Romanian Journal of Petroleum & Gas Technology
Vol. V (LXXVI) • No. 2/2024

A COMPARATIVE STUDY OF DEEP LEARNING METHODS APPLIED FOR WASTEWATER pH NEUTRALIZATION PROCESS MODELLING

Mădălina Cărbureanu [1]

[1] Petroleum-Gas University of Ploiesti, Romania
email: mcarbureanu@upg-ploiesti.ro

DOI: 10.51865/JPGT.2024.02.09

ABSTRACT

In a wastewater treatment plant (WWTP) there are several different, intricate processes with a dynamic and nonlinear behaviour. The fact that these processes are nonlinear, some of them having a high degree of nonlinearity, as is the wastewater pH neutralization process, comes with a number of problems related to their modelling and control. The identification of any method that can be used to simplify the modelling and control of such a high nonlinear process, it is a desideratum to ensure a quality effluent of the plant, because its water quality is affected by the treated wastewater discharged into it. The Deep Learning (DL) and Machine Learning (ML) techniques offer incredible solutions that can be explored in order to find out the optimal tool that can be used for wastewater pH treatment process modelling. In the present paper, seven DL solutions were implemented and tested in order to identify the most appropriate DL method for modelling this type of process, method that ensures the best result. The analysed DL methods are Feedforward Neural Networks (FNNs), Temporal Convolutional Networks (TCNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), General Regression Neural Networks (GRNNs), Time Delay Neural Networks (TDNNs) and Deep Belief Networks (DBNs), being implemented using Python 3.9 software and Tensorflow. The analysis made with the mentioned DL methods, was based on knowing the flowrate of the acid reactant (which was maintained constant), the initial alkaline reactant flowrate in the treated solution, the initial pH level, and the desired pH level, with the final goal of predicting the required quantity of alkaline reactant flowrate necessary to obtain a neutral pH.

Keywords: deep learning, machine learning, wastewater treatment plant, neural networks, wastewater pH neutralization, process modelling

INTRODUCTION

The wastewater treatment processes from a WWTP are varied and complex ones, with a high level of disturbance variability. One of the most important aspect is that these processes are nonlinear, some of them presenting high nonlinearity, as is the wastewater pH neutralization process. From all the plant chemical processes, the wastewater pH neutralization process has a dynamic behaviour and a high nonlinearity given by the static characteristics (titration curves) shape. There is a strong pH variation around the equivalence point (pH≈7 units), variation that is influenced by the reactant (acid or

alkaline type) nature and concentration, being highlighted the fact that a reduction in reactants concentration leads to a pH variation reduction around the equivalence point. Also, it was established through laboratory experiments, that the pH value has a strong variation even when it is added a small amount of acid or alkaline type reactant, the process being very instable around the equivalence point. Also, the neutralization process nonlinearity of a strong acid with a strong alkaline is more pronounced, than in the case of a week acid with a strong alkaline reactant neutralization [2, 6, 7, 16, 21, 26].

In Figure 1, is presented the wastewater pH neutralization process block diagram from the author perspective, with the associated transducer (*p*H-meter), actuators (sulphuric acid and hydrated lime reactants dosage pumps), the acid reactant flowrate (F_1) and its concentration (C_1), the hydrated lime flowrate (F_2) and its concentration (C_2), stirrer and the process admixture reaction tank.

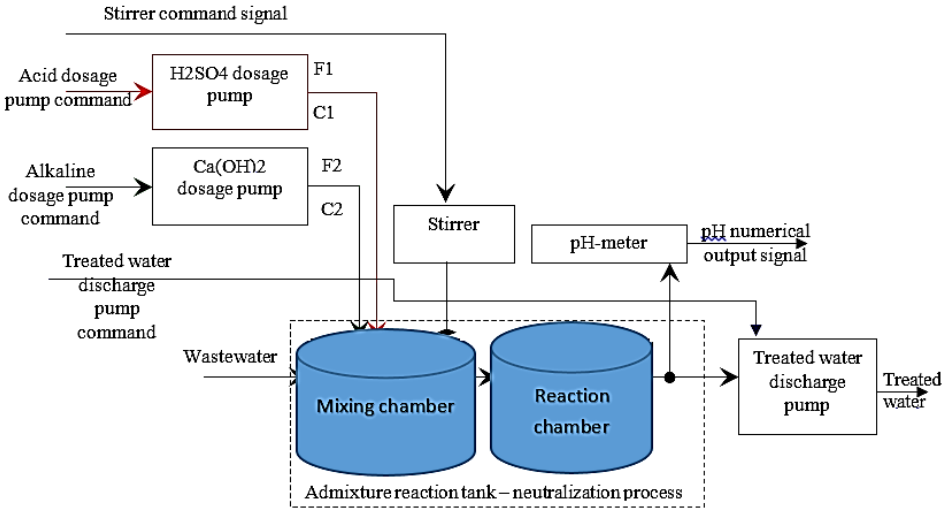

Figure 1. *Wastewater pH neutralization process block diagram*

Due to the fact that the wastewater *p*H neutralization process from a WWTP is very hard to model or control using conventional tools (PID algorithms and Gain-Scheduling PID), the goal is to identify that DL method which is more suitable to model such a high non-linear process [6, 7]. It is known that machine learning (ML) and DL are both artificial intelligence (AI) subdomains with large application in wastewater treatment domain. So, in paper [5], were compared six ML techniques (decision tree regression, linear regression, k-nearest neighbours regression, support vector machine regression, gradient boosting regression and random forest regression), establishing the most efficient ML regression algorithm (respectively, the gradient boosting regression) for modelling the wastewater *p*H neutralization. As a continuation, the present article aims to identify that DL method, from the analysed ones (RNNs, TCNs, FNNs, TDNNs, DBNs, GRNNs and LSTM) which is more suitable to model such a high non-linear process. The novelty is given by the identification of a useful tool (a DL based ANNs method) that can be successfully applied for an efficient and time-saving modelling of a dynamic and high nonlinear process, like wastewater pH neutralization from a WWTP. The identified tool can be successfully applied in developing data-driven type models dedicated to high nonlinear treatment processes from a WWTP, with real benefits for the plant human operator in the decision making process.

DEEP LEARNING METHODS APPLIED IN WASTEWATER TREATMENT PROCESSES

Deep Learning is a subset of ML, while ML is an artificial intelligence (AI) technique. So, ML is an AI technique that can automatically adapt with minimal human intervention, while DL is a subset of ML that uses ANNs in order to reproduce the human brain learning process. DL is better than other (non-AI) methods because it can handle non-linear data, can solve regression problems, can use data of different types and solve problems of different complexities, with promising results (due to the integration of neural networks advantages into the provided methods) in modelling complex and non-linear wastewater treatment processes [1, 3, 28]. DL can learn from its own errors, while ML needs the human expertize, requiring much more computational power than ML [8]. According [3], DL is a technique based on Artificial Neural Networks (ANNs) through which a complex process can be modelled by learning from examples of the process inputs and outputs. Respectively, through DL are extracted essential features from raw input data by organizing the artificial neurons (computational processing units) into layers with a certain hierarchy. A DL model once trained for a certain task with measured data, it can be used to generate the process outputs from previously unknown input data. When the neurons are hierarchical connected (the connections weights supplies the model response) into a number of layers (connected by activation functions) they form a so-called deep neural network that can be trained (the inputs and outputs weights are adjusted) to optimize the neuron weights in order to increase the accuracy of the predicted outputs.

For wastewater treatment process modelling (by which are obtained data-driven type models) a wide range of DL algorithms (supervised or unsupervised learning techniques) is used, such as: RNNs, TCNs, FNNs, TDNNs, DBNs, GRNNs, LSTM, Convolutional Neural Networks (CNNs), Gated Recurrent Unit Networks (GRUs), Deep Reinforcement Learning (DRL) and other hybrid algorithms [24]. Such DL algorithms are applied for process simulation, respectively for predicting a WWTP effluent performance and also for predicting various effluent parameters (chemical oxygen demand-COD, biochemical oxygen demand-BOD, nitrate, etc.) [3, 12, 15]. Also, both supervised and unsupervised DL methods are used in process control and automation, in state estimation and soft sensing for estimated the process variables that are difficult to measure directly, data pre-processing (respectively, pattern discovery and decision-making process improving) and wastewater treatment design, optimization and control [3, 9, 10, 13, 22, 23, 30]. So, DL algorithms usage in WWTP processes comes with various benefits for the plant operators regarding the decision making process and also in the development of data-driven type applications dedicated to the plant operators.

From the mentioned DL algorithms, based on ANNs, with applications in wastewater treatment processes domain, in the present paper were analysed seven DL algorithms, respectively RNNs, TCNs, FNNs, TDNNs, DBNs, GRNNs and LSTM, in order to identify that DL method suitable for wastewater pH neutralization process modelling. These DL algorithms have been selected due to their use in similar non-linear processes and due to the fact that are able to handle large and complex data (as other DL algorithms do), being a limited selection of them, in order to test their performances in non-ideal circumstances, and to compare all of their results to evaluate their performance for this complex task [3, 5, 9, 12, 15, 22, 29].

RNNs are a type of DL algorithm, respectively are a type of ANNs that are using in the learning process training data (sequence data or time series data). To influence the current input and output (dependent by the previous elements), RNNs are using information from previous inputs. Between each network, they are sharing the same weight parameters, respectively the weights are adjusted once more through backpropagation through time (BPTT) algorithm (the model trains itself by determining the errors from its output layer to its input layer) in order to determine the gradients [14, 31]. According [29], RNNs are suitable for handling sequential data, being an excellent tool for developing predictive models for sequential big data in WWTPs.[14, 31]

According [32], TCNs are a type of ANN architecture that in order to identify temporal dependencies in data uses dilated convolutions, respectively it uses 1D convolutional architecture (dilated convolutions), each layer having information regarding all the previous layers outputs. In the training process it is used a residual block structure (containing a set of dilated convolutions), an addition operation (for adding the input to the block output) and non-linear activations. An advantage of TCNs is the parallel computing of the input sequence elements, with effects in the training and inference process time reducing. TCNs are deep neural networks that surpass in performance RNNs in task as time-series data, audio synthesis and handwritten recognition [20, 25]. According [32], TCNs is a class of DL models suitable to handle sequence data, having applications in time series forecasting, sequence classification, anomaly detection, etc.

FNNs are basically ANNs often used for classification, in which all the information is only passed forward, respectively the knots do not have loops. The data is passed to the input nodes, then to the hidden layers and to the output nodes (there is no possibly to send back information from the output node). As they enter into the layer, the inputs are multiplied with weights this way being obtained the sum (if the values rise above an established thresholds, the output is 1, otherwise is -1) parameter. Through back-propagation, the weights are adjusted and the networks hidden layers are adjusted according to the output values produced by the final layer. Some FNNs applications are automation and machine management, physiological feed-forward system, parallel feed-forward compensation, gene regulation and feed-forward, etc [18, 27].

According [17], TDNNs are a 1D convolutional type neural network, without pooling and with dilations, respectively is a multilayer ANN architecture used for patterns classification with shift-invariance, in modeling the context of each layer of a network, etc. Are similar to feedforward networks, with the only difference that the input weight has associated a tap delay line (fact that ensures the TDNNs dynamic response to time series input data), each TDNNs layer operating at a different temporal resolution.

DBNs are a type of DL algorithm that are using layers of stochastic latent variables (called feature detectors). The difference between DBNs and classical ANNs is given by the fact that DBNs can be generative and discriminative models. They are containing an input layer with one neuron per input vector, a number of intermediate layers and a final layer were outputs are generated using the probabilities derived from previous layers' activations. DBNs are a more efficient version of feedforward neural networks, used for video sequences, motion capture data, speech recognition, image recognition and much more [19].

GRNNs are a type of radial neural networks (a feed forward ANN type model), that uses the nonparametric regression in which the training samples are in fact the mean of the radial basis neuron. They can be used for prediction, regression, classification, for modeling dynamical systems, etc. As a structure, they present four connected layers (with a certain number of neurons), respectively the input layer, the pattern layer, the summation layer and the output layer. In the GRNNs training process (which is a very fast process-the main advantage of GRNNs) the goal is to find the position where mean squared error (MSE) in minimum. Also, the GRNNs learning process from training data is much faster than the time need to train a standard feed-forward network [4, 11].

LSTM is a type of RNN capable of learning long-term dependencies in sequence prediction problems, being useful for solving complex problems, such as speech recognition, machine translation, etc. They resemble with RNNs but in the case of LSTM each recurrent node is replaced by a so called memory cell which contains the internal state (a node with a self-connected recurrent edge of fixed weight 1). The RNNs have long-term memory (weights) and short term memory (ephemeral activations), while LSTM uses an intermediary type of storage under a memory cell form [31].

THE COMPARATIVE STUDY OF THE ANALYSED DEEP LEARNING METHODS

For the mentioned DL (ANNs based) algorithms testing, were used data for a series of four hundred and twenty-seven distinct scenarios (collected over one year and two months period of time) associated with the studied wastewater pH neutralization process from a Romanian refinery. The main objective was to analyse the evolution of the alkaline reactant ($Ca(OH)_2$) final flowrate (F2f) necessary for neutralizing an acid type pH, considering the acid reactant (H_2SO_4) flowrate (F1) which was maintained constant, the initial alkaline reactant flowrate (F2i) and also the initial (pHi) and the final (pHf) wastewater pH value. The pH neutralization process mathematical model and its validation for the flowrates and volume used in the studied plant from a Romanian refinery through a set of experiments are presented in [5, 6, 7]. The analysed data (selection) structure is presented in Table 1, data that make up the csv type file necessary for analysing the DL methods with Python 3.9 software and Tensorflow.

Table 1. *The analyzed data (selection) structure that compose the csv file [5, 6, 7]*

Sample	F1 [liters/hr]	F2i [liters/hr]	pHi [units]	pHf [units]	F2f [liters/hr]
1	260	6150	2.6	7	6595.84
2	260	5704	2.7	2.755	5750
3	260	5704	2.7	2.82	5800
…	…	…	…	…	…
426	260	6148	6.99	7	6149.84
427	260	6148	6.99	7	6150

The data from the csv type developed file were divided into three different sets of data, respectively three hundred and forty-two entries (three hundred eight were used for training, while thirty-four were used for validation) were used for ANN training and validation, and eighty-five were used for test dataset.

In order to obtain a fair comparison, all of the analyzed DL methods have a similar number of trainable parameters, respectively around one hundred and twenty thousand trainable parameters, these parameters reflect the components of an individual neuron that is be presented below (Figure 2), with the model being trained in one thousand epochs (the epochs number used for training each tested DL method was established through experimental testing, respectively through an error testing process), using Adaptive Moment Estimation (ADAM) algorithm and Mean Squared Error (MSE) function. A representation from the author perspective of a neural network architecture for the analyzed process is presented in Figure 2:

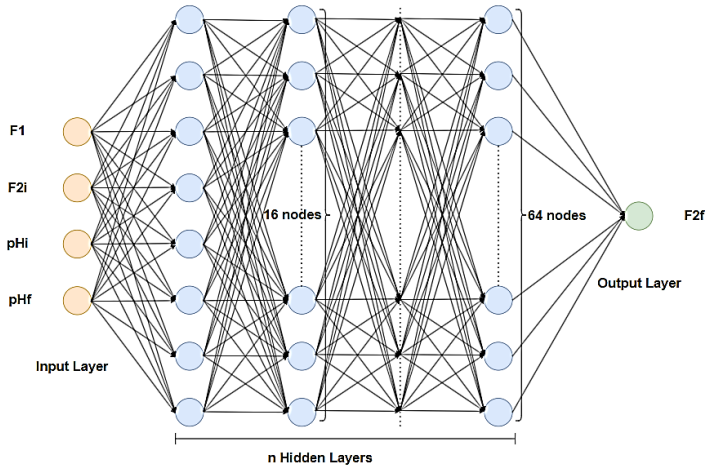

Figure 2. *The ANN proposed architecture*

In Figure 3, is presented, from the author perspective, the individual nodes of an ANN layer, where with beige color are represented the input nodes from the previous layer. Each input node has an input (I_1, I_2,…, I_n) and a weight (W_1, W_2, …,W_n), which alongside a bias are inputs to an activation function (in this case, a non-linear function called Rectified Linear Unit - ReLU), while the results of this function is the node output.

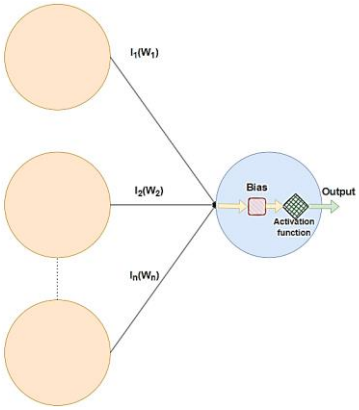

Figure 3. *The structure of an individual node*

The first DL method that was tested it was FNNs, the authors contribution being the ANN implementation that contains nine layers of densely interconnected nodes that start at four nodes and double in number up to two hundred and fifty-six nodes (it uses ReLU activation function), followed by a Dropout layer that ignores twenty percent of the data to prevent the network from overfitting, while the final layer is a dense layer that contains just one node as output. In comparison with the original FNNs, the improved method contains Dropout layers to prevent overfitting and the initial input layer was integrated into the dense layers. After it was trained for a thousand epochs, a test was conducted in order to measure the ANN parameters, and a graph was achieved in order to visualize the testing data. As such, Figure 4 represents the results between the real values obtained experimentally (represented in blue), and the predicted values (represented in orange).

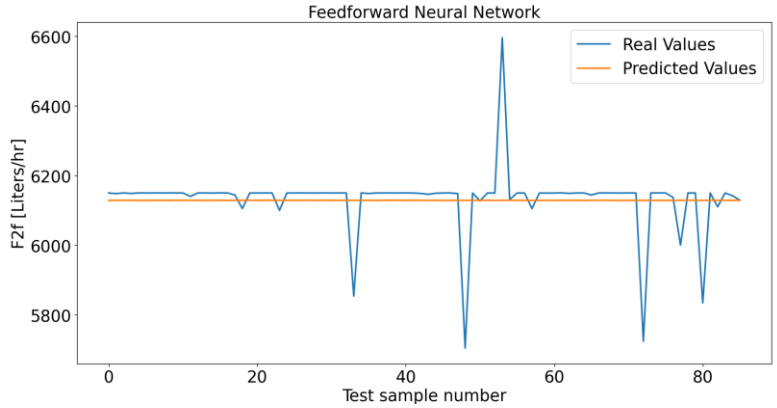

Figure 4. *FNN graph that shows the performance on the testing data*

The values from Figure 4 shows a clear lack of task understanding, as it has given similar output values ignoring the input values.

The TCNs is the second DL method that was implemented, the authors contribution being its implementation, that contains eight 1D Convolution layers that return the output filter starting at four and it doubles to a total nodes number of two hundred and fifty-six (it uses ReLU activation function), then dropping to sixty-four to keep the number of parameters identical. It is followed by a Dropout layer that ignores twenty percent of the data to prevent the network from overfitting, a Flatten Layer to flatten the output to the correct size, while the final layer is a dense layer that contains just one node as output.

In comparison with the standard TCNs, the improved method contains Dropout layers to prevent overfitting and the input shape was integrated into the convolution layers, with a given padding of *causal* and a standard *kernel_size* of 2. After it was trained for a thousand epochs, a test was conducted in order to measure the ANN parameters, and a graph was made with the intent to have a way to visualize the testing data. As such, Figure 5 represents the results between the real values found experimentally shown in blue, and the predicted values shown in orange. As it can be observed in Figure 5, the values show a pretty good performance.

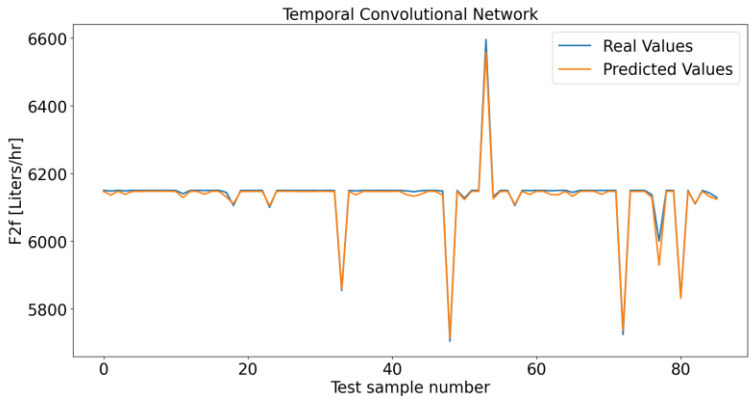

Figure 5. *TCNs graph that shows the performance on the testing data*

The RNNs is the third DL method that was tested, the authors contribution being its implementation that contains three layers of SimpleRNN layers, the first layer containing thirty-two nodes and having a return sequence, the second layer containing one hundred and twenty-eight nodes with a return sequence, while the last layer contains two hundred and fifty-six nodes without a return sequence, being the last SimpleRNN layer, followed by a Dropout layer that ignores twenty percent of the data to prevent the network from overfitting. Also, it contains a Flatten Layer to flatten the output to the correct size, while the final layer is a dense layer that contains just one node as output. In comparison with the standard RNNs, the improved method contains Dropout layers to prevent overfitting and has integrated the input shape into the RNN layers, with a set return sequence for better performances. It was tested to determine its parameters after training in thousand epochs, and a graph was created to provide a mean of visualizing the test results. Consequently, Figure 6 illustrates the differences between the actual values obtained experimentally, indicated in blue, and the predicted values, indicated in orange. As it can be observed, the graph's values presented in Figure 6 indicate generally good performance.

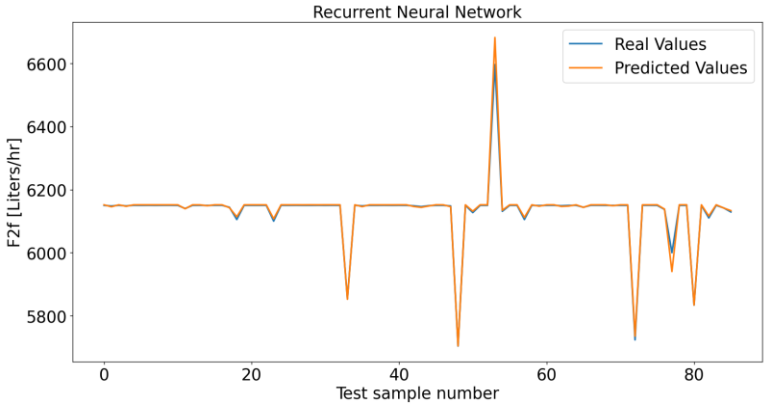

Figure 6. *RNNs graph that shows the performance on the testing data*

The Long Short-Term Memory (LSTM) is the fourth DL method that was implemented, the authors contribution being its implementation that contains a layer of one hundred LSTM nodes and Dropout of twenty percent of data to prevent overfitting. The first layer

of LSTM nodes has a return sequence, the layers that contain nodes are using the ReLU activation function, the final layer containing just one node as the output. In comparison with the standard LSTMs, the improved method contains Dropout layers to prevent overfitting and has integrated the input shape into the LSTM layers, alongside the return sequence for better performances. A test was made to determine the ANN parameters after has been trained for a thousand epochs, and a graph was created to provide a visual representation of the test results. The results between the actual values (indicated in blue), and the predicted values (indicated in orange), are thus depicted in Figure 7. As it can be observed, the LSTM performance is fairly good.

Figure 7. *LSTM graph that shows the performance on the testing data*

The GRNNs is the fifth DL method tested, the authors contribution being the implementation of five layers of densely interconnected nodes that start at sixty-four and double the number of nodes to two hundred and fifty-six nodes, the last layer having sixty-four nodes, using the Radial Basis Function (RBF) activation function, followed by a Dropout layer that ignores twenty percent of the data to prevent the network from overfitting with the final layer containing one node as the output. In comparison with the standard GRNNs, the improved method contains Dropout layers to prevent overfitting and has integrated the input shape in the dense layers. The ANN was tested to determine its parameters after a thousand epochs of training, and a graph was created to provide a means of visualizing the test results. Consequently, Figure 8 illustrates the differences between the actual values (indicated in blue), and the predicted values (indicated in orange). As it can be observed, the graph's values indicate generally good performance.

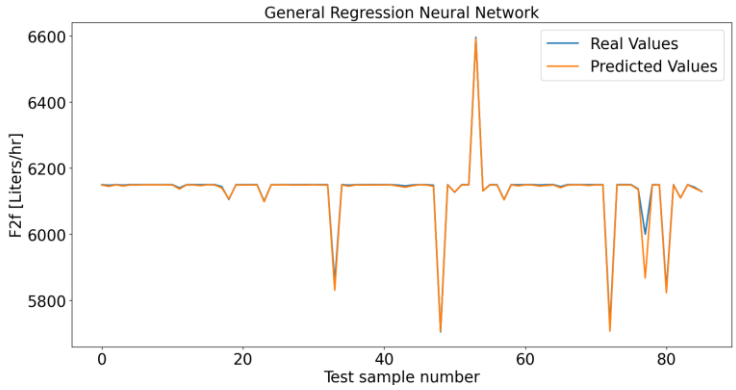

Figure 8. *GRNNs graph that shows the performance on the testing data*

The TDNN is the sixth DL method tested, the authors contribution being the configuration of five layers of densely interconnected nodes, starting at sixty-four nodes and double the number to two hundred and fifty-six nodes, the final layer being made from sixty-four nodes, using the ReLU activation function, followed by a Dropout layer to prevent the network from overfitting, a flattening layer, with the final layer containing one node as output. This DL method is similar to a FNN, but uses a different data format, as it requires a delay in order to process the data. In comparison with the standard TDNNs, the improved method contains Dropout layers to prevent overfitting and has integrated the input shape into the dense layers.

After it was trained for a thousand epochs, a test was conducted in order to measure the parameters of the ANN, and a graph was made with the intent to have a way to visualize the testing data. As such, Figure 9 represents the results between the real values shown in blue), and the predicted values (shown in orange).

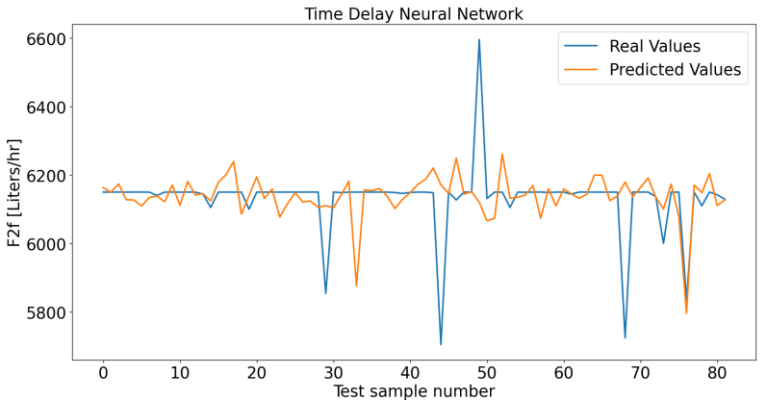

Figure 9. *TDNN graph that shows the performance on the testing data*

The values presented in Figure 9 show even worse performance that the FNN, as the data seems almost random. The DBNs is the last DL method that was tested., the authors contribution being the configuration of four layers of auto encoders, using the ReLU activation function for encoding, and Sigmoid activation function for decoding, followed by a Dropout layer of twenty percent to prevent the network from overfitting, a flattening layer, with the final layer containing one node as the output. This DL method uses four of these unsupervised individual networks trained in fifty epochs and then the entire model was trained for one thousand epochs. In comparison with the standard DBNs, the improved method contains Dropout layers to prevent overfitting and has integrated the input shape into the dense layers, with a different shape of each group of auto encoders, depending on the group needs.

This type of ANN was tested in order to determine its parameters after a thousand epochs of training, and a graph was created to provide a visual representation of the test obtained results (Figure 10). Consequently, Figure 10 illustrates the differences between the actual values (indicated in blue), and the predicted values (indicated in orange). As it can be observed, the values from the graph show pretty good performance.

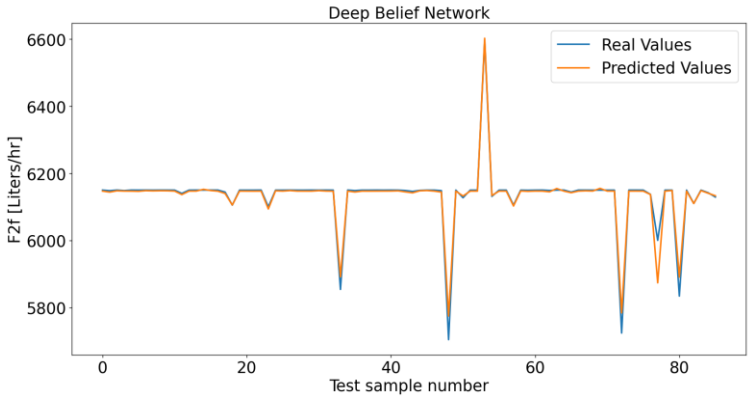

Figure 10. *DBN graph that shows the performance on the testing data*

In table 9, are centralized the metrics obtained for each analyzed DL method, respectively explained variance score (Ex. var. score), R^2 score, mean absolute error (Mean abs. err.), Mean Absolute Percentage Error (MAPE) and median absolute error (Med. Abs. err.).

Table 9*. The analyzed DL methods performance*

DL method	Ex. var. score	R^2 score	Mean abs. err.	MAPE	Med. abs. err.
FNNs	0.00043232	-0.00131538	41.72491143	0.006908723	21.243662109
TCNs	**0.99055991**	**0.98765102**	6.065544377	0.000989174	3.074375000
RNNs	0.98539030	0.98502588	3.832001726	0.000617391	1.832363281
LSTM	0.97941211	0.97941209	4.756412836	0.000777157	2.603603516
GRNNs	0.97693499	0.97564787	**3.524922102**	**0.000585175**	**1.100009766**
TDNNs	-0.18721870	-0.19238707	52.26995284	0.008613375	25.59919922
DBNs	0.961275065	0.961180891	6.867442315	0.001150351	3.063144531

It can be observed from Figure 4 and table 9, that FNNs has performed really poorly, being very far from the ideal solution for data prediction necessary for an optimal wastewater pH process modelling. While FNNs usually are not a bad choice, for the current task are not recommended, being available better DL solutions. From Figure 5 and table 9, it can be observed that the TCNs method has performed well, being a solid choice for data prediction and wastewater pH neutralization process modelling. While a regular Convolutional Neural Network (CNN) has applications in object detection and image classification, a TCN can also be used for the presented task of predicting the values of F2 final flowrate while knowing the initial inputs. Regarding RNNs, from Figure 6 and table 9, it can be observed that they have performed well, being a solid choice for wastewater pH process modeling. The LSTM and GRNNs methods has performed well (Figure 7, Figure 8 and table 9), being a solid choice for wastewater process modelling. From the obtained results presented in Figure 9 and table 9, it can be observed that the TDNNs has performed really poorly, being very far from the ideal solution for data prediction, not being a viable solution for wastewater neutralization

process modelling. While a TDNNs is not a bad choice for speech recognition, for the task in question is not a viable solution, other methods being more suitable for modeling such a process. Also, from Figure 10 and table 9, it can be observed that the DBNs method has performed well, being a solid choice for data prediction and wastewater neutralization process modelling.

In Figures 11-15, are compared each analyzed DL method using each obtained metrics (which green is highlighted the best method based on a specific metric), in order to determine which DL method from the analyzed ones is the most suitable for predicting the final F2 flowrate necessary to obtain a neutral wastewater pH, respectively the best DL method that can be used in modelling a such high nonlinear process, as is the wastewater pH neutralizing.

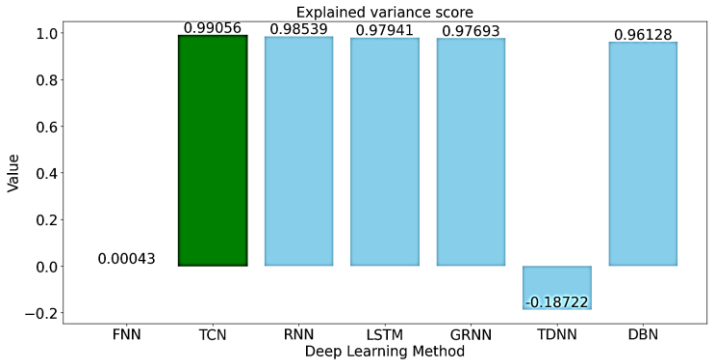

Figure 11. *Explained variance score comparison*

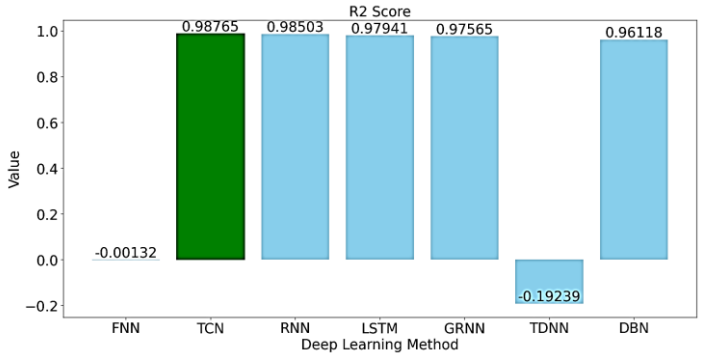

Figure 12. *R^2 score comparison*

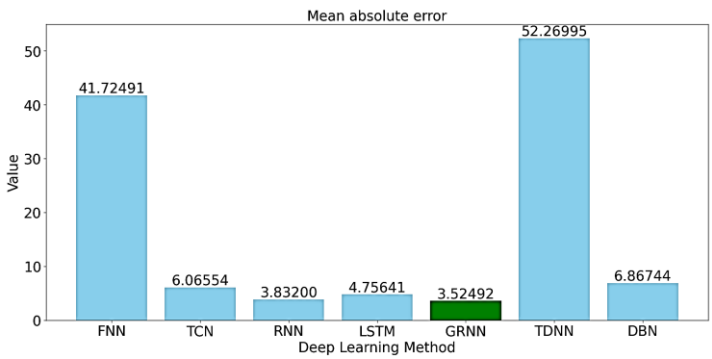

Figure 13. *Mean absolute error comparison*

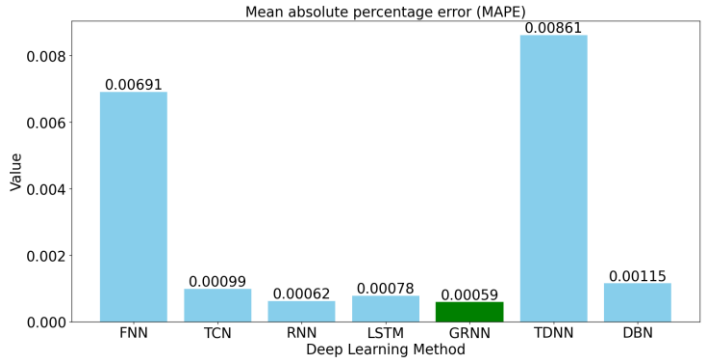

Figure 14. *Mean absolute percentage error comparison*

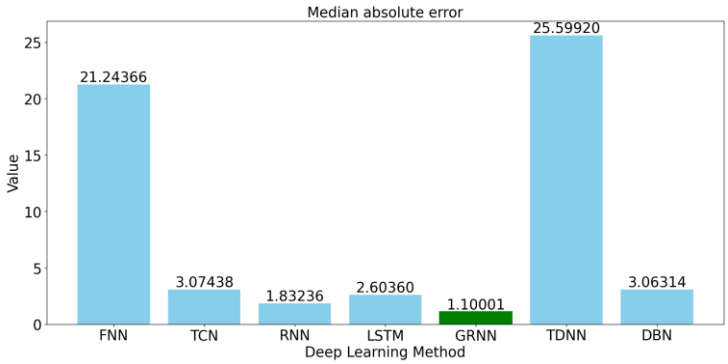

Figure 15. *Median absolute error comparison*

From the presented results (table 9, Figure 11-15), the best performer for the explained variance score metric is TCNs DL method, being a little bit better that RNNs, the worst performance being obtained for FNN and TDNN. The best performer for R^2 score metric is TCNs, being a little bit better that RNNs, the worst performance being also obtained for FNNs and TDNNs. The best performer for mean absolute error metric is GRNNs, being a little bit better that RNNs, the worst performance being again confirmed for FNNs and TDNNs. The best performer for MAPE metric is GRNNs, being a little bit better that RNNs, the worst performance being for FNNs and TDNNs. The best performer for median absolute error metric is GRNNs, being a little bit better that RNNs, the worst performance being for FNNs and TDNNs.

After comparing all the analyzed DL methods, the best performer out of them is GRNNs, with RNNs as the second best performer being slightly worse. Despite not being the best in a category, it is the second best performer for all the performance metrics compared.

CONCLUSIONS

The results presented in this paper, are promising. All seven DL methods has been tested and their performances have been analysed for the case of a dynamic and high nonlinear wastewater treatment process. So, the best DL method for the wastewater pH neutralization process modelling was identified to be General Regression Neural

Networks (GRNNs) and Recurrent Neural Networks (RNNs), while the worst have been identified to be the Feedforward Neural Networks (FNNs) and Time Delay Neural Networks (TDNNs). While Long Short-Term Memory (LSTM), Deep Belief Networks (DBNs) and Temporal Convolutional Networks (TCNs) are not the worst choices, their performance isn't as high as the other two DL methods (being solid choices that can give accurate data). The size of the models has been significantly restricted in an attempt to maintain a fair comparison without architectures of millions of trainable parameters, as the goal was to compare the performances between DL methods, not to create the most efficient neural network architecture for this goal.

The author contribution was the implementation of seven DL methods using Python 3.9 software and Tensorflow and the identification of a suitable DL method (in this case, GRNNs DL method being found the best) that can be applied for predicting the reactant (F2-hydrated lime) final flowrate necessary to obtain a neutral pH, in wastewater pH neutralization process modelling, respectively in developing data-driven type models dedicated to high nonlinear treatment processes from a WWTP, with real benefits for the plant human operator in the decision making process.

It was a continuation of the work presented din paper [5], were the best ML regression method that can be used for modelling the wastewater pH neutralization process was identified to the gradient boosting regression (GBR), while in the present paper the best DL based ANN method that can be used with the same goal, was identified to be the GRNNs method.

As future work the size and the complexity of the best performing architecture can be scaled up and can be modified in order to supply extremely accurate data in order to be integrated into a system that can control the entire wastewater pH neutralization process. Also, a comparison of the best DL methods with other methods (non-AI based, and other AI methods) can be achieved.

REFERENCES

[1] Abdullah N.H.S., Karsiti M.N., Ibrahim R., A review of pH neutralization process control, 4th International Conference on Intelligent and Advanced Systems (ICIAS2012), Kuala Lumpur, Malaysia, pp. 594-598, 2012.

[2] Agachi Ş., Automatizarea proceselor chimice, Casa Cărţii de Ştiinţă, Cluj-Napoca, pp. 259-263, 1994.

[3] Alvi M., Batstone D., Mbamba C.K., Keymer P., French T., Ward A., Dwyer J., Cardell-Oliver R., Deep learning in wastewater treatment: a critical review, Water Research, vol. 245, pp. 1-15, 2023.

[4] Amjad J., Lyu Z., Rodrigues M., Regression with Deep Neural Networks: Generalization Error Guarantees, Learning Algorithms, and Regularizers, 29th European Signal Processing Conference (EUSIPCO), Dublin, Ireland, pp. 1481-1485, 2021.

[5] Cărbureanu M., Mihalache S.F., Zamfir F., Machine Learning Methods Applied for Wastewater pH Neutralization Process Modeling, 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1-6, 2022.

[6] Cărbureanu M., Neuro-fuzzy expert system for wastewater treatment processes control, Doctoral thesis, Petroleum-Gas University of Ploiesti, 2014.

[7] Cărbureanu M., The Development of a Neuro-Fuzzy Expert System for Wastewater pH Control, Journal of Control Engineering and Applied Informatics, vol. 16, no. 4, pp. 30-41, 2014.

[8] Chauhan, N.K., Singh K., A review on conventional machine learning vs deep learning, International conference on computing, power and communication technologies (GUCON), India, pp. 347-352, 2018.

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

[10] Corominas L., Garrido-Baserba M., Villez K., Olsson G., Cortés U., Poch M, Transforming data into knowledge for improved wastewater treatment operation: A critical review of techniques, Environ. Model. Softw., vol.106, pp. 89-103, 2018.

[11] Currit, N., Inductive regression: overcoming OLS limitations with the general regression neural network, Computers, environment and urban systems, vol. 26, no. 4, pp. 335-353, 2002.

[12] El-Rawy M., Abd-Ellah M.K., Fathi H., Ahmed A.K.A., Forecasting effluent and performance of wastewater treatment plant using different machine learning techniques, J. Water Process Eng., vol. 44, 2021.

[13] Filipe J., Bessa R.J., Reis M., Alves R., Póvoa P., Data-driven predictive energy optimization in a wastewater pumping station, Appl. Energy, vol. 252, 2019.

[14] Grossberg, S., Recurrent neural networks, Scholarpedia 8, no. 2, 2013.

[15] Guo H., Jeong K., Lim J., Jo J., Kim Y.M., Park J.P, Kim J.H., Cho K.H., Prediction of effluent concentration in a wastewater treatment plant using machine learning models, J. Environ. Sci., vol. 32, pp. 90-101, 2015.

[16] Marinoiu V., Paraschiv N., Automatizarea proceselor chimice, Vol. I+II, Editura Tehnică, Bucureşti, pp.320-325, 1992.

[17] Meng, H., Bianchi-Berthouze N., Deng Y., Cheng J., Cosmas J.P., Time-delay neural network for continuous emotional dimension prediction from facial expression sequences, IEEE transactions on cybernetics, vol. 46, no. 4, pp. 916-929, 2015.

[18] Montana D.J., Lawrence D., Training feedforward neural networks using genetic algorithms, IJCAI, vol. 89, no. 1989, pp. 762-767, 1989.

[19]Movahedi F., James L.C., Ervin S., Deep belief networks for electroencephalography: A review of recent contributions and future outlooks, IEEE journal of biomedical and health informatics, vol. 22, no. 3, pp. 642-652, 2017.

[20] Pandey A., Wang D., TCNN: Temporal Convolutional Neural Network for Real-time Speech Enhancement in the Time Domain, ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp. 6875-6879, 2019.

[21] Pietrzyk D. J., Frank C.W., Chimie analitică, Editura Tehnică, Bucureşti, pp.110-140, 1989.

[22] Qiao J., Huang X., Han H., Recurrent Neural Network-Based Control for Wastewater Treatment Process. In: Wang, J., Yen, G.G., Polycarpou, M.M. (eds) Advances in Neural Networks, Lecture Notes in Computer Science, vol. 7368. Springer, Berlin, Heidelberg, 2012.

[23] Qdais H.A., Hani K.B., Shatnawi N., Modeling and optimization of biogas production from a waste digester using artificial neural network and genetic algorithm, Resour. Conserv. Recy., vol. 54, pp. 359-363, 2010.

[24] Sarker, I.H., Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. SN COMPUT. SCI., vol. 2, 2021.

[25] Sharma A., Jayagopi D.B., Towards efficient unconstrained handwriting recognition using Dilated Temporal Convolution Network, Expert Systems with Applications, vol. 164, 2021.

[26] Skoog D.A., West D.M., Holler F.J., Fundamentals of Analytical Chemistry, Fifth Edition, Saunders College Publishing, pp.182-230, 1988.

[27] Svozil, D., Kvasnicka, V., Pospichal, J., Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems, vol. 39, no. 1, pp. 43-62, 1997.

[28] Wang X., Zhao Y., Pourpanah F., Recent advances in deep learning. Int. J. Mach. Learn. & Cyber, vol.11, pp 747–750, 2020.

[29] Wongburi P., Park J., Prediction of Wastewater Treatment Plant Effluent Water Quality Using Recurrent Neural Network (RNN) Models, Water 2023, vol. 15, no. 19, pp. 3325, 2023.

[30] Yan P., Gai M., Wang Y., Gao X., Review of soft sensors in anaerobic digestion process Processes, vol. 9, no.8, pp. 1434, 2021.

[31] Yu Y., Si X., Hu C., Zhang J., A review of recurrent neural networks: LSTM cells and network architectures, Neural computation, vol. 31, no. 7, pp. 1235-1270, 2019.

[32] Zhao W., Gao Y., Ji T., Wan X., Ye F., Bai G., Deep temporal convolutional networks for short-term traffic flow forecasting, IEEE Access, vol. 7, pp. 114496-114507, 2019.

Received: September 2024; Revised: October 2024; Accepted: October 2024; Published: October 2024