Temporal Convolutional Network-based Approach for Forecasting Fluctuations Differential Pressure in Reverse Osmosis Systems

The Son Phan[1], Thanh-Ha Do[1], Phuc Do [2]

[1] *HaNoi University of Science, Hanoi, Vietnam*
phantheson_t65@hus.edu.vn
dothanhha@hus.edu.vn

[2] *University of Lorraine, Nancy, France*
phuc.do@univ-lorraine.fr

ABSTRACT

Providing forecasts of pressure fluctuations and changes will aid in selecting appropriate maintenance strategies to optimize efficiency and costs. This paper presents a deep-learning-based model to forecast the degradation evolution of membrane biological fouling in RO (Reverse Osmosis) systems. Although applying deep learning in forecasting still faces many challenges, applying convolutional operations in convolution 1D has yielded promising results for sequential data, particularly time series data. Thus, in this paper, we study and develop the 1D convolution operation-based Temporal Convolutional Network (TCN) model to predict pressure dynamics at both ends of the RO vessel. In addition, since the deep learning technique has yet to be widely explored in this field, thus we also need to pre-process the data collected from the Carlsbad Desalination Plant in California, such as the proposed model can identify complex relationships between timestamps and pressure features. The experiment results were evaluated and compared with other existing models, such as LSTM, CNN & LSTM, and GRU. The results show that the TCN-based prediction model had the slightest error in the test dataset.

1. INTRODUCTION

Water covers approximately 71 % of the Earth's surface, and more than 97 % of the Earth's water is saltwater. Under population growth, the need for clean water is highly critical. Various methods have been developed to generate clean water for industrial and domestic use to meet this demand, one such method being desalination Nour et al., 2022 found. Therefore, RO (Reverse Osmosis) technology-based desalination plants have been put into operation. Nour et al., 2022 show that the issue arises when these plants face system impairment after operation due to membrane fouling. Depending on the type of accumulated residue, membrane fouling can be categorized into particle fouling, organic fouling, inorganic fouling, and biofouling, with biofouling being considered the most severe and challenging to solve.

One of the common causes of biofouling is algal blooms (van Rooij, Scarf, and Do, 2021), (Villacorte et al., 2017) and (Jiang, Li, and Ladewig, 2017). Organic compounds produced during algal blooms are the leading cause of biofouling on membrane surfaces. These compounds create a slippery layer on the membrane surface, increasing the salts and pressure passing through the filter membrane. When these membranes become fouled, they can reduce filtration efficiency or, more seriously, damage the membrane system (Koutsakos and Moxey, 2007) found. There are three ways to improve membrane longevity: (a) membrane performance monitoring, (b) membrane cleaning (clean-in-place (CIP) method) (Koutsakos and Moxey, 2007; van Rooij, Scarf, and Do, 2021), and (c) membrane replacement (Koutsakos and Moxey, 2007; van Rooij, Scarf, and Do, 2021).

Many works have been proposed to mitigate the impact of biofouling, and some of them have yielded positive results. In the direction of maintaining membrane biological fouling, there are many causes. However, biological fouling is challenging to resolve, and the algal bloom is the root cause of membrane fouling (Nour et al., 2022). (Koutsakos and Moxey, 2007) proposed methods for monitoring membrane systems and maintenance options for the system. Maintenance of this system includes cleaning the membranes with chemical solutions, known as the CIP method. Replacing and rearranging the elements is essential when the system be-

comes heavily fouled. V.Rooij *et al.* have provided a solution to monitor the system's state. This method builds mathematical models based on extensive knowledge and experience in the desalination field. Statistical methods are used to estimate parameters for the models. Digital Twins (Sharma et al., 2022) is employed to implement the mathematical models. The results achieved are auspicious.

Besides traditional statistical models such as ARIMA, VAR, etc., deep learning models have recently been used to address problems due to their ease of implementation. Famous deep learning models include RNN, LSTM, GRU, GNN, and more recently, the Transformer of Yong Liu *et al.* (Yong et al., 2022) has emerged. The introduction of 1D convolutional layers (Kiranyaz et al., 2019) and the Temporal Convolutional Network (Lea et al., 2016) (TCN) has also yielded some promising results.

Operating and maintaining the system is highly expensive (Asif et al., 2021; van Rooij, Scarf, and Do, 2021), necessitating a strategy for monitoring the system's state to derive optimal maintenance solutions. This paper focuses on predicting pressure evolution in the desalination system using RO technology. Moreover, the paper develops a TCN-based model to predict pressure dynamics at both ends of the RO vessel. This model forecasts target variable values. In addition, we propose new techniques to present the relationship between the positions of elements or sockets, the operation status of the train, and the target variable and stabilize the normalized differential pressure (NDP) value.

The organization of the paper is as follows. Section (2) presents the desalination plant system. Section 3 describes the Temporal Convolutional Networks and proposed techniques in data processing. The experiment results on the performance of the proposed TCN-based prediction method are shown in Section 4. Finally, Section 5 presents the conclusion and future works.

2. DESALINATION PLANT SYSTEMS

The Carlsbad desalination plant system consists of over 2000 vessels. These vessels are evenly divided into components we call RO trains. Specifically, we have 14 independently operating trains. These vessels operate in parallel with the trains. Within these vessels are eight serially connected sockets, each filled with membrane elements (Fig. 1). Typically, these membrane elements are spirally wound, allowing seawater to be pushed through the membrane surface of all elements from the first element's feed source to the last element's tail end.

This vessel is divided into eight sockets, each filled by a membrane element rolled in a spiral form. Water is pumped into the vessel in a high-pressure sequence from the first to the tail element.

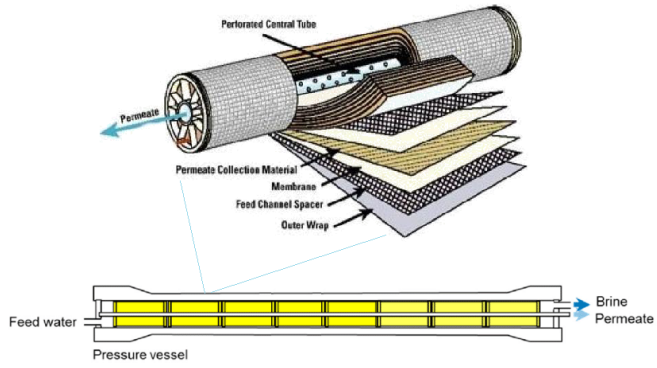


Figure 1. The structure of a vessel in an RO train. The vessel is divided into eight sockets, each filled by a membrane element rolled in a spiral form. The water is pumped into the vessel with a high-pressure sequence from the first to the tail element, from (Rooij, 2022)

2.1. Membranes Biofouling

Algal blooms significantly accelerate the biofouling process in RO membranes (van Rooij, Scarf, and Do, 2021). The quality of RO feed water reflects the influence of these blooms. The primary cause of accelerated degradation is the organic matter produced during algal blooms, known as algae-derived organic matter (AOM), particularly transparent exopolymer particles (TEP). As algae cells perish, they release AOM, which adheres to clean membranes and even more to fouled ones, intensifying biofouling.

TEP absorption manifests as a slimy substance clinging to surfaces. Dinoflagellates, a phytoplankton responsible for algal blooms, produce considerable quantities of TEP once nutrients are depleted. Consequently, biofouling-induced degradation continues beyond the duration of an algal bloom.

2.2. Membranes Restoration

When a membrane becomes fouling, measures must be taken to improve its state. A vessel will have various methods for restoring the membrane, including:

1. Clean-In-Place method (CIP) that includes: C_1, a cleaning method in which the membrane is cleaned with a high pH solution (NaOH solution), followed by a low pH solution (HCl solution) (Rooij, 2022). C_2 is a method in which the elements are soaked in a sodium bisulfate ($NaHSO_4$) solution, followed by the application of C_1.
2. C_3 is a method that combines the redistribution of the structure of the elements and cleaning. In this method, the first element (S_1), which usually has the highest biomass (most fouling), is removed for cleaning and then placed at the bottom of the vessel. The elements from S_2 to S_8 are each moved one position forward (Rooij, 2022).
3. Swapping and redistributing the vessel structure is a

method that includes planned restructuring of the positions of the elements, sometimes involving replacing a few positions with new ones (Rooij, 2022).

4. Unidentified correction, e.g., Low Salinity flushing, depressurization, or instrument adjustment.

Each method yields different results; therefore, having a predefined plan will help achieve optimal outcomes in terms of efficiency and cost.

3. DEEP LEARNING-BASED APPROACH FOR FORECASTING FLUCTUATION DIFFERENTIAL PRESSURE IN RO SYSTEM

This section describes the proposed model developed from the TCN model, which uses different convolution operations and multiple blocks instead of one TCN layer. In addition, the rule-based new techniques for processing data are also described in more detail.

3.1. Long short-term memory - LSTM

Long short-term memory (LSTM) (Alex and Alex, 2012) is a type of recurrent neural network (RNN) aimed at dealing with the vanishing gradient problem (Sherstinsky, 2018)present in traditional RNNs. Its advantage over other RNNs, hidden Markov models, and other sequence learning methods is its relative insensitivity to gap length. It aims to provide a short-term memory for RNN that can last thousands of timesteps, thus "long short-term memory." It applies to the classification, processing, and prediction of data based on time series, such as handwriting, speech recognition, machine translation, and speech activity detection.

One can view the cell state (Fig. 2) as a kind of long-term memory that retains at least a part of the information in earlier states by using a combination of partial forgetting and increment operations on the previous cell states. Specifically, the LSTM operation follows:

Forget gate: This function determines how much important information from the previous cell state should be retained. Since the sigmoid function σ outputs values in the range [0, 1], where one indicates keeping all the information, the forget gate can be calculated as (1).

$$F_t = \sigma(H_{t-1} \cdot W_{hf} + X_t \cdot W_{fx} + b_f) \tag{1}$$

Update Cell State: This is the process of updating the cell state information along with the retained information from the forget gate (8), (3), (4):

$$I_t = \sigma(W_{it} \cdot H_{t-1} + X_{ix} \cdot X_t + b_i) \tag{2}$$

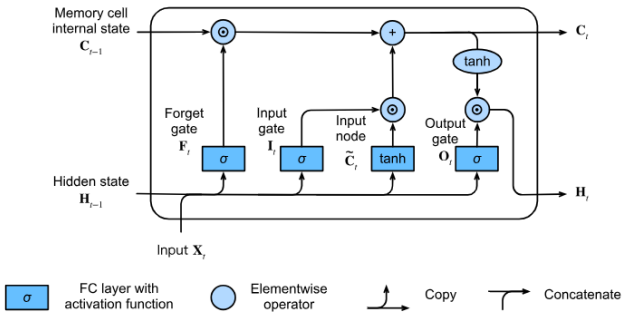


Figure 2. **Long short-term memory** , information have store in cell state. The operation of LSTM is similar to that of RNN; however, the significant difference that LSTM brings is in the hidden state. LSTM can store information over a long period, helping to solve the vanishing gradient issue.

$$\tilde{C}_t = \tanh(W_{ch} \cdot H_{t-1} + W_{cx} \cdot X_t + b_c) \tag{3}$$

$$C_t = C_{t-1} \odot F_t + I_t \odot \tilde{C}_t \tag{4}$$

Output State: This function helps compute the value of the hidden state (or provide the final prediction result if it is the final cell) (5), (6):

$$O_t = \sigma(W_{oh} \cdot H_{t-1} + W_{ox} \cdot X_t + b_o) \tag{5}$$

$$H_t = \tanh(C_t) \odot O_t \tag{6}$$

3.2. Gated Recurrent Unit - GRU

Gated Recurrent Unit - GRU maintains the reset and update gates. The reset gate (7) helps control how much of the previous state is retained. Similarly, the update gate (8) allows control over how much of the new state will resemble the old state, given the current time step input X_t and the previous hidden state H_{t-1}.

$$R_t = \sigma(W_{xr} \cdot W_t + W_{hr} \cdot H_{t-1} + b_r) \tag{7}$$

$$Z_t = \sigma(W_{xz} \cdot W_t + W_{hz} \cdot H_{t-1} + b_z) \tag{8}$$

Operation of Reset Gate

$$\tilde{H}_t = \tanh(W_{xh} \cdot X_t + (R_t \odot H_{t-1}) \cdot W_{hh} + b_h) \tag{9}$$

Similar to RNNs (Sherstinsky, 2018), GRUs maintain a hidden state and a candidate hidden state $\tilde{H}$ (9). The information from H_{t-1} passes through the reset gate R_t, adjusting the information retained. If R_t is close to 1, the result will be sim-

ilar to an RNN: $\tilde{H}_t = \tanh(W_{xh} \cdot X_t + H_{t-1} \cdot W_{hh} + b_h)$, if R_t is close to 0, the hidden state will be the output of a multi-layer perceptron with X_t as the input.

Operation of Update Gate

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \tag{10}$$

This gate (10) determines the similarity between the new state H_t and the old state H_{t-1}, as well as the extent to which the potential hidden state $\tilde{H}_t$ is used. The value of H_t will be determined by Z_t and $\tilde{H}_t$. If the values in the update gate Z_t are equal to 1, we retain the old state. In this case, the information from X_t is essentially ignored, equivalent to skipping the time step t in the dependency chain. Conversely, if Z_t is close to 0, the hidden state H_t will be close to the potential hidden state $\tilde{H}_t$. These designs can help us address the vanishing gradient problem in RNNs and better capture long-term dependencies in time series.

3.3. TCN prediction approach

The TCN (Temporal Convolutional Network) is a neural network that processes data in the type of time series data and is model (Shaojie, Zico, and Vladlen, 2018) has been proven effective in various tasks such as time series prediction and sequence classification. This architecture combines 1D convolutional layers with techniques such as causal convolutions and residual connections (see Figure (**??**)).

In general, TCN model (Shaojie, Zico, and Vladlen, 2018) satisfies two fundamental rules: 1) the input and output of the model must be the same size, and 2) the model must ensure that there is no information from the future in the current prediction step. Convolutional layers are padded with (kernel_size - 1) zeros to achieve the first criterion. This padding is placed on the left side of the data to reach the second criterion: the model only depends on historical data before the prediction point.

Dialated Convolutions: The causal convolution can only look at several previous elements in the sequence. Thus, dilated convolutions are employed in this paper since they help the model learn information from further distances in the data and focus on essential time steps while ignoring unimportant ones.

For a 1-D sequence input x and a filter f, the dilated convolution operation at element s of the sequence $F(s)$ is defined as a sum over the filter (k) multiplied by the input elements at specific positions determined by the dilation factor d (see Equation11).

$$F(s) = \sum_{i=1}^{k-1} f(i) \cdot X_{s-d \cdot i} \tag{11}$$

Larger filter sizes (k) and increasing the dilation factor (d) are two ways to improve the receptive field of the Temporal Convolutional Network (TCN). Larger filter sizes allow the model to consider more elements in the sequence while increasing the dilation factor, which enables the model to capture information from further back in the sequence.

Residual Connections The residual block He et al., 2015 helps the TCN model control. The residual block He et al., 2015 helps the TCN model control the amount of information, minimizing the vanishing gradient problem and enabling the model to learn more effectively and deeply. A residual block contains a branch leading out to a series of transformations F, whose outputs are added to the input x of the block.

$$\sigma = activation(x + F(x)) \tag{12}$$

The TCN has two layers of dilated convolution and nonlinearity within a residual block. In addition, To effectively predict pressure dynamics at both ends of the RO vessel and replace causal convolutional with dilated convolution, we propose a deep-learning-based model with two TCN blocks. The input data passes through the first TCN block, and its output is the repeated end timestamp, which is used as the input for the second TCN block as the output block. The proposed model with the first TCN block has 80 filters, a kernel size of 3, dilation rates of [1, 2, 4, 8, 16, 32], and a dropout rate of 0.1 for the first TCN layer and second TCN layer. The final layer will be fully connected at each timestamp.

More specifically, with its two TCN blocks, the proposed model is designed to provide precise predictions for time series data (see Figure 3). The first TCN block, comprising two TCN layers, is tailored to learn complex data structures within the time series. The first TCN layer takes the time series as input and returns the hidden states at each step, while the second TCN layer returns the hidden state at the last step. The output of the first TCN block is then multiplied and passed through the second TCN block to make predictions for the following days of the time series. Using a specific number, the proposed approach helps minimize the accumulation of errors in the projections, ensuring the precision of the model's predictions.

The model aims to minimize discrepancies between output and observer values. The proposed model employs the Mean Square Error, a robust and widely used loss function to achieve this. This choice of loss function underscores the model's commitment to accuracy and precision in its predictions.

3.4. Data Preprocessing

The dataset used in this paper comes from the Carlsbad desalination plant in California. The data comprises 14 trains

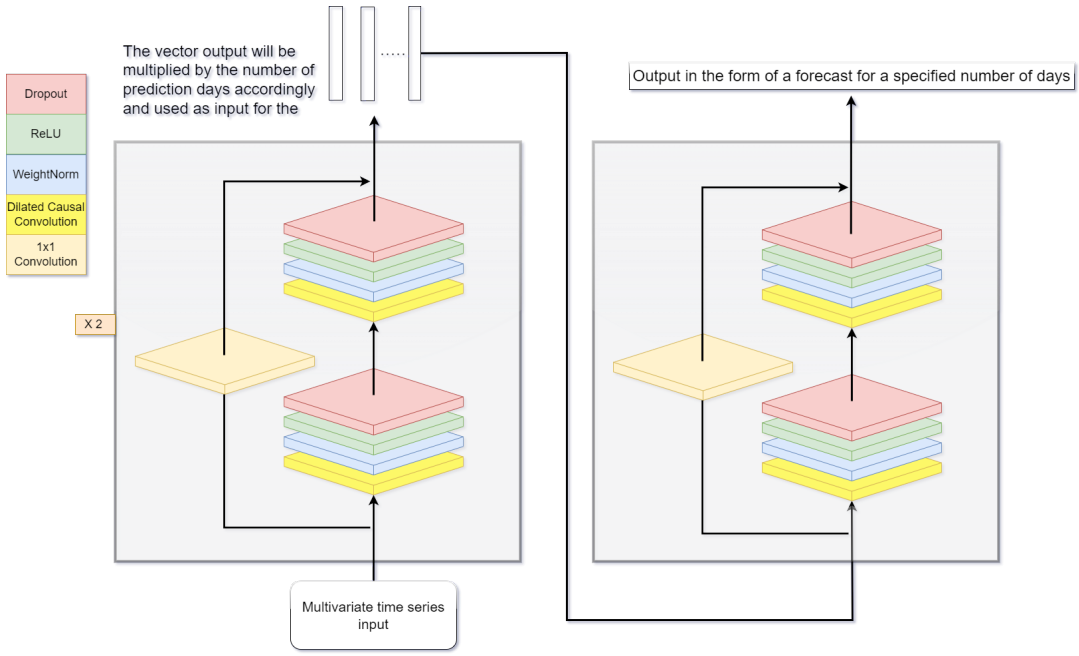


Figure 3. Proposed models based on deep learning seq-to-seq architectures: Input data will be organized as multivariate time series data. The first TCN block (encoder block) consists of 2 TCN layers with kernel size = 3, dropout rate = 0.1, dilated factor = [2,4,8,16,32], and the output is the final output vector, which is then scaled according to the number of forecast days and fed into the second TCN layer to predict the number of required days.

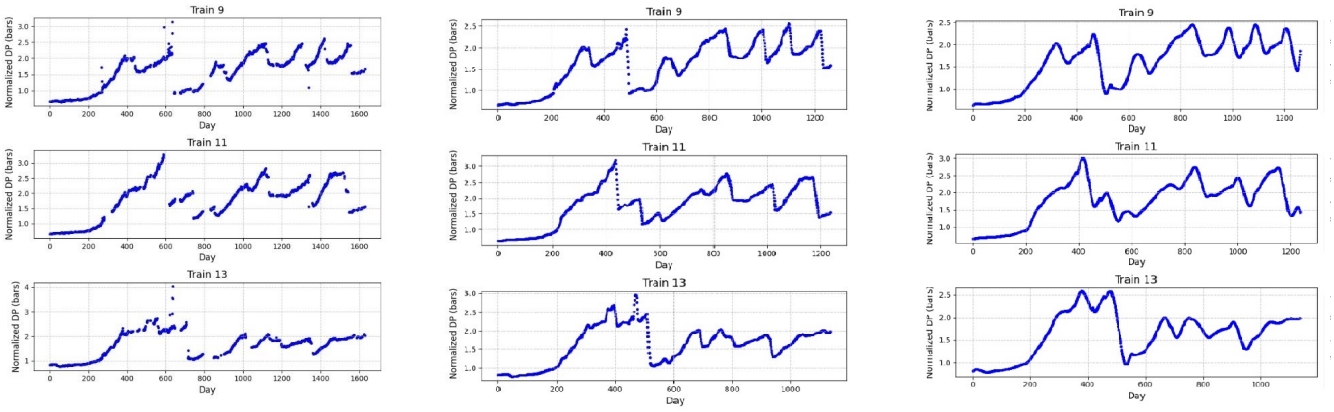
collected from 2015 to 2020. The dataset uses two kinds of values: the target variable Normalized Differential Pressure (NDP) and Train Status. The Train Status are values representing the operational condition of the trains, in which the following values should be noted: *Operation,* C_1*,* C_2*,* C_3*, Permutations*. The value *Permutations* consists of sequences of numbers from 1 to 8, and the letter 'N' represents that the element at that position is replaced with a new one. For instance, value '324N5678' is interpreted as element three being placed in the first position, element two remaining in the second position, element four is placed in the third position, a new element replaces the one in the fourth position, and the elements in positions 5, 6, 7, and 8 remain unchanged.

To present the relationship between the positions of elements or sockets, the operating status of the train, and the target variable, we propose some rule-based techniques as follows:

1. Generate additional features $S_i, (i = 1, ..8)$ representing the values of the elements within the trainfor example, the value "32405678" with 0 corresponding to replacing the element at position i. The values from Train Status will be converted to *Permutation* for the element restructuring method.
2. The C_3 method is described by values from S_1 to S_8 as follows: "23456781". It should be noted that the values of the sockets will be taken from the most recent membrane restoration event when the system is operational.
3. Typically, maintenance methods involve system downtime for a few days (3 - 5 days) to ensure temporal consistency. In this paper, we merge these days into one, during which the target variable receives NaN values. We then fill in the missing values with the NDP value from the previous period.
4. The Train Status values will be encoded using One-Hot Encoding, which transforms categorical variables into binary vectors $(0 - 1)$, simplifying data processing and computation. In more detail, each categorical variable with a set of labels $L = \{l_1, l_2, \ldots, l_n\}$ is represented as a vector v_i of length equal to the number of labels in the set, where only one element is 1 (corresponding to the chosen label). All other elements are 0.
5. Data is normalized to the range $(0, 1)$. For Train Status, the paper performs One-Hot Encoding.

We employ advanced data smoothing techniques to ensure the precision and reliability of our data analysis. These techniques, namely the SavitzkyGolay filter (B, 2020) and the Simple Moving Average (Rooij, 2022), are designed to stabilize the NDP values (Figure 4). We compare the smoothing of different smoothing methods to the original data. By smoothing the data, we make it easier for the model to learn the data trends, thereby enhancing the accuracy of our analysis.

A SavitzkyGolay filter (S-G filter) is a digital filter that can be applied to a set of digital data points to smooth the data, that



(a) (b) (c)

Figure 4. The Normalized Differential Pressure (NDP) values from Train 9 to Train 13, with data collected from 2015 to 2020. The sudden decrease in NDP values occurs when the system undergoes membrane restoration interventions. We compare the other smoothed data to the original data. Fig.4a The original data shows significant fluctuations and the presence of outliers; Fig.4b the data smoothed using a Moving Average (MA) with a window size of 10; Fig.4c the data smoothed using a Savitzky-Golay (SG) filter with a window size of 150 and a 4th-degree polynomial.

is, to increase the precision of the data without distorting the signal tendency. The S-G filter operates by approximating real values with a polynomial. Suppose we have a set x_j, $j = 1 \ldots n$ data points. The S-G filter smooths the data using a polynomial of degree k within a window of w data points. This polynomial has the form: $\hat{y}_i = a_0 + a_1 z + a_2 z^2 + \ldots + a_k z^k$. z is a vector that takes values from $\frac{1-w}{2}$ to $\frac{w-1}{2}$, or $\hat{Y} = J \cdot a$, where J is the Vandermonde matrix, and a is the coefficient of the polynomial. The coefficients a will be solved following Equation (13).

$$\hat{a} = \arg\min_a \|J \cdot a - Y\|^2 \tag{13}$$

A Simple Moving Average (SMA) is the unweighted mean of financial applications' previous k data points. However, in science and engineering, the mean is normally taken from an equal number of data on either side of a central value. SMA ensures that variations in the mean are aligned with the variations in the data rather than being shifted in time. An example of a simple equally weighted running mean is the mean over the last k entries of a data set containing n. Let those data points be $\{p_1, p_2, \ldots, p_n\}$. The mean over the last k data points is denoted as MA and calculated as in Equation (14).

$$\text{SMA} = \frac{\sum_{i=n-k+1}^{n} p_i}{k} \tag{14}$$

4. EXPERIMENT RESULTS

The dataset used in this paper comes from the Carlsbad desalination plant in California that comprises 14 trains collected and is pre-processing as described in Section 3.4.

For the training model, 80 % of the data in each train is used as the training and validation dataset. The last ten percent of each train is the testing set.

Three measures are used to evaluate the system's performance: RMSE (Root Mean Squared Error), MAPE, and R^2. The values are calculated for each train, and the final value is the average value of all trains. RMSE quantifies the distance between data points in a regression model and the actual data. RMSE calculates the square root of the squared error average between predicted and actual values. Thus, it estimates the average distance between data points in the regression model and the actual data. Mean Absolute Percentage Error (MAPE) measures the average absolute percentage difference between actual and forecasted values, providing insights into the forecasting model's overall performance. R-squared (R^2) or the coefficient of determination) is a statistical measure in a regression model that evaluates how well the data fit the regression model.

We train the model for 400 epochs. We employ an early stopping mechanism during training to prevent overfitting the data. The batch size is set to 64. We use optimizer Adam with learning rate = 0.001 to train the model, and the activation function is the ReLu function.

The proposed model is evaluated and compared to other deep-learning models, such as LSTM, GRU, etc. All models use the same dataset after applying the pre-processing stage, and the optimal parameters are the number of units = 80, dropout rate = 0.1, and kernel size of the convolution = 3.

In this paper, we designed three groups of experiments. The first one aims to evaluate the impact of window size t on the

prediction results of the target variable. In this experiment, the smoothing techniques, such as the S-G and SMA filters, have not been applied yet. Table 1 indicates that a window size of $t = 120$ yields relatively stable results with $RMSE = 0.629$ and $R^2 = 0.371$.

Table 1. Effect of timestamp (*input length*) on prediction results

Timestamp	RMSE	MAPE	R^2
90	0.843	25.041	0.157
120	**0.629**	**22.212**	**0.371**
150	0.917	24.021	0.087
180	0.929	28.089	0.025

The data must also be considered to evaluate the impact of smoothing and denoising; the second experiment is designed to assess the performance of the proposed approach with different smoothing methods. In the second experiment, two filters, the S-G and SMA filters, were used with different window sizes: S-G filters (Rooij, 2022 is verified with the window of w data points corresponding to $90, 120, 150, 180$ and SMA with the value being $5, 10, 20, 30$.

The proposed model provides long-term forecast days to minimize the accumulation of errors. The result in Table 2 is the forecasting results over 60 days and is calculated as an average of over 14 trains of RO systems. In this experiment, the performance of the proposed model is also compared to another model, including LSTM, CNN & LSTM, and GRU models.

Table 2 shows that data smoothing methods significantly influence the forecasting performance of models. Choosing a smoothing window size w that is too short may not effectively address the issue while selecting one that is too long can potentially lead to losing important information in the data. It can be seen that with a smoothing window size of SMA equals 10, an increase of 6 % compared to the window size 5. This indicates that the data is smoothed at an acceptable level, preserving essential information and reducing the impact of outliers. Besides, when applied experimentally, the S-G filter yielded poor results due to over-smoothing of the data. Compared to the three existing approaches (LSTM, CNN+LSTM, GRU), it is clearly shown that the proposed TCN-based prediction approach provides better results.

In addition, we conducted a third experiment group to see the smoothing technique's effect on each train of the RO system more deeply. Table 3 presents the obtained results. It can be seen that the SMA filter provides better outcomes. In Table 3, the values across the RO trains are generally promising; however, there are still some instances where results are less favorable, such as Train 9 with $RMSE = 0.669$, $MAPE = 18.331$, and $R^2 = 0.331$. To evaluate the multi-step prediction process significantly affected by the accumulation of errors, we conducted an experiment to test this. The model outcomes are depicted in Figure 5. From the obtained results in Figure 5, we conducted forecasts of all points in the RO train with train data (the dataset used for training the model, which accounts for about 80% of the RO train, and 10% each for validation and test) and the test set (the remaining part). We attempted to make predictions on the entire RO train based on the initial 120 timestamps. However, we recommend making short-term forecasts that are approximately equal to the output range of the model.

Additionally, based on Table 2, the negative R^2 (GRUs model) value indicates that your regression model is making worse predictions compared to simply using the mean of the dependent variable (i.e., $\hat{y}$, $\bar{Y}$) for predictions. This suggests that the model may need improvement, or there might be issues with data selection or preparation. Experiments with such models often result in a straight or very poor line. R^2 will be negative if the model's predictions are worse than using the mean value of the target variable for all predictions. This occurs when the sum of squared differences between the actual values and the predicted values is greater than the sum of squared differences between the actual values and the mean value:

$$R^2 < 0 \text{ if } \sum (y_i - \hat{y}_i)^2 > \sum (y_i - \bar{y})^2$$

5. CONCLUSION

In this paper, we gained a deep understanding of the impact of the algal bloom phenomena in seawater on seawater desalination membranes using RO technology, with a critical indicator of membrane fouling being increased filtration pressure. This knowledge is the base for us to continue research and propose a deep learning-based method to predict pressure dynamics at both ends of the RO vessel, a technique has yet to be widely explored in this context. In addition, we meticulously smoothed the data to clarify trends using Savitzky-Golay and Moving Average filters. The experiment results of the proposed approach over data collected from the Carlsbad desalination plant in California, including 14 RO trains from 2015 to 2020 (over 16,000 data points), presented compelling predictive results, opening up a promising for research and application in the field.

ACKNOWLEDGMENT

This document is the results of the research project funded by QG.23.71 of Vietnam National University, Hanoi

Table 2. Comparison of forecasting methods with different smoothing methods

Method	Window	Proposed Model			LSTM			CNN + LSTM			GRU		
		RMSE	MAPE	R^2	RMSE	MAPE	R^2	RMSE	MAPE	R^2	RMSE	MAPE	R^2
Moving average	5	0.130	6.551	0.891	0.857	25.078	0.157	1.011	25.023	-1.021	1.960	44.241	-17.982
	10	**0.097**	**3.091**	**0.945**	0.675	12.192	0.671	0.127	6.042	0.894	1.976	44.511	-16.276
	20	0.180	8.411	0.851	0.725	20.901	0.229	0.568	12.879	0.442	1.949	43.763	-12.991
	30	0.251	9.891	0.723	0.877	26.981	0.114	0.564	12.943	0.421	1.901	43.671	-11.143
Savitzky-Golay filter	90	0.138	8.212	0.857	1.037	29.891	-0.112	0.431	12.591	0.572	2.571	50.112	-43.951
	120	0.177	8.156	0.871	0.769	16.901	0.241	0.376	12.912	0.670	2.522	48.687	-40.112
	150	0.098	3.521	0.919	0.704	14.768	0.299	0.138	6.621	0.871	2.438	46.112	-31.112
	180	0.101	4.023	0.904	0.767	16.872	0.243	0.893	22.119	0.084	2.429	46.991	-32.651

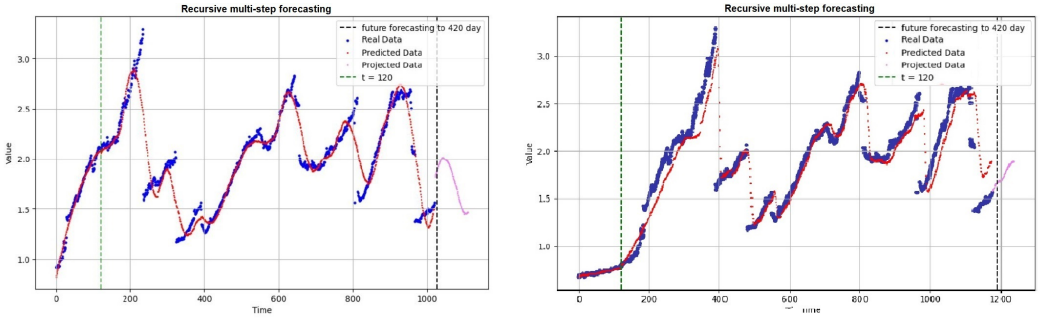


Figure 5. The proposed model results when applying recursive forecasting on both training and testing data, using S-G filter (left) and SMA (right). The proposed model takes several days to forecast for the next few days.

Table 3. Performance of the proposed approach on each train (above 1300 days in all RO trains) using S-G filter ($w = 150$, degree poly $= 4$) and Moving Average ($w = 10$)

	S-G filter			Moving average		
Train number	RMSE	MAPE	R^2	RMSE	MAPE	R^2
1	0.106	6.648	0.872	0.099	4.924	0.924
2	0.098	4.803	0.930	0.090	2.713	0.979
3	0.062	2.617	0.984	0.050	1.108	0.996
4	0.269	9.886	0.731	0.119	6.222	0.850
5	0.078	6.498	0.967	0.101	3.493	0.920
6	0.128	6.767	0.863	0.051	1.408	0.992
7	0.122	6.123	0.882	0.080	3.023	0.949
8	0.122	6.172	0.881	0.178	8.895	0.846
9	0.288	9.963	0.712	0.669	18.331	0.331
10	0.119	5.999	0.888	0.055	1.519	0.991
11	0.061	2.049	0.989	0.074	2.390	0.971
12	0.379	12.628	0.674	0.389	12.428	0.678
13	0.127	6.733	0.868	0.079	6.525	0.964
14	0.321	11.803	0.682	0.185	6.913	0.838
Average	**0.162**	**7.049**	**0.851**	**0.158**	**5.706**	**0.873**

REFERENCES

Alex, Graves and Graves Alex (2012). "Long short-term memory". In: *Supervised sequence labelling with recurrent neural networks*, pp. 37–45.

Asif, Matin et al. (2021). "Fouling control in reverse osmosis for water desalination & reuse: Current practices & emerging environment-friendly technologies". In: *Science of the total Environment* 765, p. 142721.

B, Gallagher Neal (2020). "Savitzky-Golay smoothing and differentiation filter". In: *Eigenvector Research Incorporated*. URL: https://eigenvector.com/wp

-content/uploads/2020/01/SavitzkyGolay.pdf.

He, Kaiming et al. (2015). *Deep Residual Learning for Image Recognition*. arXiv: 1512.03385 [cs.CV].

Jiang, Shanxue, Yuening Li, and Bradley P. Ladewig (Oct. 2017). "Biofouling in capillary and spiral wound membranes facilitated by marine algal bloom". In: *Desalination* 424, pp. 74–84.

Kiranyaz, Serkan et al. (2019). *1D Convolutional Neural Networks and Applications: A Survey*. arXiv: 1905.03554 [eess.SP].

Koutsakos, Erineos and David Moxey (2007). "Membrane Management System". In: *Desalination* 203.1. EuroMed 2006, pp. 307–311. ISSN: 0011-9164.

Lea, Colin et al. (2016). *Temporal Convolutional Networks for Action Segmentation and Detection*. arXiv: 1611.05267 [cs.CV].

Nour, AlSawaftah et al. (2022). "A Review on Membrane Biofouling: Prediction, Characterization, and Mitigation". In: *Membranes* 12.12.

Rooij, F van (2022). "Managing the restoration of membranes in reverse osmosis desalination using a digital twin". Depositing User : van Rooij, FREDERICUS IGNATIUS MARIA Comments and Notes (inc. copyright restrictions) : Made live with one month restriction - SD, 12/12/2022 Department : Salford Business School. PhD thesis.

Shaojie, Bai, Kolter J Zico, and Koltun Vladlen (2018). "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271*.

Sharma, Angira et al. (2022). "Digital Twins: State of the art theory and practice, challenges, and open research questions". In: *Journal of Industrial Information Integration* 30, p. 100383. ISSN: 2452-414X.

Sherstinsky, Alex (2018). "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network". In: *CoRR* abs/1808.03314. arXiv: 1808.03314. URL: http://arxiv.org/abs/1808.03314.

van Rooij, Frits, Philip Scarf, and Phuc Do (2021). "Planning the restoration of membranes in RO desalination using a digital twin". In: *Desalination* 519, p. 115214. ISSN: 0011-9164.

Villacorte, L.O. et al. (2017). "Biofouling in capillary and spiral wound membranes facilitated by marine algal bloom". In: *Desalination* 424, pp. 74–84. ISSN: 0011-9164.

Yong, Liu et al. (2022). "Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., pp. 9881–9893.