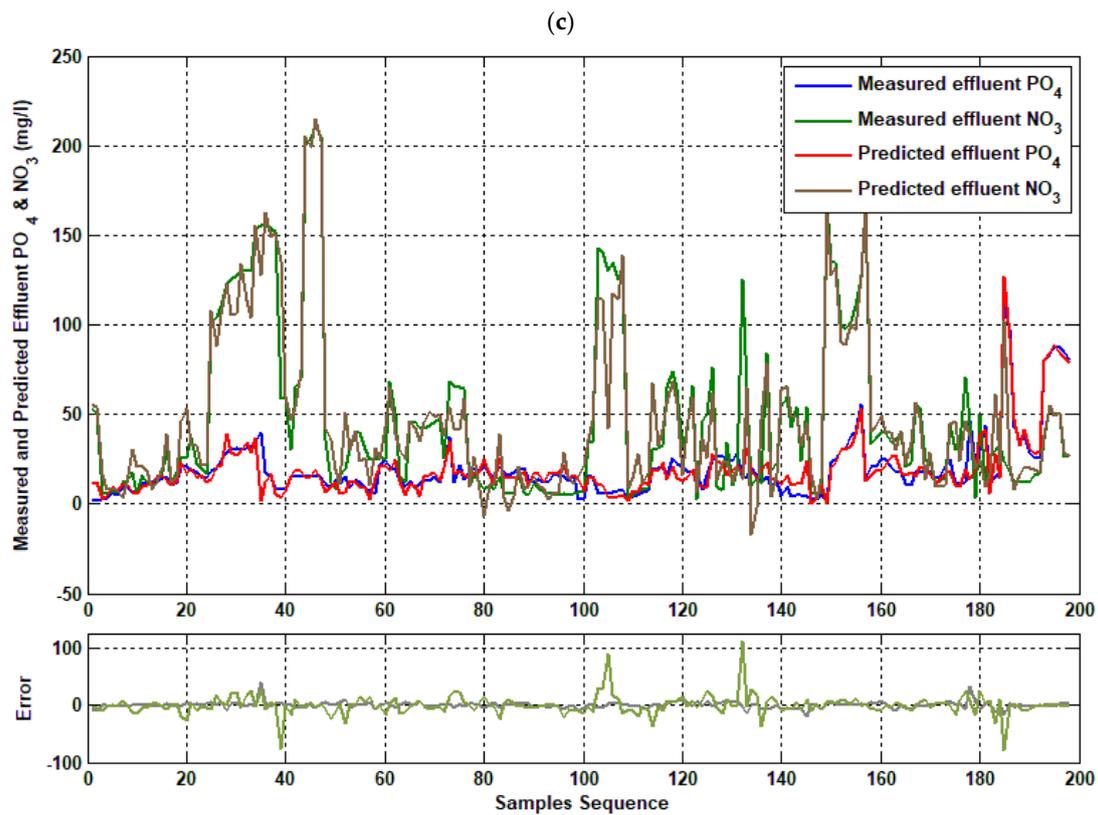


Figure 6. (a) The values of correlation coefficient (R) in all stages of Model M29-S1 in the first scenario. (b) The values of correlation coefficient (R) in all stages of Model M29-S2 in the second scenario. (c) Measured vs. predicted samples sequence obtained by the best ensemble (PO_4^{-3} and NO_3^-) effluent models M29-S2.

Table 5. Performance of the (PO_4^{-3} eff and NO_3^- eff) artificial neural network models.

Model No.	Network Input	Network Output	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden layers	MSE
M26-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf}	PO_4^{-3} eff &	0.631	0.376	0.223	0.496	67	2025.37
M26-S2		NO_3^- eff	0.786	0.625	0.761	0.764	40–60	606.61
M27-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$	PO_4^{-3} eff &	0.753	0.488	0.617	0.685	65	1653.70
M27-S2		NO_3^- eff	0.862	0.870	0.8	0.854	30–55	391.07
M28-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$, $PO_4^{-3}_{inf}$	PO_4^{-3} eff &	0.923	0.14	0.48	0.58	65	5828.85
M28-S2		NO_3^- eff	0.939	0.938	0.85	0.929	40–60	179.08
M29-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$, $PO_4^{-3}_{inf}$, BOD_{5inf}	PO_4^{-3} eff &	0.818	0.386	0.64	0.725	55	2850.58
M29-S2		NO_3^- eff	0.958	0.966	0.738	0.942	30–50	56.94
M30-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , $NO_3^-_{inf}$, $PO_4^{-3}_{in}$, BOD_{5inf} , COD_{inf}	PO_4^{-3} eff &	0.639	0.296	0.442	0.513	60	3318.54
M30-S2		NO_3^- eff	0.962	0.931	0.957	0.957	30–50	198.4

4.7. Ensemble Model (COD, BOD_5 , PO_4^{-3} and NO_3^- Effluent)

We built several artificial neural networks with a single input layer made up of five subscenarios for the input variables: (4, 5, 6, 7, 8 neurons), as shown in Table A3. All the input values were taken at the WWTP inlet, whereas the output layer was limited to one layer, but with four neurons representing the values of (COD_{eff} , BOD_{5eff} , PO_4^{-3} eff and NO_3^- eff) at the plant's outlet. Table A3 shows the best performance of the (COD_{eff} , BOD_{5eff} , PO_4^{-3} eff and NO_3^- eff) models selected from several studied architectures. Figure A5a,b represent the best models' performance and the correlation coefficients for all training, validation, and testing groups, in addition to the set of all the data for the two scenarios (one and two hidden layers).

The results demonstrated that, for predicting the studied variables in full-scale WWTP, both single and combined models provide good results.

It is worth noting that using the two-hidden-layers SNN model (which aimed to decrease the weaknesses of the one-hidden-layer SNN model and coming up with improved and composite models, which are favorable and more reliable with high accuracy) gave superior results compared to one-hidden-layer models. For example, in relation to the ensemble model (COD, BOD₅, PO₄⁻³ and NO₃⁻ effluent), we can find from Table A3 that in the one-hidden-layer model (M35-S1) with architecture (8-60-4), eight influent pollution variable input, 60 neurons in the single hidden layer, and four effluent pollution variable output gave a 0.757 correlation coefficient for all the data and a mean squared error (MSE) of 622, whereas the two-hidden-layer model (M35-S2) with architecture (8-50-30-4), eight influent pollution variable input, 50 neurons in the first hidden layer, 30 neurons in the second hidden layer, and four effluent pollution variable output achieved higher correlation coefficient 0.936 for all the data and a mean squared error (MSE) of 51.05.

Moreover, the performance of all models with one and two hidden layers is improved by increasing the input plant's pollution variables.

The performance of the SNN and DNN models was satisfactory in the calibration and verification stages due to the solidity of the neural networks approach in processing nonlinear interactions and the capability to backpropagate the produced error through the calibration stage until the desired result was obtained.

The use of artificial neural network technology gave remarkable results in forecasting the performance of the WWTP. In addition, the application of shallow and deep learning increased the efficiency of this predicting, which contributes to the treatment plant's enhanced quality.

These ANN models aid in predicting the effluent quality of wastewater treatment plants, and thus, ANN models can provide a useful tool for modeling and predicting wastewater treatment plants' performance.

5. Conclusions

This paper presented the use of SNN and DNN models to predict the performances of a full-scale WWTP. It showed that the use of the influent and effluent concentrations of several pollution parameters (COD, BOD₅, PO₄⁻³, NO₃⁻, T, pH, EC, and TDS) gave a high modeling accuracy.

We studied the performance of several neural network techniques and architectures. Our results indicate that increasing the number of input variables and neurons in hidden layers improved the accuracy of the SNN models. Moreover, the ensemble models were more reliable, robust, and efficient than others.

In conclusion, we note that the first SNN scenario (one-hidden-layer model) generally gave good correlation values. Still, the correlation values in the validation and testing phase were also at acceptable levels. The second SNN scenario (two-hidden-layers models) and the random forest gave excellent correlation values in all stages (training, validation, testing).

DNN (CNN, RNN, SAE DNN) can be used for WWTP modeling, but due to the small datasets, it gave a lower performance and accuracy. Due to the general availability of small datasets in wastewater management, the shallow (one or two hidden layer) neural networks are highly recommended for modeling the WWTP process. The use of these models contributes to significantly reducing the periodic laboratory measurements, which minimizes the operational cost of these plants, and assessing the stability of environmental balance. Moreover, it is possible to add some operating parameters from the aerated activated sludge tank to the models and comparing it, which we recommend to include in future works.

Author Contributions: Conceptualization, R.J., A.A., K.J. and I.S.; methodology, R.J., A.A., K.J. and I.S.; software, R.J. and K.J.; validation, R.J., A.A., K.J. and I.S.; formal analysis, R.J., A.A., K.J. and I.S.; data curation, R.J. and K.J.; writing—original draft preparation, R.J., A.A., K.J. and I.S.; writing—review and editing, R.J., A.A., K.J. and I.S.; visualization, R.J. and K.J.; supervision, A.A. and I.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

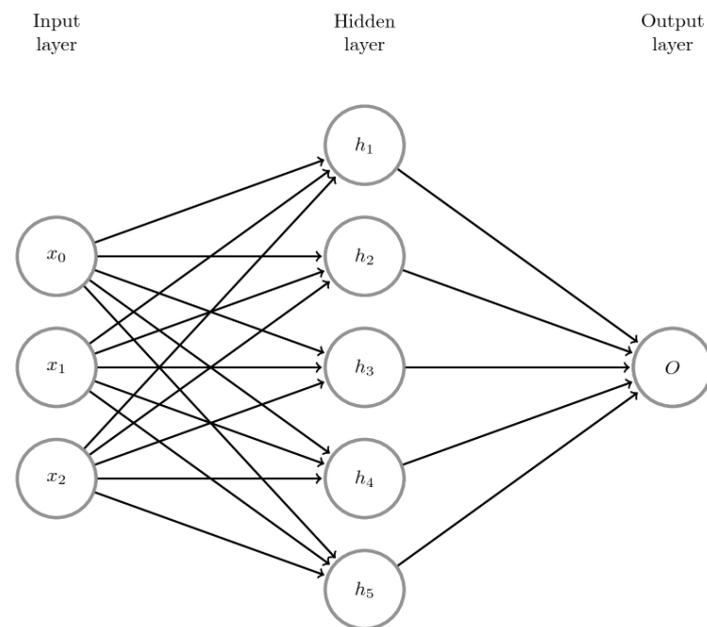


Figure A1. Schematic of a three-layer artificial neural network.

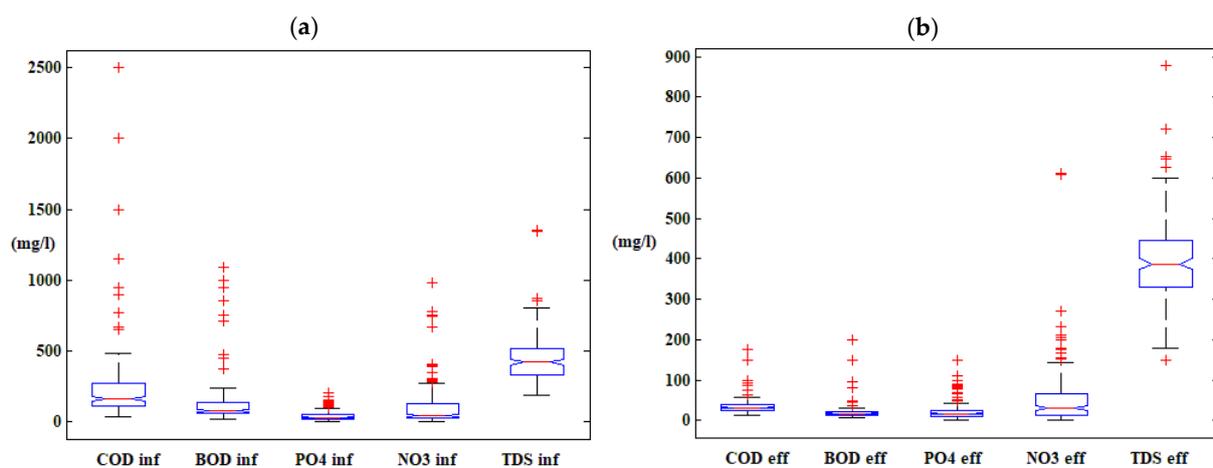


Figure A2. (a) Analysis of influent parameters using ANOVA1, (b) analysis of effluent parameters using ANOVA1.

Table A1. Performance of the BOD_{5eff} artificial neural network models.

Model No.	Model Input Variables	Model Output Variable(S)	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden Layers	MSE
M6-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf}	BOD _{5eff}	0.515	0.774	0.564	0.541	67	13.964
M6-S2			0.685	0.361	0.825	0.681	40–60	18.28
M7-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf}	BOD _{5eff}	0.69	0.69	0.624	0.68	65	10.926
M7-S2			0.675	0.520	0.820	0.678	30–55	18.51
M8-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf} , PO ₄ ⁻³ _{inf}	BOD _{5eff}	0.534	0.596	0.687	0.557	65	19.223
M8-S2			0.704	0.648	0.825	0.715	40–60	11.10
M9-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf} , PO ₄ ⁻³ _{inf} , BOD _{5inf}	BOD _{5eff}	0.753	0.82	0.87	0.78	55	10.09
M9-S2			0.901	0.942	0.895	0.906	30–50	2.14
M10-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf} , PO ₄ ⁻³ _{inf} , BOD _{5inf} , COD _{inf}	BOD _{5eff}	0.674	0.775	0.51	0.651	60	20.224
M10-S2			0.890	0.935	0.898	0.898	30–50	6.55

Table A2. Performance of the NO₃⁻ eff artificial neural network models.

Model No.	Network Input	Network Output	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden Layers	MSE
M16-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf}	NO ₃ ⁻ _{eff}	0.606	0.273	0.245	0.489	67	4135.91
M16-S2			0.83	0.485	0.209	0.607	40–60	1915
M17-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{in}	NO ₃ ⁻ _{eff}	0.907	0.448	0.505	0.715	65	5015.33
M17-S2			0.901	0.988	0.809	0.912	30–55	223.42
M18-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf inf} , PO ₄ ⁻³ _{inf}	NO ₃ ⁻ _{eff}	0.972	0.598	0.661	0.789	65	2054.71
M18-S2			0.90	0.975	0.864	0.909	40–60	188.87
M19-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf} , PO ₄ ⁻³ _{inf} , BOD _{5inf}	NO ₃ ⁻ _{eff}	0.969	0.698	0.581	0.793	55	2520.53
M19-S2			0.905	0.844	0.914	0.9	30–50	406.11
M20-S1	T _{inf} , pH _{inf} , EC _{inf} , TDS _{inf} , NO ₃ ⁻ _{inf} , PO ₄ ⁻³ _{inf} , BOD _{5inf} , COD _{inf}	NO ₃ ⁻ _{eff}	0.876	0.779	0.576	0.822	60	2038.57
M20-S2			0.928	0.953	0.934	0.932	30–50	65.94

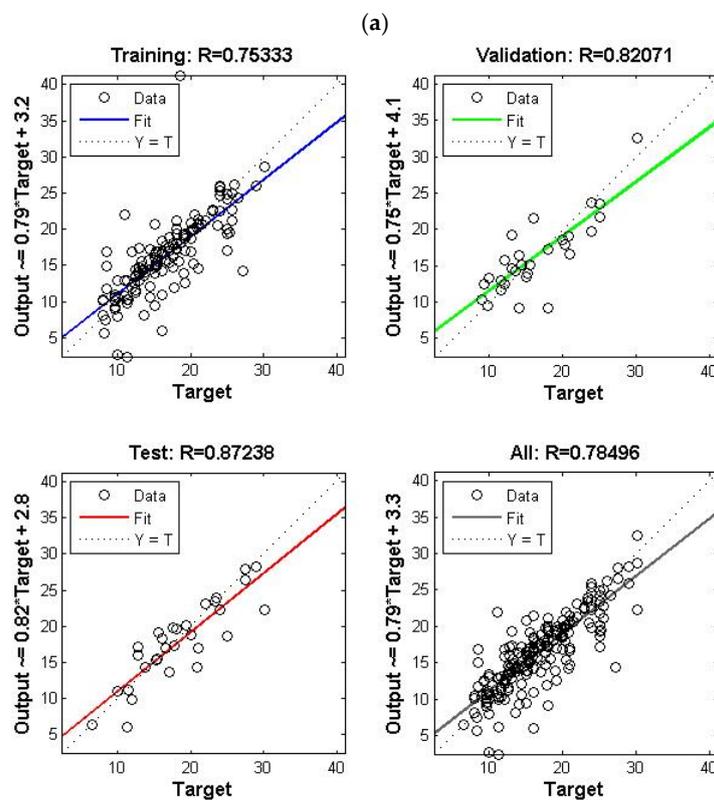


Figure A3. Cont.

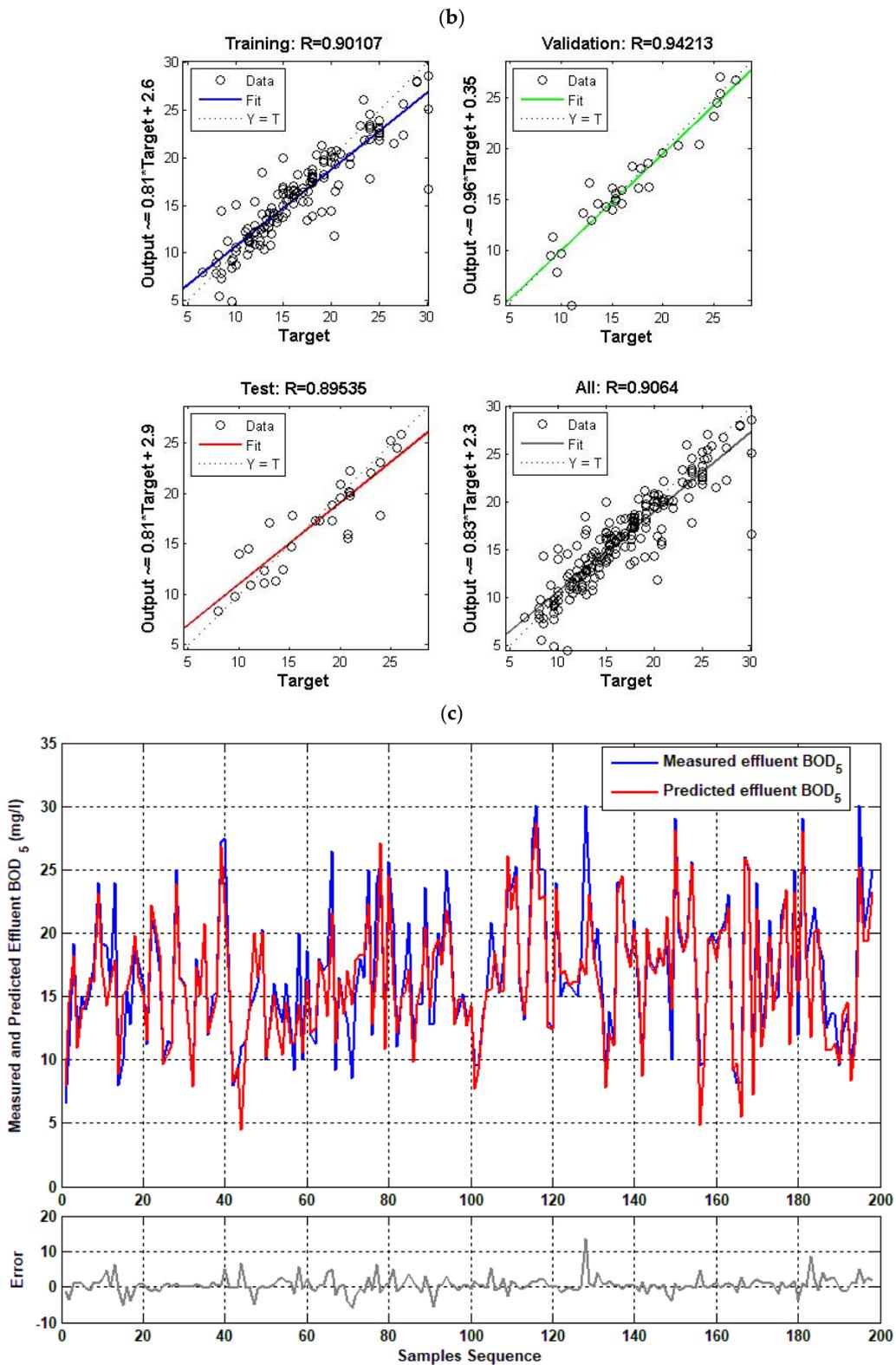


Figure A3. Cont.

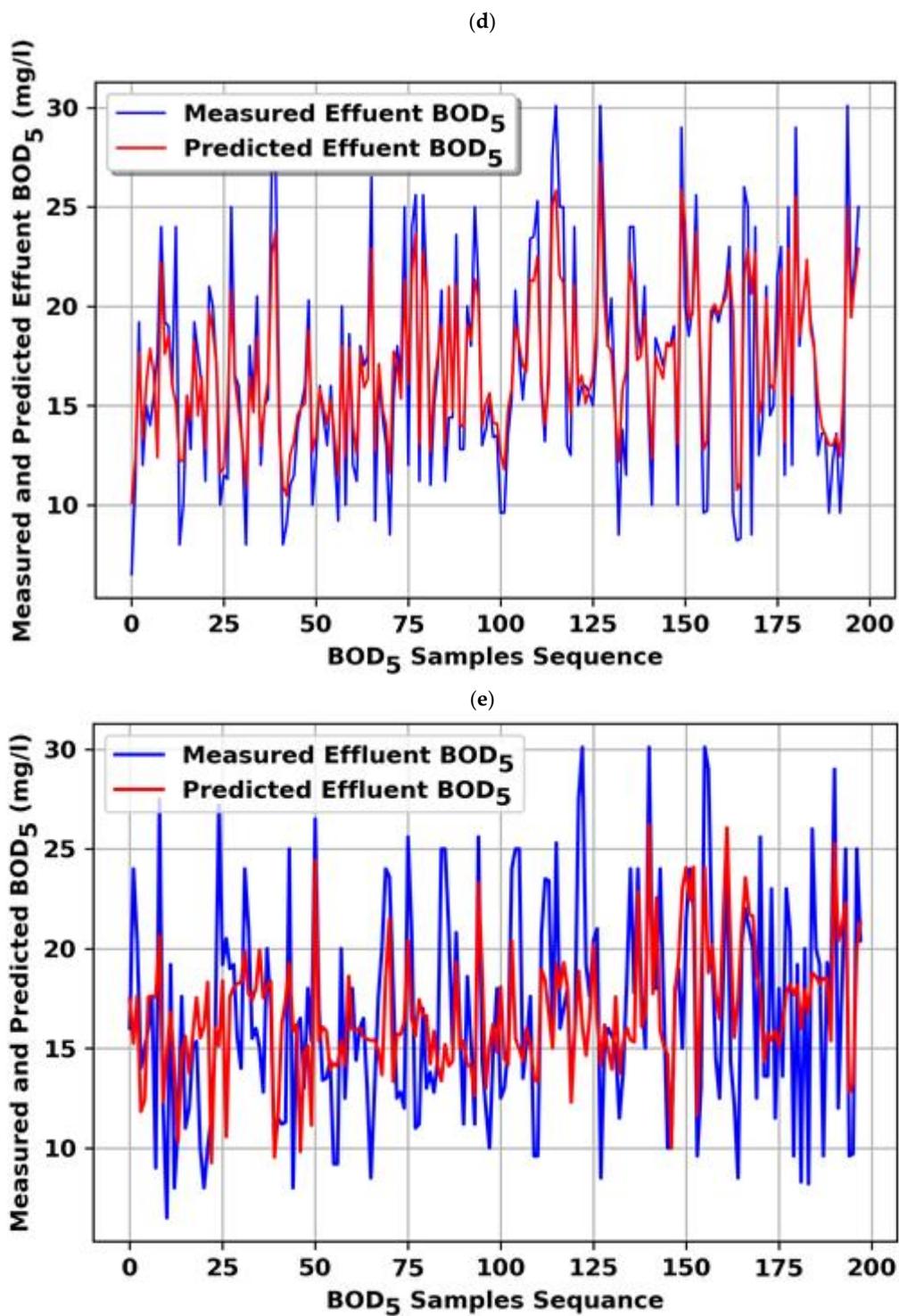


Figure A3. Cont.

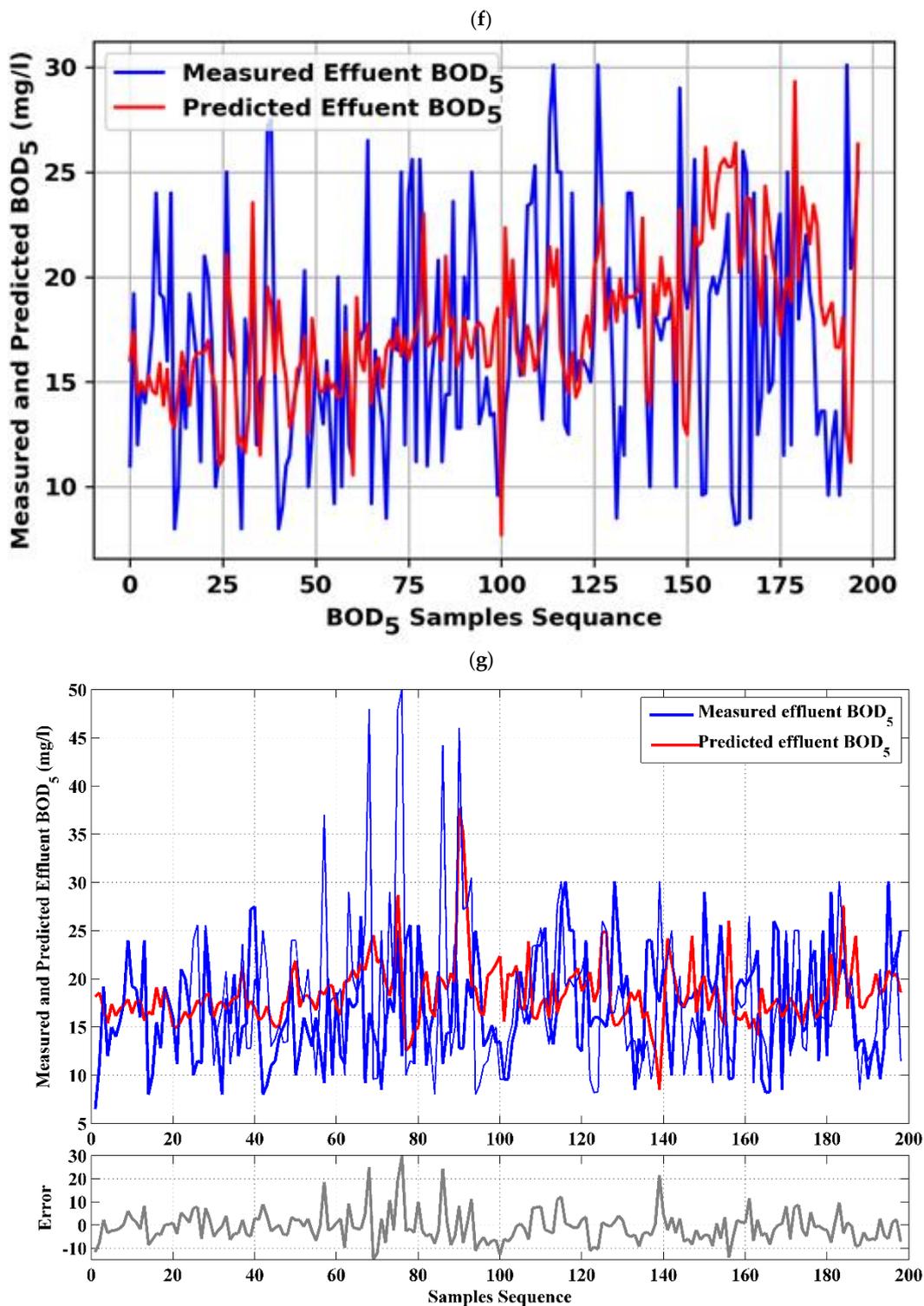


Figure A3. (a) The values of correlation coefficient (R) in all stages of Model M9-S1 in the first scenario. (b) The values of correlation coefficient (R) in all stages of Model M9-S2 in the second scenario. (c) Measured vs. predicted samples sequence achieved by best single BOD₅ effluent models M9-S2. (d) Measured vs. predicted samples sequence obtained by random forest (RF) BOD₅ effluent models. (e) Measured vs. predicted samples sequence obtained by (CNN) convolutional neural network BOD₅ effluent models. (f) Measured vs. predicted samples sequence obtained by (RNN) recurrent neural network BOD₅ effluent models. (g) Measured vs. predicted samples sequence obtained by (DNN) SAE neural network BOD₅ effluent models.

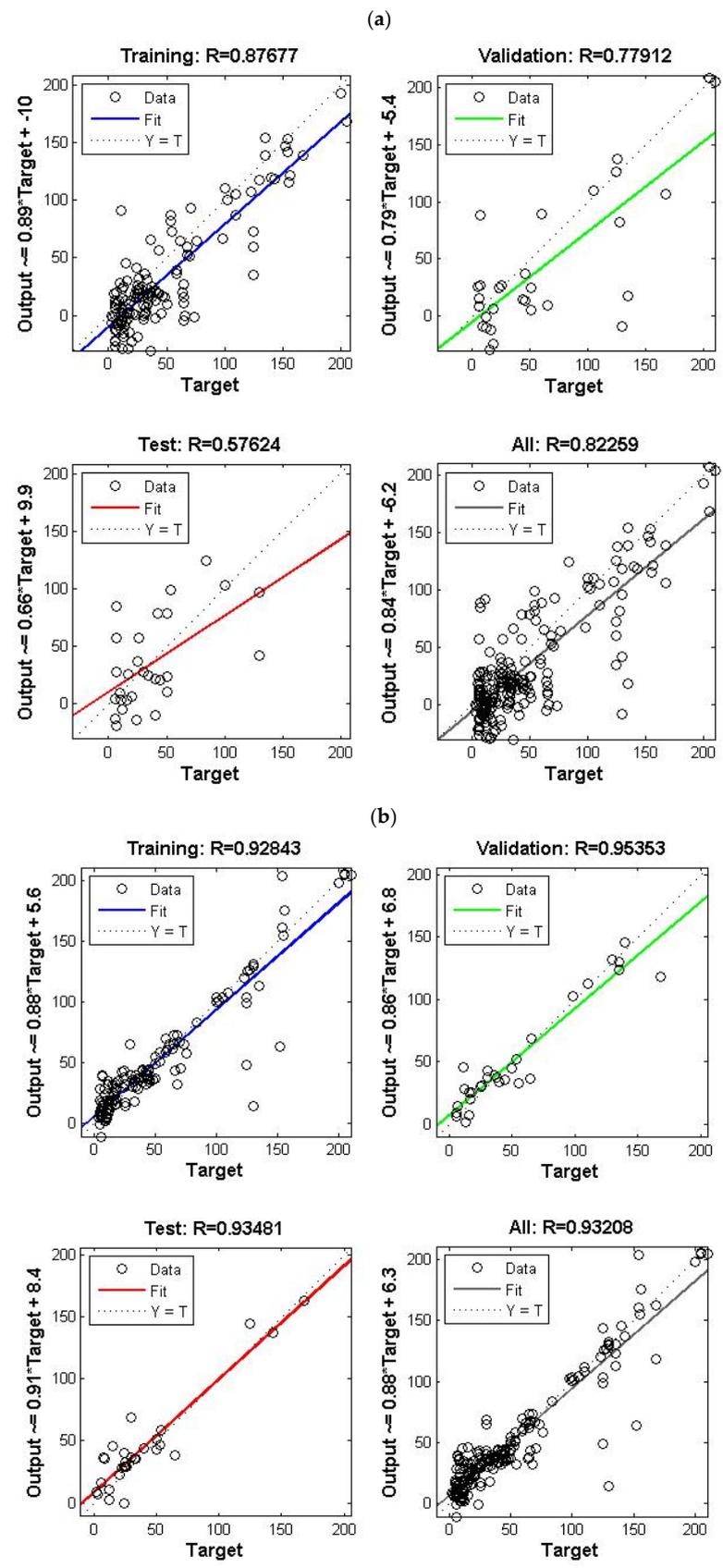


Figure A4. Cont.

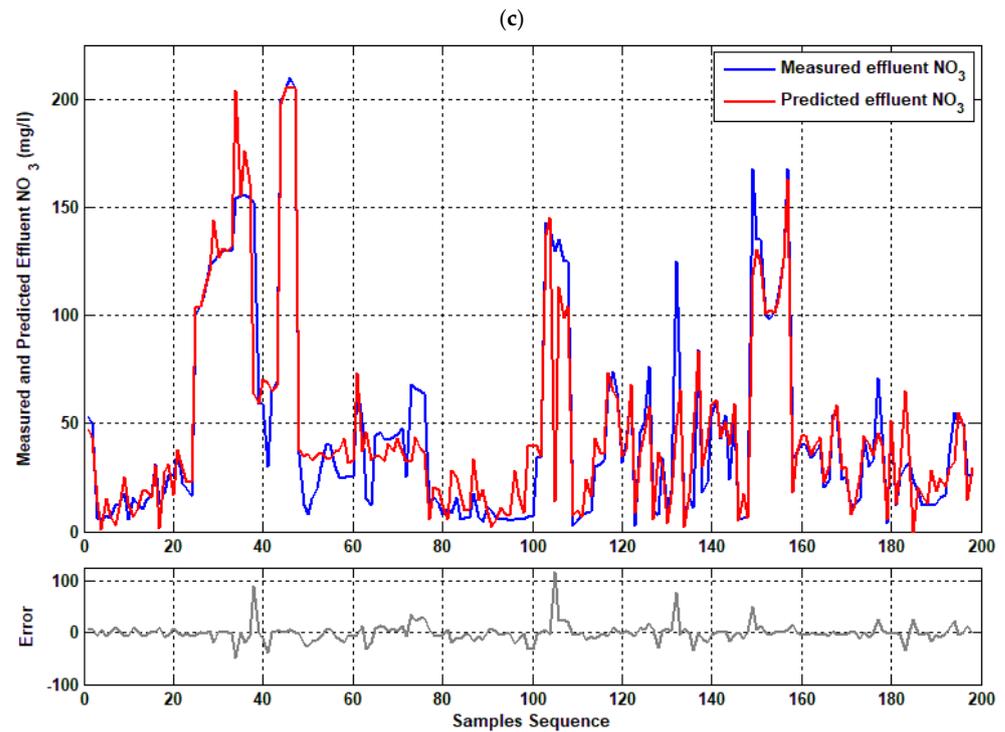


Figure A4. (a) The values of correlation coefficient (R) in all stages of Model M20-S1 in the first scenario. (b) The values of correlation coefficient (R) in all stages of Model M20-S2 in the second scenario. (c) Measured vs. predicted samples sequence achieved by best single NO_3^- effluent models M20-S2.

Table A3. Performance of the (COD_{eff} , $\text{BOD}_{5\text{eff}}$, PO_4^{-3} eff and NO_3^- eff) artificial neural network models.

Model No.	Network Input	Network Output	Training (Correlation Coefficient)	Validation (Correlation Coefficient)	Testing (Correlation Coefficient)	All Data (Correlation Coefficient)	No. Neurons in Hidden Layers	MSE
M31-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf}	COD_{eff} & $\text{BOD}_{5\text{eff}}$ &	0.872	0.30	0.62	0.682	67	1707.26
M31-S2		PO_4^{-3} eff & NO_3^- eff	0.803	0.490	0.603	0.732	40–60	665.98
M32-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , NO_3^- inf	COD_{eff} & $\text{BOD}_{5\text{eff}}$ &	0.848	0.487	0.604	0.75	65	870.369
M32-S2		PO_4^{-3} eff & NO_3^- eff	0.932	0.852	0.755	0.872	30–55	185.41
M33-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , NO_3^- inf , PO_4^{-3} inf	COD_{eff} & $\text{BOD}_{5\text{eff}}$ &	0.829	0.418	0.352	0.671	65	748.01
M33-S2		PO_4^{-3} eff & NO_3^- eff	0.938	0.931	0.858	0.625	40–60	78.93
M34-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , NO_3^- inf , PO_4^{-3} inf , $\text{BOD}_{5\text{inf}}$	COD_{eff} & $\text{BOD}_{5\text{eff}}$ &	0.793	0.641	0.497	0.706	55	711.238
M34-S2		PO_4^{-3} eff & NO_3^- eff	0.909	0.965	0.973	0.928	30–50	69.71
M35-S1	T_{inf} , pH_{inf} , EC_{inf} , TDS_{inf} , NO_3^- inf , PO_4^{-3} inf , $\text{BOD}_{5\text{inf}}$, COD_{inf}	COD_{eff} & $\text{BOD}_{5\text{eff}}$ &	0.946	0.571	0.339	0.757	60	621.78
M35-S2		PO_4^{-3} eff & NO_3^- eff	0.925	0.952	0.972	0.936	30–50	51.05

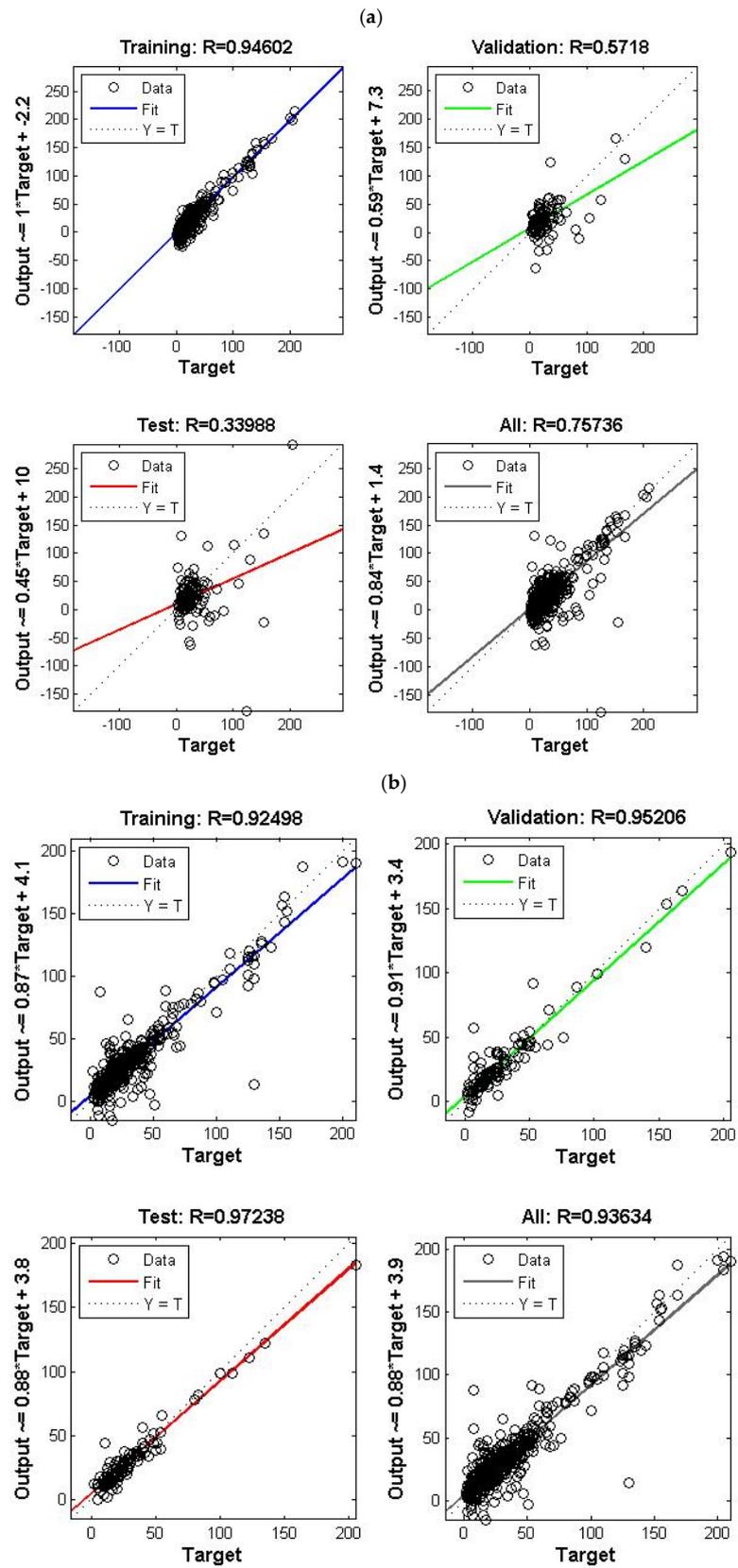


Figure A5. (a) The values of correlation coefficient (R) in all stages of Model M35-S1 in the first scenario. (b) The values of correlation coefficient (R) in all stages of Model M35-S2 in the second scenario.

References

1. Vanrolleghem, P.; Verstraete, W. Simultaneous biokinetic characterization of heterotrophic and nitrifying populations of activated sludge with an on-line respirographic biosensor. *Water Sci. Technol.* **1993**, *28*, 377–387. [CrossRef]
2. Vassos, T.D. Future directions in instrumentation, control and automation in the water and wastewater industry. *Water Sci. Technol.* **1993**, *28*, 9–14. [CrossRef]
3. Harremoë, P.; Capodaglio, A.G.; Hellström, B.G.; Henze, M.; Jensen, K.N.; Lynggaard-Jensen, A.; Otterpohl, R.; Søeberg, H. Wastewater treatment plants under transient loading—Performance, modelling and control. *Water Sci. Technol.* **1993**, *27*, 71. [CrossRef]
4. Mjalli, F.S.; Al-Asheh, S.; Alfadala, H. Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance. *J. Environ. Manag.* **2007**, *83*, 329–338. [CrossRef]
5. Hamoda, M.F.; Al-Ghusain, I.A.; Hassan, A.H. Integrated wastewater treatment plant performance evaluation using artificial neural networks. *Water Sci. Technol.* **1999**, *40*, 55–65. [CrossRef]
6. Nasr, M.S.; Moustafa, M.A.E.; Seif, H.A.E.; El Kobrosy, G. Application of Artificial Neural Network (ANN) for the prediction of EL-AGAMY wastewater treatment plant performance-EGYPT. *Alex. Eng. J.* **2012**, *51*, 37–43. [CrossRef]
7. Hong, Y.-S.T.; Rosen, M.R.; Bhamidimarri, R. Analysis of a municipal wastewater treatment plant using a neural network-based pattern analysis. *Water Res.* **2003**, *37*, 1608–1618. [CrossRef] [PubMed]
8. Lee, D.S.; Park, J.M. Neural network modeling for on-line estimation of nutrient dynamics in a sequentially-operated batch reactor. *J. Biotechnol.* **1999**, *75*, 229–239. [CrossRef]
9. Côte, M.; Grandjean, B.P.A.; Lessard, P.; Thibault, J. Dynamic modelling of the activated sludge process: Improving prediction using neural networks. *Water Res.* **1995**, *29*, 995–1004. [CrossRef]
10. Hamed, M.M.; Khalafallah, M.G.; Hassani, E.A. Prediction of wastewater treatment plant performance using artificial neural networks. *Environ. Model. Softw.* **2004**, *19*, 919–928. [CrossRef]
11. Maier, H.R.; Dandy, G.C. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environ. Model. Softw.* **2000**, *15*, 101–124. [CrossRef]
12. Blaesi, J.; Jensen, B. Can Neural Networks Compete with Process Calculations. *InTech* **1992**, *39*. Available online: <https://www.osti.gov/biblio/6370708> (accessed on 2 October 2022).
13. Rene, E.R.; Saidutta, M. Prediction of BOD and COD of a refinery wastewater using multilayer artificial neural networks. *J. Urban Environ. Eng.* **2008**, *2*, 1–7. [CrossRef]
14. Vyas, M.; Modhera, B.; Vyas, V.; Sharma, A. Performance forecasting of common effluent treatment plant parameters by artificial neural network. *ARPN J. Eng. Appl. Sci.* **2011**, *6*, 38–42.
15. Jami, M.S.; Husain, I.; Kabbashi, N.A.; Abdullah, N. Multiple inputs artificial neural network model for the prediction of wastewater treatment plant performance. *Aust. J. Basic Appl. Sci.* **2012**, *6*, 62–69.
16. Pakrou, S.; Mehrdadi, N.; Baghvand, A. Artificial neural networks modeling for predicting treatment efficiency and considering effects of input parameters in prediction accuracy: A case study in tabriz treatment plant. *Indian J. Fundam. Appl. Life Sci.* **2014**, *4*, 2231–6345.
17. Nourani, V.; Elkiran, G.; Abba, S. Wastewater treatment plant performance analysis using artificial intelligence—An ensemble approach. *Water Sci. Technol.* **2018**, *78*, 2064–2076. [CrossRef]
18. Wang, D.; Thunéll, S.; Lindberg, U.; Jiang, L.; Trygg, J.; Tysklind, M.; Souihi, N.A. machine learning framework to improve effluent quality control in wastewater treatment plants. *Sci. Total Environ.* **2021**, *784*, 147138. [CrossRef]
19. Zhu, X.; Xu, Z.; You, S.; Komárek, M.; Alessi, D.S.; Yuan, X.; Palansooriya, K.N.; Ok, Y.S.; Tsang, D.C.W. Machine learning exploration of the direct and indirect roles of Fe impregnation on Cr (VI) removal by engineered biochar. *Chem. Eng. J.* **2022**, *428*, 131967. [CrossRef]
20. Zhu, X.; Wan, Z.; Tsang, D.C.W.; He, M.; Hou, D.; Su, Z.; Shang, J. Machine learning for the selection of carbon-based materials for tetracycline and sulfamethoxazole adsorption. *Chem. Eng. J.* **2021**, *406*, 126782. [CrossRef]
21. Alsulaili, A.; Refaie, A. Artificial neural network modeling approach for the prediction of five-day biological oxygen demand and wastewater treatment plant performance. *Water Supply* **2021**, *21*, 1861–1877. [CrossRef]
22. Wu, X.; Yang, Y.; Wu, G.; Mao, J.; Zhou, T. Simulation and optimization of a coking wastewater biological treatment process by activated sludge models (ASM). *J. Environ. Manag.* **2016**, *165*, 235–242. [CrossRef]
23. Henze, M.; Gujer, W.; Mino, T.; Matsuo, T.; Wentzel, M.C.; Marais, G.V.R.; van Loosdrecht, M.C.M. Activated sludge model no. 2d, ASM2d. *Water Sci. Technol.* **1999**, *39*, 165–182. [CrossRef]
24. Müller, A.C.; Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2016.
25. Delgrange, N.; Cabassud, C.; Cabassud, M.; Durand-Bourlier, L.; Lainé, J.M. Neural networks for prediction of ultrafiltration transmembrane pressure—application to drinking water production. *J. Membr. Sci.* **1998**, *150*, 111–123. [CrossRef]
26. Eslamian, S.; Gohari, A.; Biabanaki, M.; Malekian, R. Estimation of monthly pan evaporation using artificial neural networks and support vector machines. *J. Appl. Sci.* **2008**, *8*, 3497–3502. [CrossRef]
27. Taylor, J.G. *Neural Networks and Their Applications*; John Wiley and Sons: Hoboken, NJ, USA, 1996; p. 322.
28. José, C.; Principe, N.R.E.; Lefebvre, W.C. *Neural and Adaptive Systems: Fundamentals through Simulations*; John Wiley and Sons: Hoboken, NJ, USA, 1999.

29. Das, H.S.; Roy, P. A Deep Dive into Deep Learning Techniques for Solving Spoken Language Identification Problems. In *Intelligent Speech Signal Processing*; Elsevier: Amsterdam, The Netherlands, 2019; pp. 81–100.
30. Feng, S.; Zhou, H.; Dong, H. Using deep neural network with small dataset to predict material defects. *Mater. Des.* **2018**, *162*, 300–310. [[CrossRef](#)]
31. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
32. Ward, L.; Agrawal, A.; Choudhary, A.; Wolverson, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *NPJ Comput. Mater.* **2016**, *2*, 16028. [[CrossRef](#)]
33. Hagan, M.T.; Demuth, H.B.; Beale, M.H.; De Jesus, O. *Neural Network Design*; Martin Hagan: Stillwater, OK, USA, 2014.
34. Nourani, V.; Baghanam, A.H.; Gebremichael, M. Investigating the Ability of Artificial Neural Network (ANN) Models to Estimate Missing Rain-gauge Data. *J. Environ. Inform.* **2012**, *19*, 38–50. [[CrossRef](#)]
35. Nourani, V.; Hakimzadeh, H.; Amini, A.B. Implementation of artificial neural network technique in the simulation of dam breach hydrograph. *J. Hydroinform.* **2012**, *14*, 478–496. [[CrossRef](#)]
36. Breiman, L.; Friedman, J.; Stone, C.J.; Olshen, R.A. *Classification and Regression Trees*; Routledge: Milton Park, Abingdon-on-Thames, UK, 2017.
37. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, 1–21. [[CrossRef](#)] [[PubMed](#)]
38. Wu, J. *Introduction to Convolutional Neural Networks*; National Key Lab for Novel Software Technology, Nanjing University: Nanjing, China, 2017; Volume 5, p. 23.
39. Apaydin, H.; Feizi, H.; Sattari, M.T.; Colak, M.S.; Shamshirband, S.; Chau, K.-W. Comparative analysis of recurrent neural network architectures for reservoir inflow forecasting. *Water* **2020**, *12*, 1500. [[CrossRef](#)]
40. Li, L.; Jiang, P.; Xu, H.; Lin, G.; Guo, D.; Wu, H. Water quality prediction based on recurrent neural network and improved evidence theory: A case study of Qiantang River, China. *Environ. Sci. Pollut. Res.* **2019**, *26*, 19879–19896. [[CrossRef](#)] [[PubMed](#)]