Using Deep Learning models to Predict the Electrical Conductivity of the influent in a Wastewater Treatment Plant

João Pereira[1][0009−0001−0819−6955], Pedro Oliveira[1][0000−0001−7143−5413], M. Salomé Duarte[2,3][0000−0003−4645−908X], Gilberto Martins[2,3][0000−0001−7187−0538], and Paulo Novais[1][0000−0002−3549−0754]

[1] LASI/ALGORITMI Centre, University of Minho, Braga, Portugal
joaoprp111@gmail.com,pedro.jose.oliveira@algoritmi.uminho.pt,pjon@di.uminho.pt
[2] CEB – Centre of Biological Engineering, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
salomeduarte@ceb.uminho.pt,gilberto.martins@deb.uminho.pt
[3] LABBELS – Associate Laboratory, Braga/Guimarães, Portugal

Abstract. Nowadays, human population face increasing water pollution problems, so treating and managing this resource is crucial. Wastewater Treatment Plants (WWTPs) provide essential services for human life since they treat wastewater and monitor its parameters to preserve water quality standards. One of these parameters is electrical conductivity, essential in quantifying water salinity levels. Therefore, this paper aims to forecast the influent conductivity in a WWTP for the next two timesteps. Hence, several experiments were conducted, considering the use of Transformers and Long Short-Term Memory (LSTMs) candidate models that were developed, tuned, and evaluated, utilizing a recursive multi-step forecasting approach. The best candidate model was based on a Transformer architecture with encoding and obtained a RMSE of 155.2 µS/cm.

Keywords: Deep Learning · Influent Conductivity · Time Series · Wastewater Treatment Plants.

1 Introduction

During the last decades, human activities resulted in the pollution of an exceedingly significant natural resource, specifically water. Since water is essential to human life, it is crucial to combat its pollution [1]. Thus, Wastewater Treatment Plants (WWTPs) are essential for reducing water courses contamination and monitoring its quality. WWTPs can identify and develop the necessary treatments to transform the influent wastewater into a higher quality water [2]. Therefore, one of the essential tasks in a WWTP is monitoring influent wastewater indicators to control its concentrations since the wastewater is later discharged to the environment [3]. Additionally, this task also helps to manage the WWTP resources in order to achieve the best wastewater treatment possible.

One of the parameters found in wastewaters, which can play an essential role in WWTPs, is electrical conductivity. It is relevant in a WWTP because it can provide insights about the chemical processes of these infrastructures [4]. Conductivity describes the salinity level of the water, which can help detect seawater intrusions, measure the concentration of ionized chemicals in the water, and identify possible illegal water discharges sources, through seasonality [5]. Herewith, conductivity can present higher or lower values than usual, which can help detect water pollution and environmental changes. These variations in the values can be influenced by some factors, such as temperature, inorganic dissolved solids, the geology of the area where the water flows, and sea tides [6].

Therefore, the present article consists of developing, optimising and evaluating some Deep Learning (DL) candidate models to forecast the influent conductivity in a WWTP for the next two days. Thus, the selected models for this task were based on Long Short-Term Memory (LSTMs) and Transformers, utilising a multivariate recursive multi-step forecasting approach. The former model can be advantageous in this work because of its capacity to learn sequences and long-term dependencies [7], while the latter can capture non-sequential dependencies and process the elements of the sequence in parallel [8]. Many experiments were considered to obtain the best hyperparameters set for each base model. The succeeding sections of this document are organised as follows: the next section describes the literature review, considering the influent conductivity forecasting in WWTPs. The third section describes all the steps related to data collection, manipulation, conceived DL models, and evaluation metrics. The various experiments and the discussion of the results are presented in the fourth and fifth sections, respectively. Finally, the last section presents the conclusions and future work.

2 State of the Art

The prediction of the conductivity parameter in a WWTP is a topic barely explored. However, some studies present interesting approaches for conductivity prediction in a WWTP, Water Treatment Plant (WTP), or Water Quality Station [9–11].

In a study carried out by Maleki et al. [9], the work consisted in forecasting the influent characteristics of a WTP, using an Artificial Neural Network (ANN) based model. The dataset contains daily records (collected over 2 years) of some influent parameters, like alkalinity and electrical conductivity, of the Sanandaj WTP located in Iran. Both parameters, alongside other influent characteristics, were used as inputs, and the goal of the model was to predict each input parameter for the next day of the sequence. Some steps of data manipulation took place, namely the linear interpolation to fill the missing data and the Box-Cox method to transform the data into a normal distribution. Besides these techniques, there is no reference to other ones, like feature engineering. Following this step, the data is partitioned into 70% for training, 10% for validation and 20% for testing, and the implemented model is based on a Nonlinear Autoregressive (NAR) neural network. Regarding the model training, there is no reference for cross-validation, overfitting analysis and hyperparameter optimization. Finally, in the evaluation phase, authors utilize the coefficient of determination (R^2) as the evaluation metric, and through the obtained results, the electrical conductivity achieved a R^2 of 0.55.

Another work by Fu et al. [10] focused on predicting wastewater quality parameters using a model based on a wavelet de-noised Adaptive Neuro-fuzzy Inference System (ANFIS). This study utilizes datasets that contain registers of influent parameters, such as Total Dissolved Solids (TDS) and conductivity, measured every 2 weeks in Las Vegas, Wash since 2007. The authors use this information to implement an Improved Wavelet-ANFIS (IWT-ANFIS) model capable of predicting TDS and conductivity in a multivariate forecasting approach. With this in mind, the input data is carefully selected, considering the Pearson Correlation coefficients between some chemical parameters, such as chloride and fluoride, in addition to TDS and conductivity. Considering the manipulation of the data and model training, the authors did not present too much information about it. It is possible to know that the data was divided into train and test, but there is no indication of the percentages. In addition, techniques like cross-validation, feature engineering and hyperparameter optimization are not mentioned. To complete the study, the final model is implemented and compared to five types of models, namely Multiple Linear Regressor (MLR), Multilayer Perceptron with ANN (MLP-ANN), ANFIS, ANN-ANFIS, and WT-ANFIS. Mean Absolute Percentage Error (MAPE) and R^2 are the evaluation metrics used. Furthermore, the final results of the testing phase, in terms of MAPE and R^2, show that the conductivity forecasting reaches the scores of 0.577 and 0.985, respectively.

Najah et al. [11] proposed a study to forecast water quality parameters utilising a Wavelet (WT) ANFIS-based model with hold-out validation. In order to carry out this work, they used monthly measurements of water quality indicators, such as temperature, conductivity, turbidity and TDS, between 1998 and 2007. These registers correspond to the Johor River Water Quality Station region in Malaysia. Furthermore, the main goal was to forecast conductivity, turbidity and TDS using a multivariate approach, with parameters such as pH and temperature. The authors implemented different architectures of the WT-ANFIS model for each predicted parameter. Using the hold-out validation technique and overfitting analysis, the implemented model outperformed the traditional ANFIS. Although the authors never mention the units of measurement for conductivity, the results show that this parameter reached a Mean Absolute Error (MAE) of 30.6 in the testing set. Regarding the data split step, 20% of the data was utilised for testing, 8% for validating and 72% for training. There is no reference to their implementation in this work regarding other essential techniques like hyperparameter optimisation and feature engineering.

Considering the mentioned studies and particularly the cross-validation techniques, one specific technique, namely *TimeSeriesSplit* (suited for time series problems), was not utilized. Besides, other relevant techniques utilized in Machine Learning (ML) problems, such as feature engineering and hyperparameter optimization, were not mentioned. Additionally, data preparation tasks and underfitting/overfitting analysis are topics that were not considered in all the studies. When facing a time series problem, most of these techniques are essential for conceiving the models. Finally, the DL models, such as LSTMs, not considered in any of the studies, can be an advantage in this work because of their capacity to handle temporal dependencies.

3 Materials and Methods

This section provides information about the materials used in this study, namely data collection, exploration and manipulation, the devised DL models, and the evaluation metrics. In addition, elucidation is provided regarding the details of the datasets.

3.1 Data Collection

Concerning the data collection step, the dataset was made available by a portuguese multimunicipal company treating urban wastewaters. In addition, this data represents real events that occurred in a WWTP situated near the coast, and it contains data between January 2^{nd}, 2019 and October 31^{th}, 2022, registered every 2 days. Another dataset was utilized regarding the weather data in the exact location of the WWTP. This one was provided by *OpenWeatherMap*, in the same date range as the first dataset but with a different periodicity. The dataset contained hourly records between January 2^{nd}, 2019 and October 31^{th}, 2022.

3.2 Data Exploration

Given the data points, in the first dataset analysis, it is possible to verify that the WWTP parameters dataset has a total of 700 observations, while the weather dataset contains 33576. Table 1 presents the features of the WWTP parameters, and Table 2 describes some of the weather features. A short description, the data type and the units of measure are also presented. These tables show that most features contain *doubles*, except the features that represent the measurement date.

Table 1. Features of the WWTP parameters dataset.

#	Feature	Description	Data type	Unit of measure
1	Conductivity	Influent conductivity value	*double*	µS/cm
2	Flow Rate	Influent flow rate value	*double*	m^3/day
3	Date	Date of measurement	*timestamp*	Date

In the WWTP parameters dataset, each observation contains two features representing wastewater indicators: electrical conductivity and water flow rate.

Another feature is the time label, which represents the observation timestamp. Conversely, the weather dataset contains thirteen features representing weather characteristics, including air temperature, atmospheric pressure, wind speed and clouds. As the first dataset mentioned, each observation is associated with the respective timestamp, represented by the *date_time* feature.

Table 2. Some features of the weather dataset.

#	Feature	Description	Data type	Unit of measure
1	temp	Temperature	*double*	*Celsius*
2	pressure	Atmospheric pressure	*double*	hPa
3	humidity	Humidity	*double*	%
4	wind_speed	Wind speed	*double*	m/s
5	wind_deg	Wind direction	*double*	Weather degrees
6	date_time	Date of measurement	*timestamp*	Date

Considering the features distribution, we performed a statistical analysis and the Kolmogorov-Smirnov test to find if the data distribution is related to a Gaussian distribution. The final conclusion was that none of the WWTP parameters and weather features present a Gaussian data distribution, because p value obtained from the test was lower than 0.05.

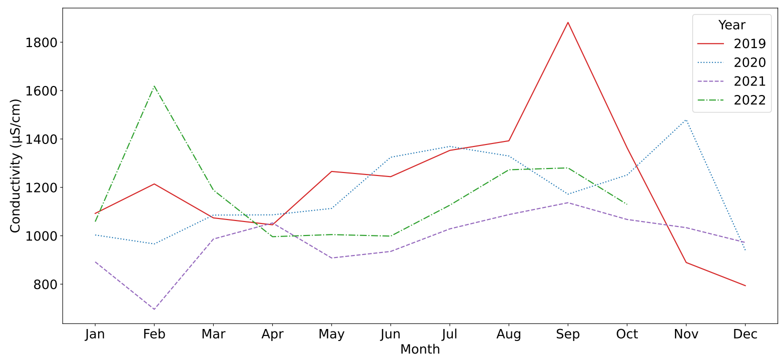
Considering the statistical details regarding conductivity, it yields a *skewness* value equal to 5.42 and 63.7 for *kurtosis*, which means the distribution is right skewed and leptokurtic, respectively. Besides, the maximum, minimum, standard deviation, and mean values are 6320, 5, 389.57, and 1136.55, respectively.

Regarding the date range of the collected data, since this is a time series problem, it is essential to analyze if there are missing timestamps or missing values. So, after the exploration, no missing timestamps are found in WWTP parameters and weather datasets. However, there are missing values in the WWTP parameters dataset. Therefore, these values needed to be filled, and this treatment is explained in detail in the data preparation section.

Finally, a plot was developed to verify the monthly average conductivity values for each year to complete the data exploration. Afterwards, this exploration task consisted of finding patterns between the different years to search for seasonality or cyclic behaviours in the data.

As shown in Figure 1, it is possible to get an overall look at the monthly average variations of the values. There is a similarity in the values between May and December because there is a predominant increase between May and September, followed by a fall between September and December every year. Thus, this could indicate a small amount of seasonality at this time. Furthermore, it is possible to note that the graph presents the highest average conductivity values in September 2019 and the lowest in February 2021. Hence, this could be an insight into high and low values tendencies.

Fig. 1. Monthly average of the conductivity values for each year.



3.3 Data Preparation

Another essential step to having a clean source of knowledge for the learning process of the candidate models is data preparation. The first task consisted of changing the weather dataset date range to allow its joining with the WWTP parameters dataset. Since this data was registered hourly, the average of the 24 daily values was calculated to get the daily values.

Subsequently, the goal was to identify possible insertion errors in both datasets. To complete this task, the data was carefully explored, where the values presented significant deviations from the majority, i.e., near the maximum and minimum data points. Conductivity and flow rate columns appeared to have insertion errors in some cases, that is to say, some digits could be missing. However, there were no apparent insertion errors in each weather column. So, regarding conductivity and flow rate columns, it was decided that the possible insertion errors would be treated later as outliers.

As mentioned above, some missing values exist in the WWTP parameters dataset. There is a total of 13 for conductivity and 2 for flow rate. With this in mind, the linear interpolation technique filled the missing data of the WWTP parameters features. Afterwards, the data treatment process replaced the data points that could be outliers. Hence, to find those data points, it was necessary to analyze in detail the data distribution. Then, maximum and minimum thresholds were selected for each WWTP parameter to define outliers' limits. Finally, the values greater or lower than the maximum and minimum thresholds were replaced by the last four registers mean.

The following step consisted in joining both datasets utilising the inner join approach. In this way, datasets were concatenated by the timestamp, and consequently, some weather dataset timestamps were discarded. Subsequently, new features related to the timestamp, namely the day, month, week, year, season, semester and trimester, were added to the dataset. Finally, to select the best features of the dataset, we performed a correlation analysis between all of the features. As mentioned in the previous section, none of the features presented Gaussian data distribution, so we utilized Spearman's correlation coefficient test.

Regarding the weather data, most of the features were discarded after the correlation analysis, such as *clouds*, *visibility* and *feels_like*. Thus, the remaining weather features slightly correlated to conductivity were described in Table 2. Afterwards, the features with moderate correlation coefficients concerning conductivity (target) were season and temperature. Hence, all the other features were removed from the dataset. Finally, the data were normalized between the values of -1 and 1 in the LSTM-based models and between 0 and 1 in the case of the Transformer-based models.

3.4 Transformers

Transformers are a type of DL model architecture introduced in 2017 [12]. The vanilla architecture consists of an encoder and a decoder, both composed of multiple layers of self-attention and feed-forward neural networks. Self-attention allows the model to capture dependencies between different positions in a sequence, while the feed-forward networks help process and transform the representations. The vanilla Transformer has been widely used in Natural Language Processing (NLP) tasks and has achieved state-of-the-art performance in various domains, including machine translation and language generation [13].

This algorithm has also shown effectiveness in time series forecasting tasks, for example, in energy consumption prediction [14], since it can capture non-sequential long-range dependencies and process sequences in parallel. These characteristics can help achieve better results when compared to Recurrent Neural Networks (RNNs) [15]. Although, its performance can vary depending on the specific task and the dataset's quality, among other factors.

3.5 LSTMs

LSTMs are a variant of RNNs frequently applied due to their ability to capture order dependencies in sequential data problems. Additionally, they solve the RNNs vanishing gradient problem handling long-term dependencies [16]. In their architecture, a group of cells contains properties that allow them to remember or forget information over time. These cells, or units, have three gates, namely input, output, and forget gates.

Considering these three gates, the first controls how much information is added to the cell. The output gate controls the information that continues to the subsequent layers. Finally, forget gate manages the information that is discarded by the network. With these characteristics, LSTM is prepared to manage and process information over long data sequences [17]. It can remember information over a long period because a model based on an LSTM can capture essential features from past inputs and preserve the information taken from it [18].

3.6 Evaluation Metrics

Since the problem faced in this work is a regression one, two evaluation metrics were considered. These metrics help to evaluate the candidate models developed. Hence, the metrics considered for this task are MAE and Root Mean Squared Error (RMSE). MAE measures the average magnitude of the errors between predicted and real values. Furthermore, calculating the absolute differences between them makes it possible to know if the models' predictions match the true values. MAE is less affected by outliers compared to other metrics like RMSE. The following formula shows how MAE is calculated:

$$MAE = \frac{\sum_{i=1}^{n} |yi - yi_{pred}|}{n} \tag{1}$$

The other metric, RMSE, is similar to MAE, but instead, it calculates the square root of the average of the squared differences between the predicted and true values. It considers both the magnitude and direction errors. Thus, it is more sensitive to outliers. The formula is the following:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (yi - yi_{pred})^2}{n}} \tag{2}$$

In equations 1 and 2, yi, yi_{pred} and n represent the actual values, predicted values, and the number of data points, respectively.

4 Experiments

Several experiments were developed in order to predict influent conductivity. To obtain the best hyperparameters combination, transformer-based and LSTM-based candidate models were included in a tuning process, following a multivariate recursive multi-step forecasting approach. During this process, season and temperature data fed the models to predict the next two conductivity values. Regarding this tunning process, Table 3 summarizes the hyperparameters ranges of each model.

Table 3. Hyperparameters' searching space for each model.

Parameters	Transformer value range	LSTM value range
Activation Function	[ReLU,tanh]	[ReLU,tanh]
Batch Size	[10,20]	[10,20]
Dropout	[0.0,0.5]	[0.0,0.5]
Encoding Layers	[4,8]	-
Epochs	[50]	[25]
Layers	-	[3,4,5]
MLP Layers	[3,4]	-
Neurons	[64,128]	[32,64,128]
Number of heads	[4,8]	-
Timesteps	[4,6,8]	[4,6,8]

All candidate models are evaluated in terms of performance by the two metrics mentioned in the previous section. Different values for each hyperparameter were considered for both base models. One important hyperparameter in this context is the number of timesteps. They are given as input to the candidate models, and the values 4, 6 and 8 were considered in both models. Considering 4 timesteps as an example, a sequence of 4 records (recorded every 2 days) is utilized to predict the following two conductivity registers (with a periodicity of 2 days). Additionally, some of the other hyperparameters are specific to the model, like *head size* and *ff_dim*, which are used only in Transformer, with the values of 128 and 512, respectively. During the experiment runs, all candidate models had the same seed value, in this case, 91195003.

The conceived tests were supported by plotting the learning curves to prevent overfitting and underfitting situations. Afterwards, these curves were analyzed to choose the number of epochs. In addition, it was crucial to use the TimeSeriesSplit cross-validator with a k value equal to 3, since we are dealing with a time series problem.

Concerning the technologies used, in terms of programming language, Python, version 3.10, was used for conceiving, tuning and evaluating the different candidate models. Also, libraries like Numpy, Pandas, Matplotlib, and scikit-learn, among others, were considered. Regarding the development of ML models, TensorFlow v2.0.0 was used. Finally, all hardware used in the development of this study was provided by Google's Collaboratory.

5 Results and Discussion

After all the experiments were performed, it was necessary to analyze their results. Table 4 summarizes the results according to the model's nature, namely Transformer and LSTM candidate models. In this table, it is possible to verify the hyperparameters space, utilized in each of the top-five candidate models and their scores in terms of RMSE, MAE, and training time.

Table 4. Top-five candidate models results, for Transformers and LSTMs.

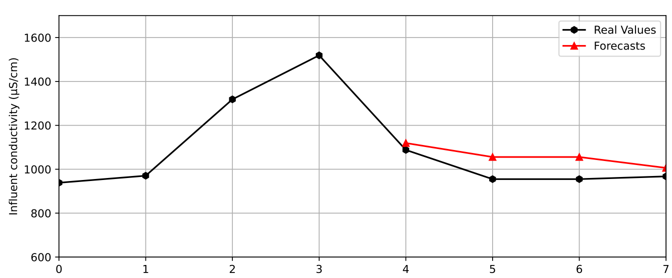
Timesteps	Batch	Layers	Neurons	Dropout	Activation	Epochs	Encoding layers	Heads	RMSE	MAE	Time(s)
Transformer candidate models											
4	10	4	64	0.0	tanh	50	8	8	155.2	135.4	355.6
6	10	3	128	0.5	tanh	50	4	8	156.0	141.6	199.0
4	20	3	128	0.5	tanh	50	4	4	156.6	146.0	119.7
8	20	3	64	0.0	tanh	50	4	4	157.7	141.7	133.5
6	10	4	128	0.5	tanh	50	4	8	159.2	149.6	203.2
LSTM candidate models											
4	10	4	128	0.0	tanh	25	-	-	179.9	162.4	69.4
4	10	4	128	0.5	tanh	25	-	-	187.1	170.4	72.1
4	10	5	128	0.0	tanh	25	-	-	187.9	170.3	82.0
4	20	4	128	0.0	relu	25	-	-	188.2	169.9	57.8
4	20	3	128	0.5	relu	25	-	-	190.0	172.6	50.6

Regarding the Transformer-based models, some hyperparameters utilized are specific to Transformers, like the number of encoding layers and the number of heads, since Transformer uses an encoding layer, and LSTM does not. Considering these candidate models, the best one achieved a RMSE of 155.2 and a MAE of 135.4. In addition, the activation function considered was the *tanh*, the dropout rate was set to 0.0, and the number of layers and neurons was 4 and 64, respectively. Considering the number of timesteps utilized as input, it was 4. Concerning the top-five Transformer-based models, there are some predominant hyperparameters such as the layers (3), encoding layers (4), and activation function (*tanh*).

When considering the LSTM-based models, the best one achieved a RMSE of 179.9 and a MAE of 162.4. This candidate model needed 4 timesteps as input to achieve the best performance and utilized 0.0 as the dropout value. Additionally, *tanh* was utilized as the activation function, and the number of neurons was 128. Finally, the batch size was 10, and the number of layers of this model was 4. Considering the top-five LSTM-based models, there are homogeneous hyperparameters, namely the timesteps (4) and the number of neurons (128). Apart from this, the other parameters are not homogeneous but show some predominance in the values, like the dropout (0.0) in the majority of the candidate models and the activation function (*tanh*).

Finally, considering both LSTM-based and Transformer-based models, it is possible to observe that Transformer performed better in this time series task when considering top-five candidate models. However, these models required longer times to train (more than 100 seconds), and the number of epochs was superior to the ones utilized in the LSTM-based models (50 instead of 25). Finally, in the fourth Transformer-based model of the top five, the required number of input timesteps was more significant (8). Figure 2 illustrates four predictions of the influent conductivity of the best candidate model, the Transformer-based one. As can be observed in the figure, the number of timesteps used as input was 4 to forecast the next four timesteps recursively.

Fig. 2. Four multi-step forecasts of influent conductivity.



6 Conclusions

Monitoring the water conductivity in a WWTP can be extremely helpful during wastewater treatment and management tasks since it can give important information about the water's salinity and help detect problems in water quality. In fact, it can be possible to detect saltwater intrusions, and consequently, adjust the wastewater treatment, since this phenomen might dilute the influent and present higher flow rates. Influent conductivity, in particular, provides earlier insights about the overall water quality entering the WWTP. Besides, WWTP operators can make adjustments to make the treatment process more efficient in the following stages of the WWTP. With this in mind, this work consisted of forecasting the next two timesteps (in a periodicity of two days) regarding the influent conductivity using LSTM and Transformer-based models.

Therefore, several experiments were considered to achieve the best hyperparameter set and performance for LSTM and Transformer architectures. Additionally, two features of the weather dataset were considered in the conception and training of the models, namely the temperature and season of the year, which presented a correlation with conductivity. The best model was based on a Transformer with encoding, achieving an RMSE of 155.2 and an MAE of 135.4.

In the future, this work could be improved by utilizing more models for comparison with the models utilized in this study, such as CNNs and ANFIS. More specifically, hybrid models that might achieve better performances in influent conductivity forecasting, such as CNN-LSTM and GRU-LSTM. Additionaly, new features, such as TDS, could be used to find if the influent conductivity forecasting performance can improve.

Acknowledgements. This work is financed by National Funds through the Portuguese funding agency, FCT - Fundação para a Ciência e a Tecnologia within project 2022.06822.PTDC. The work of Pedro Oliveira was supported by the doctoral Grant PRT/BD/154311/2022 financed by the Portuguese Foundation for Science and Technology (FCT), and with funds from European Union, under MIT Portugal Program.

References

1. Chaudhry, F. N., & Malik, M. F. (2017). Factors affecting water pollution: a review. J. Ecosyst. Ecography, 7(1), 225-231. doi: 10.4172/2157-7625.1000225
2. Salgot, M., & Folch, M. (2018). Wastewater treatment and water reuse. Current Opinion in Environmental Science & Health, 2, 64-74. doi: 10.1016/J.COESH.2018.03.005.
3. Ahmed, U., Mumtaz, R., Anwar, H., Mumtaz, S., & Qamar, A. M. (2020). Water quality monitoring: from conventional to emerging technologies. Water Supply, 20(1), 28-45. doi: 10.2166/ws.2019.144
4. Banna, M. H., et al. (2014). Miniaturized water quality monitoring pH and conductivity sensors. Sensors and Actuators B: Chemical, 193, 434-441. doi: 10.1016/j.snb.2013.12.002

5. Rusydi, A. F. (2018, February). Correlation between conductivity and total dissolved solid in various type of water: A review. In IOP conference series: earth and environmental science (Vol. 118, p. 012019). IOP Publishing. doi: 10.1088/1755-1315/118/1/012019
6. Ramos, P. M., Pereira, J. D., Ramos, H. M. G., & Ribeiro, A. L. (2008). A four-terminal water-quality-monitoring conductivity sensor. IEEE Transactions on instrumentation and measurement, 57(3), 577-583. doi: 10.1109/TIM.2007.911703
7. Elsworth, S., & Güttel, S. (2020). Time series forecasting using LSTM networks: A symbolic approach. arXiv preprint arXiv:2003.05672. doi: 10.48550/arXiv.2003.05672
8. Yang, Y., & Lu, J. (2022). A Fusion Transformer for Multivariable Time Series Forecasting: The Mooney Viscosity Prediction Case. Entropy, 24(4), 528. doi: 10.3390/e24040528
9. Solaimany-Aminabad, M., Maleki, A., & Hadi, M. (2013). Application of artificial neural network (ANN) for the prediction of water treatment plant influent characteristics. Journal of advances in environmental health research, 1(2), 89-100. doi: 10.22102/jaehr.2013.40130
10. Fu, Z., Cheng, J., Yang, M., & Batista, J. (2018, January). Prediction of industrial wastewater quality parameters based on wavelet de-noised ANFIS model. In 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 301-306). IEEE. doi: 10.1109/CCWC.2018.8301761
11. Najah, A. A., El-Shafie, A., Karim, O. A., & Jaafar, O. (2012). Water quality prediction model utilizing integrated wavelet-ANFIS model with cross-validation. Neural Computing and Applications, 21, 833-841. doi: 10.1007/s00521-010-0486-1
12. Vaswani, A., et al. (2017). Attention is all you need. Advances in neural information processing systems, 30. doi: 10.48550/arXiv.1706.03762
13. Zhao, Y., Zhang, J., & Zong, C. (2023). Transformer: A General Framework from Machine Translation to Others. Machine Intelligence Research, 1-25. doi: 10.1007/s11633-022-1393-5
14. Saoud, L. S., Al-Marzouqi, H., & Hussein, R. (2022). Household energy consumption prediction using the stationary wavelet transform and transformers. IEEE Access, 10, 5171-5183. doi: 10.1109/ACCESS.2022.3140818
15. Zhou, H., et al. (2021, May). Informer: Beyond efficient transformer for long sequence time-series forecasting. In Proceedings of the AAAI conference on artificial intelligence (Vol. 35, No. 12, pp. 11106-11115). doi: 10.1609/aaai.v35i12.17325
16. Rehmer, A., & Kroll, A. (2020). On the vanishing and exploding gradient problem in Gated Recurrent Units. IFAC-PapersOnLine, 53(2), 1243-1248. doi: 10.1016/j.ifacol.2020.12.1342
17. Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018, December). A comparison of ARIMA and LSTM in forecasting time series. In 2018 17th IEEE international conference on machine learning and applications (ICMLA) (pp. 1394-1401). IEEE. doi: 10.1109/ICMLA.2018.00227
18. Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In 2019 IEEE International Conference on Big Data (Big Data) (pp. 3285-3292). IEEE. doi: 10.1109/BigData47090.2019.9005997