SENSING WATER POLLUTION IN THE KAFUE RIVER USING CLOUD COMPUTING AND MACHINE LEARNING

[1] Mumbi Mumbi
University of Zambia
Department of Computer Science
Lusaka, Zambia
mumbi.mumbi@cs.unza.zm

[2] Jackson Phiri
University of Zambia
Department of Physics
Lusaka, Zambia
jackson.phiri@cs.unza.zm

Abstract

Clean water and sanitation are the sixth goal under the UN Sustainable Development Goals. However, reports have shown that over 129 countries are not on track to reach this goal by 2030. Besides the lack of basin management, countries are behind on monitoring of the water bodies. Water pollution affects the livelihood of people living in the catchment area of the river especially the people living in the rural areas and the animals that are dependent on that water. People who live in rural areas do not have the privilege of a piped water network that has treated water. Currently, in Zambia, water is monitored once every quarter and so, this leaves the water unmonitored for most of the time. This research proposed the development of a model based on IoT, Cloud Computing and AI for data collection and monitoring, and developed a prototype based on this model which uses machine learning to predict the quality of water. A water monitoring device was built using sensors, an Arduino and a Raspberry pi. The sensors used measured pH, temperature, electrical conductivity, total dissolved solids and turbidity. An Artificial Neural Network with one hidden layer was used to predict the Water Quality Index. This index was based off the National Sanitation Foundation Water Quality Index (NSF-WQI). The results of the model showed that it had an R^2 score of 0.953, MAE and MSE of 0.835 and 1.280 respectively. These results support the use of an ANN in the predicting WQI

Keywords
Water monitoring, IoT, Machine learning, Pollution, ANN

I. Introduction

Water pollution is a problem that has affected the world on a large scale. Access to safe water is a basic human need for health and wellbeing. Billions of people will lack access to these basic services by 2030 [1]. One of the limitations to achieving this UN SDG is lack of considerable effort towards regularly measuring water quality parameters [2]. For at least 3 billion people, the water quality they rely upon is unknown owing to lack of monitoring [1].
The Kafue River catchment spans an area of 156034.386km^2 covering 20% of Zambia [3]. This shows that a great number of households is dependent on the river. Water pollution monitoring is crucial due to its environmental and health implications. This study addresses these limitations by using cloud computing and machine learning to enhance water quality monitoring. The goal was to create an integrated system that enables real-time monitoring and advanced analytics for comprehensive water quality assessment. By leveraging cloud infrastructure and machine learning algorithms, this approach aims to overcome the constraints of conventional methods and improve the detection and response to contamination incidents. A prototype device was proposed that records readings for pH, temperature, total dissolved solids, electrical conductivity and turbidity. This used an Arduino to connect to the sensors and Raspberry pi that was used to locally store the readings and upload them to an online database. An Artificial Neural Network was then used to predict a value for the WQI.

II. Literature Review

The section looks at work that has been done concerning water quality monitoring and the use of machine learning to determine the quality of the water. The first part of the review looks at the studies that monitored water quality including what parameters were being read and the microcontrollers used as well as how the readings were stored and displayed. The latter part looks at the different machine learning algorithms employed to determine the quality of water.
Previous work on monitoring water quality using IoT have been done. Detection of contamination for both soil and water was carried out in a study [4], they used edge computing to communicate between IoT gateway and sensor system to send health alerts. The parameters that were being looked at were temperature, pH, turbidity, chemical oxygen demand, total hardness, total dissolved solids, magnesium and chloride. Vijaykumar [5] proposed to design a low-cost monitoring system that used a Raspberry Pi as a core controller. Sensors for water quality parameters were connected to the core controller. These parameters were pH, turbidity, electrical conductivity and dissolved oxygen. An IoT module was used to connect the Raspberry pi to the internet and send data. Nikhil [6] connected sensors to a NodeMCU microcontroller which used Wi-Fi to send the data to Azure Event Hub. They used Power Bi to display the sensor values in the form of a web page. A machine learning model was hosted online which predicted the temperature of the water at a given time of the year. Kamaludin [7] proposed a wireless sensor network that used radio Frequency at 920MHz instead of a Zigbee connection. This was implemented because of its ability to surpass attenuation in vegetation areas. The water quality

parameters that were being monitored were pH, dissolved oxygen, chemical oxygen demand, biochemical oxygen demand, total suspended solids and ammoniacal nitrogen.
Five classification algorithms were compared for performance using different attributes [8].
The Naïve Bayes model achieved the highest accuracy (85.19%) when using all the parameters, while the Kstar model performed best (86.67%) when only six attributes were selected. Feature selection algorithms identified three attributes and the Bagging model achieved the highest accuracy (67.41%). This research showed that the attributes selected affect the performance of the classification model. Advanced AI algorithms were developed to predict WQI [9]. The model showed accurate WQI predictions, with the NARNET model performing slightly better than the LTSM, and the SVM achieved the highest accuracy of (97.01%) for water classification. A neural network model to predict surface water WQI based on physiochemical parameters was proposed by [10]. The model achieved high accuracy with a correlation coefficient of 0.9792, low MSE and RMSE of 0.625 indicating that the effectiveness of neural networks for WQI prediction. Water quality indices such as CCME and NSF are valuable tools for assessing and communicating surface water quality [11]. In this study, these WQIs were applied to water quality data from Polyphytos reservoir-Aliakmon River in Greece. The performance and suitability of the indices were compared, with the NSF-WQI found to be more robust and closer to the classification of the WFD-ECOFRAME approach as compared to CCME.
These studies employed different sensor systems, connectivity, methods and data analysis techniques. The results demonstrated the effectiveness of these approaches in accurately assessing and predicting water quality. Furthermore, the comparison of water quality indices highlighted the suitability of the NSF-WQI for robust classification in line with regulatory frameworks.

III. Methodology

To achieve the objective of this research, a prototype device was built. The device was used to collect the readings of the water quality parameters and storing them locally and importing it to an online database. The data in the database was then used to display real-time readings of the water quality parameters on a web application.
The monitoring device was built using an Arduino mega 2560. Four sensors that were used to monitor the water quality were connected to the Arduino. The four sensors

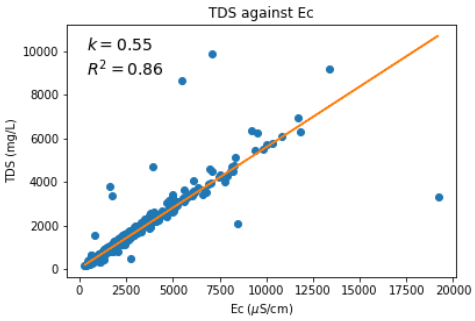

Figure 1: Graph showing the relationship between TDS and EC

were measuring pH, Temperature, Total Dissolved Solids (TDS) and Turbidity. To measure Electrical Conductivity (EC), the readings from the TDS were used to calculate EC by using the correlation between the two parameters. TDS and EC have a linear relationship when in fresh water [12]. This relationship is represented by the gradient k of the graph of TDS against EC. The equation of the line that fits the graph is given by equation (1).

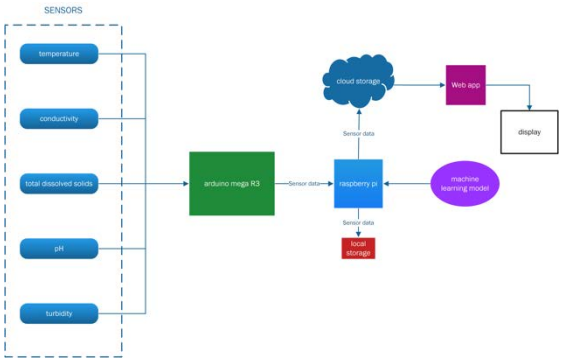

Figure 2: Design of the water monitoring system

The values of TDS and EC in the dataset obtained from Toronto and Region and Conservation Authority (TRCA) were used to plot a graph of TDS against EC shown in Figure 1. The gradient of this graph represented the correlation between the two parameters. This is what was used to determine the EC from the TDS sensor used in the water monitoring device. The readings from the TDS sensor were used to determine the EC hence it is represented in Figure 2 as one of the parameters read with the sensors.

$$TDS = k \times Ec \qquad (1)$$

Except for the temperature sensor, all of the sensors used were analogue. The analogue signals were converted to digital readings using the equations provided in each sensor's user handbook. The readings were then delivered to the Arduino's port as a single line separated by commas. On the Raspberry Pi, a Python program was built and executed that used the Serial library to read data from the USB port to which the Arduino was connected. The Asynchronous Reception and Transmission (UART) Protocol was utilised by both boards. The Python code divided the data from the Arduino using the commas in the data and stored it in a Python dictionary with the key being the name of the parameter corresponding to its value. The other key-value pairs were the current date and time, as well as an index set to an arbitrary integer. This Python dictionary was converted to JSON and stored in a MongoDB collection. There were two MongoDB collections created. The first was used to store all data acquired while the device was running, and the second was used to save data displayed on the web app. The reading in the web app collection was removed and replaced with the most recent reading. The index is what was used to delete and replace the old entry. A CSV file was also created to store the data locally in case of no internet connection.
The dataset which was used to train the model was from the Toronto Region and Conservation Authority (TRCA) open

datasets under the Toronto and Region Conservation Authority Open data V1.0 license. This data was collected over 41 monitoring stations in TRCA jurisdiction. The data was collected once every third week of the month for five years. The dataset had 69 monitored parameters. From these, only eight were used to train the model. These were pH, Temperature, Total Dissolved Solids, Nitrate, Phosphate, Turbidity, Dissolved Oxygen (% Saturation) and Biochemical Oxygen Demand. The NSF-WQI requires nine parameters, to determine the WQI, however, from the dataset available, Faecal Coliform which is one the parameters used was not present. A study by Hefni [13] addresses this particular matter. They too used eight parameters to determine the NSF-WQI. The missing parameter was also Faecal Coliform. The weights of the parameters were adjusted to accommodate this change.
To have real-time monitoring of the water parameters, a web application was used to display the results.

Table 1:Monitored water parameter description

Parameter	Description
pH	Measures how acidic or basic the water is.
Temperature	Palatability, viscosity, odour and chemical reactions are influenced by temperature
Total Dissolved Solids	How much of particles have dissolved in the water
Electrical Conductivity	The electrical conductivity of water is the ability of water to carry an electrical or conduct electricity
Turbidity	How clear the water is. It indicates the presence of pathogens, bacteria and other contaminants such as lead and mercury

The water parameters were stored in a MongoDB database. Flask, a Python framework for backend development was used to communicate with the server. To get the data from the MongoDB database, a Python driver for MongoDB called PyMongo was used. The display was updated with the latest results by using a JavaScript code which checked every second if the value in the field was the same as the one from the database or it had changed.
If it was the same, nothing happened but if it was different, the field would be updated with the new value.

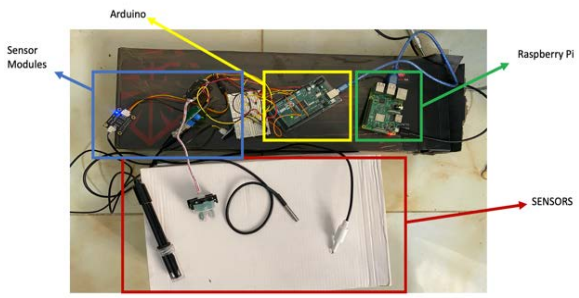

Figure 3: Image showing the components of the monitoring device

$$NSFWQI = \sum_{i=1}^{n} W_i \times Q_i \qquad (2)$$

Where: W_i is the weight score
Q_i is the sub-index

Although the original dataset had 69 parameters, only eight of them could be used to train the model based on the modified NSF-WQI, so any unnecessary parameters were dropped from the dataset. Certain metrics, such as pH and TDS, had data gathered both in the lab and in the field, and this data was utilised to fill any gaps in either column. After that, the field columns were removed from the dataset.

Table 2: Filling in the missing values

Parameter	Missing Values	
	Before	After
pH (lab)	110	36
Ph (field)	139	36
Solids, Dissolved (TDS)	143	67
Solids, Dissolved (TDS, Field)	793	67

A matrix plot was created using a Python library known as missingno to visualise the missing values in the dataset. The plot showed that a considerable number of readings was missing and any pre-processing steps of using mean or mode to fill them would have created an inaccurate dataset so these rows were dropped from the dataset.

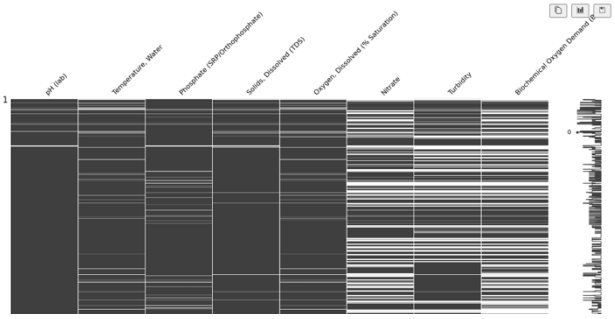

Figure 4: Matrix plot showing missing values

Anomalies and outliers were also searched for in the dataset using boxplots. The TDS column showed to have very high values far from the mean but these were not outliers, Oxygen Dissolved (% Saturation) and Turbidity had some negative values, these were few so they were also dropped from the dataset.

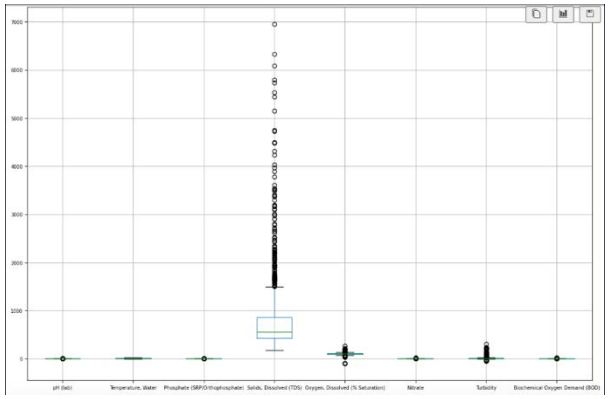

Figure 5: Boxplot for each parameter

To label the dataset, the NSF-WQI method was used. This meant that the sub-index (Q_i) for each reading had to be read from the Q-charts. In order to determine the equations of the Q-charts, the plots were replicated by entering values that ranged the entire scale for each parameter. For example, to plot the Q-Chart for Biochemical Oxygen Demand (BOD), values ranging 0 to 30 with a step size of 1 were entered into the online NSF-WQI calculator found on https://www.knowyourh2o.com, the outputs were then plotted using Google Sheets. The trendline feature was then used to fit a line to the graph plotted, the equation of that line represented the equation of the Q-Chart for that particular parameter. This process was repeated on the other parameters each with its own scale and step size. Only five equations could be found through this process, so for the rest of the parameters, the subindex had to be calculated without the use of the Q-Chart equations. The Weights used in the NSF-WQI are predetermined, but were modified since eight parameters were used instead of nine.

Table 3: Modified NSF-WQI Weight Scores

Parameter	Original Weight	Modified Weight
DO	0.17	0.20
pH	0.11	0.13
BOD	0.11	0.13
Temperature	0.10	0.12
Phosphate	0.10	0.12
Nitrate	0.10	0.12
Turbidity	0.08	0.10
TDS	0.07	0.08
Faecal Coliform	0.16	

An Artificial Neural Network (ANN) was the preferred ML model used to predict the WQI. Neural networks are more capable of modelling data that have more intricate patterns.

Table 4:Range and step size for each parameter used to plot the Q-Charts

Parameter	Range	Step size
DO	0 – 140	5
Phosphate	0 – 10	0.5
Nitrate	0 – 100	5
Turbidity	0 – 100	5
BOD	0 – 30	1

The artificial neural network (ANN) that was constructed for this study consisted of a single hidden layer. Leveraging the capabilities of the Keras library within the TensorFlow framework, the model's architecture was developed. Within the hidden layer, the chosen activation function was the widely used "Rectified Linear Unit" (ReLU), a choice motivated by its effectiveness in handling various data complexities. Correspondingly, given the nature of the model as a regression-based one, the activation function implemented in the output layer was specified as "Linear," aligning with the objective of predicting continuous values. To gauge the effectiveness of the model and guide its refinement, a two-pronged strategy was employed for defining the loss function and optimizer. The "mean squared error" was selected as the loss function, enabling the model to quantify the disparity between predicted and actual values. Complementing this, the "adam" optimizer was harnessed, enhancing the network's ability to converge towards optimal weights and biases efficiently.

Delving into the pivotal task of configuring the hidden layer, determining the appropriate number of nodes was a paramount concern. Addressing this challenge, the Keras_tuner, a specialized hyperparameter tuning tool integrated within the Keras ecosystem, was harnessed. The range of nodes was systematically set, spanning from eight (n) to twenty-one (2n+5), where n symbolizes the number of input nodes, as specified [14]. This approach allowed for an exploration of varying complexities in the hidden layer's representation.

Furthermore, in a bid to optimize the learning process of the model, the learning rate, a vital hyperparameter for the optimizer, was methodically tuned. The learning rate was trialled across discrete values, specifically 0.1, 0.01, and 0.001, unveiling the impact of different learning rates on convergence and performance.

Partitioning the dataset effectively is a prerequisite for model training and validation. In this study, the dataset was judiciously split into distinct segments: 70% for training, 15% for validation, and an additional 15% for testing. This segregation enabled rigorous training, unbiased validation, and robust testing, contributing to a comprehensive assessment of the model's generalization and predictive capabilities.

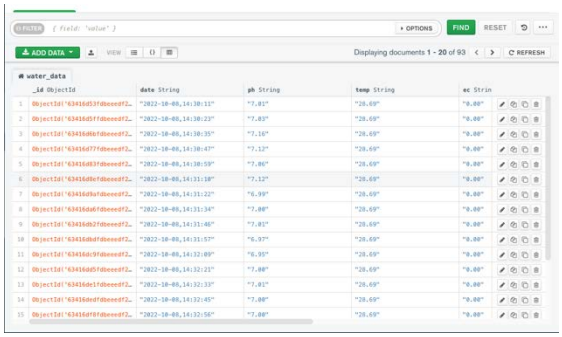

Figure 6: MongoDB collection with readings from the sensors

The interface for the model was made using the Python's Tkinter library. This can easily run Python programs like the model trained. It is cross-platform so, the same code works on Windows, macOS and Linux.

Its simplicity and ease of use makes it straightforward and intuitive to create GUI's. It allows for faster development and iteration of the user interface.

IV. Results and Discussion

This research demonstrated the successful development and implementation of a water monitoring device that effectively stores readings in an online database and displays real-time results on a web application. The device's seamless integration with the web application allows for continuous monitoring of water quality, enabling timely decision-making and proactive management.

The Q-charts plotted were identical to the known Q-charts, this supports the use of the equations generated from the trendline in Google sheets. With these equations, it means that it is not necessary to refer to the Q-charts when looking for the sub-index of a parameter. This meant that to get the sub-index for the five columns (DO, Phosphate, Nitrate, Turbidity and BOD) used to calculate the WQI used as the label, the five equations were used.

$$Q_{BOD} = 98.9 - 28.5lnx \quad (3)$$
$$Q_{DO} = 2.72 + 0.016x + 0.0243x^2 \quad (4)$$
$$Q_{phosphate} = 41 - 16.5lnx \quad (5)$$
$$Q_{nitrate} = 102 - 22.5lnx \quad (6)$$
$$Q_{turb} = 91.3 - 1.33x + 0.00739x^2 \quad (7)$$

The optimal number of nodes in the hidden layer was found to be 21, this agrees with [14]. The learning rate for the optimiser was 0.01.

Finding optimal hyperparameters overcomes challenges a trained model might experience like overfitting or underfitting. An inappropriate learning rate can lead to slow convergence during the training process.

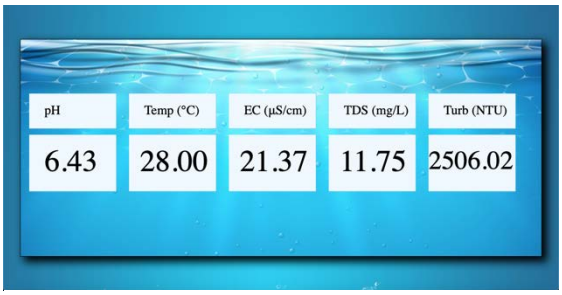

Figure 7: Web application display

Table 5:Metric scores for the ANN model

Metric	Score
R^2	0.953
MAE	0.853
MSE	1.280

The metric scores represented a good model and supported the use of ANN in predicting the NSF-WQI.

The trained ANN model was then compared with the analytical way of calculating the NSF-WQI. The results obtained were very close. This showed that a trained ANN model can be used in place of analytically calculating NSF-WQI and is even faster and can work autonomously.

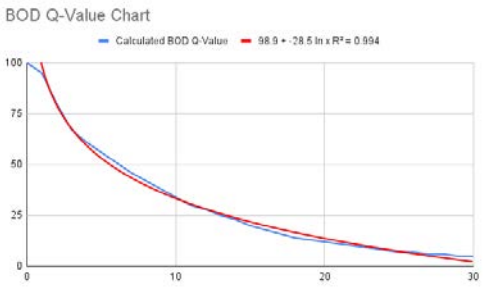

Figure 8: Replotted Q-chart for Biochemical Oxygen Demand

The analytical method requires having to check the sub-index (Q_i) for each parameter on the Q-chart. After that, each sub-index would have to be multiplied by the weight of that parameter, the summation of those products would then give the value of the NSF-WQI. This is a tiresome process and may lead to errors when multiple values are being read especially with the immense data that would be generated from the water monitoring device working around the clock. With the model, this process is entirely removed quicken the determination of the WQI.

Table 6: Analytical calculation of NSF-WQI

Parameter	Readings	Q_i	W_i	Q_i x W_i
pH	7.6	92	0.13	11.96
Temperature	14	30	0.12	3.6
Phosphate	0.014	98	0.12	11.76
TDS	502	20	0.08	1.6
DO	98	98	0.20	19.6
Nitrate	0.79	98	0.12	11.76
Turbidity	10	76	0.10	767.6
BOD	1.1	97	0.13	12.61
			WQI	80.49

The results of the analytical method and prediction demonstrated a close relationship, indicating the potential of the model to replace the analytical method. The model utilised AI to predict the water quality index based on the water parameters without the need of knowing the weights of the parameters or their sub-index value. The comparison showed high degree of accuracy and correlation between the two approaches, suggesting that the model can effectively replicate the results obtained through the analytical method. This implies that the model has the capability to serve as a reliable alternative to the labour-intensive and time-consuming analytical method.

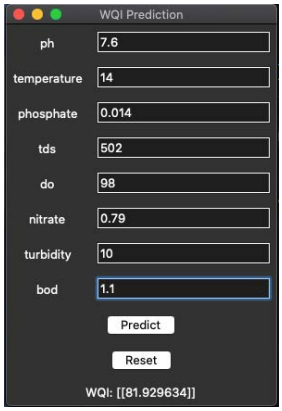

Figure 9: ANN model prediction of NSF-WQI

V. Conclusion

This research presented the successful development of a water quality monitoring device capable of storing and displaying real-time readings from select water quality parameters. Additionally, an Artificial Neural Network was trained to predict NSF-WQI. The integration of the

monitoring device with ANN provides a comprehensive solution for continuous water quality monitoring and assessment. The findings of this research highlight the potential of this approach in facilitating effective water management and decision making for ensuring safe and sustainable water resources.

VI. Recommendations

Further study should focus on increasing the number of the of sensors in the water quality monitoring system to enhance the understanding and management of water resources. This would provide a more comprehensive and detailed assessment of water quality parameters enabling better identification of pollution sources and early detection of contamination events. Since the model used in this study utilised the NSF-WQI, adding the nine sensors needed to predict this quality index would make it possible to have a real-time prediction of the NSF-WQI.

References

[1] United Nations Economic and Social Council, "Progress Toward the Sustainable Development Goals," 2022.

[2] United Nations, "SDG Metadata Indicator 6.3.2," 2022.

[3] Water Resources Management Authority, "Kafue Catchment," WARMA, [Online]. Available: https://warma.org.zm/?page_id=1549. [Accessed 12 November 2022].

[4] M. V. Ramesh, K. V. Nibi, A. Karup, R. Mohan, A. Aiswarya, A. Arsha and P. R. Sarang, "Water Quality Monitoring and Waste Management using IoT," *IEEE Global Humanitarian Technology Conference,* 2017.

[5] N. VijayKumar and R. Ramya, "The Real Time Monitoring of Water Quality in IoT Environment," *IEEE International Conference on Circuit, Power and Computing Technologies,* 2015.

[6] K. K. Nikhil and S. P. Purnendu, "Water Quality Monitoring System using IoT and Machine Learning," *IEEE International Confererence on Research in Intelligent and Computing Engineering,* 2018.

[7] K. H. Kamaludin and W. Ismail, "Water Quality Monitoring," *IEEE Conference on Systems, Process and Control,* 2017.

[8] S. Y. Muhammad, M. Makhtar, A. Rozaimee, A. A. Aziz and A. A. Jamal, "Classification Model for Water Quality using Machine Learning Techniques," *International Journal of Software Engineering and Its Application,* vol. 9, no. 6, pp. 45-52, 2016.

[9] T. H. Aldhyami, M. Al-Yaari, H. Alkhahtani and M. Maashi, "Water quality prediction using artificial intelligence algorithms," *Applied Bionics and Biomechanics,* vol. 2020, 202.

[10] M. Kulisz and J. Kujawska, "Application of artificial neural network (ANN) for water quality index (WQI) prediction for the river Warta, Poland," *Journal of Physics: Conference Series,* vol. 2130, no. 1, 2021.

[11] D. Alexakis, V. A. Tsihrintzis, G. Tsakiris and G. D. Gikas, "Suitability of water quality indices for application in lakes in the Mediterranean," *Water Resources Management,* vol. 30, pp. 1621-1633, 2016.

[12] A. F. Rusydi, "Correlation between conductivity and total dissolved solids in various type of water: A review," *IOP Conference Series: Earth and Environmental Science,* vol. 118, 2018.

[13] H. Effendi and Y. Wardiatno, "water quality status of Ciambulawung River, Banten Province, based on pollution index and NSF-WQI," *Procedia Environmental Sciences,* vol. 24, pp. 228-237, 2015.

[14] J. G. Nayak, L. Patil and V. K. Patki, "Artificial neural network based water quality index (WQI) for river Godavari (India)," *Materials Today: Proceedings,* 2021.