

The application of artificial neural network model to predicting the acid mine drainage from long-term lab scale kinetic test

Muhammad Sonny Abfertiawan^{1*}, *Muchammad Daniyal Kautsar*², *Faiz Hasan*³, *Yoseph Palinggi*⁴, and *Kris Pranoto*⁴

¹Water and Wastewater Research Group, Faculty of Civil and Environmental Engineering, Bandung Institute of Technology

²Information Engineering Undergraduate Program, Department of Electrical and Information Engineering, Universitas Gadjah Mada

³Environmental Engineering Master Program, Faculty of Civil and Environmental Engineering, Bandung Institute of Technology

⁴Environmental Department, PT Kaltim Prima Coal, Indonesia

Abstract. Acid mine drainage (AMD) is one of the common environmental problems in the coal mining industry that was formed by the oxidation of sulfide minerals in the overburden or waste rock. The prediction of acid generation through AMD is important to do in overburden management and planning the post-mining land use. One of the methods used to predict AMD is a lab-scale kinetic test to determine the rate of acid formation over time using representative samples in the field. However, this test requires a long-time procedure and large amount of chemical reagents lead to inefficient cost. On the other hand, there is potential for machine learning to learn the pattern behind the lab-scale kinetic test data. This study describes an approach to use artificial neural network (ANN) modeling to predict the result from lab-scale kinetic tests. Various ANN model is used based on 83 weeks experiments of lab-scale kinetic tests with 100% potential acid-forming rock. The model approaches the monitoring of pH, ORP, conductivity, TDS, sulfate, and heavy metals (Fe and Mn). The overall Nash-Sutcliffe Efficiency (NSE) obtained in this study was 0.99 on training and validation data, indicating a strong correlation and accurate prediction compared to the actual lab-scale kinetic tests data. This show the ANN ability to learn patterns, trends, and seasonality from past data for accurate forecasting, thereby highlighting its significant contribution to solving AMD problems. This research is also expected to establish the foundation for a new approach to predict AMD, with time efficient, accurate, and cost-effectiveness in future applications.

* Corresponding author: msa@itb.ac.id

1 Introduction

For many years, Artificial Neural Networks (ANNs) have emerged as an incredible tool, celebrated for their remarkable capability and robustness when dealing with complex datasets. ANNs, inspired by the human brain's neural structure [1], have witnessed rapid and major developments, proving their capabilities across many domains and setting new benchmarks as State-of-the-Art (SOTA) models in various modalities.

The potential of ANNs lies in their ability to comprehend intricate patterns and relationships within data, often transcending the limits of conventional analytical techniques. This paper explores whether this ability of ANNs can be implemented in the environmental engineering domain. Specifically, to investigate their utility in understanding and forecasting Acid Mine Drainage (AMD) through Long-Term Lab Scale Kinetic Tests.

AMD is strong, acidic wastewater rich in high concentrations of dissolved ferrous and non-ferrous metal sulfates and salts, and if AMD is left untreated, it can contaminate ground and surface watercourses, damaging the health of plants, humans, wildlife, and aquatic species [2]. Traditionally, AMD is analyzed using the kinetic test. This process involves simulating mine drainage production from samples influenced by mining activities, and incorporates dynamic elements encompassing physical, chemical, and biological systems and processes that govern the generation of acidic or alkaline mine drainage. Kinetic tests primarily focus on studying reaction rates and mechanisms leading to acidic or alkaline mine drainage, typically requiring larger sample volumes and extended durations than static tests, and are typically conducted in laboratory settings. These tests yield crucial information regarding sulfide mineral oxidation rates, acid production rates, and drainage water quality [3]. Usually, comprehending the complex kinetics of AMD generation and forecasting its behavior through kinetic tests has been laborious and resource-intensive. Consequently, developing cost-effective and sustainable remediation solutions for the AMD problem has been the subject of extensive research.

Predicting and forecasting analysis using ANN or Machine learning (ML) in general, has been used in various domain, including nonlinear timeseries and have gained overwhelming attention over the past years [4]. Forecasting mining influenced water data using various ML technique including tree based method and ANN show positive result, with close to accurate prediction [5]. Beside ML method, traditional model such as auto regressive integrated moving average (ARIMA), Box-Jenkins, etc. has been applied to this domain problem, but also coming with some drawback since traditional model assume that time series data are linear processes [6]. This study will focus on developing ANN model with various technique that suits the time series domain and will focus on capturing the relation between data over time.

2 Methodology

2.1 Feedforward neural network (FNN)

One of the most common type of ANN architecture is a feedforward neural network (FNN). This type of model consists of several layers of linear model, with each corresponding number of neurons. A linear model (Eq. 1) is the basic structure of any neural network model [1]. Each linear model with a non-linear activation function is called a neuron, and usually, every layer of FNN or ANN in general, consists of a few or maybe hundreds of neurons. Each neuron in a layer is connected to every neuron in the previous and next layers (Fig. 1). These are called dense or fully connected layers. FNN usually consist of input layer, a few hidden layer, and an output layer (Fig. 1). The information flows and processes in one

direction, from the input layer through the hidden layers to the output layer, without any loops or cycles.

$$f(\mathbf{x}, \mathbf{w}) = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_n \cdot w_n + b = \mathbf{x}^T \mathbf{w} + b \tag{1}$$

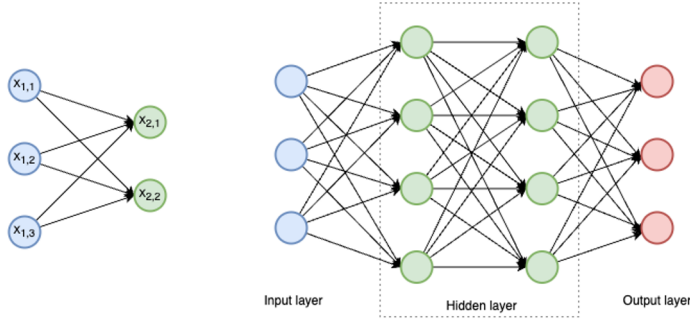


Fig. 1. (Left) interconnected neurons. (right) feedforward network architecture.

2.2 Multivariate long-short term memory (LSTM)

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) that differs from traditional RNN. LSTM cell uses a state that represents a "memory" or a "context" besides the inputs and the outputs [7], which aims to learn when to remember and when to forget pertinent information. LSTM contains three gates to control the dependencies, as shown in Fig. 3 an input gate to select the inputs, a forget gate to free some part of the memory, and an output gate to control the output. LSTM solve the vanishing gradient problem of traditional RNN when dealing with long sequences of data due to the application of Back Propagation Through Time (BPTT) for a specific Horizon [8]. LSTM consist of these operations.

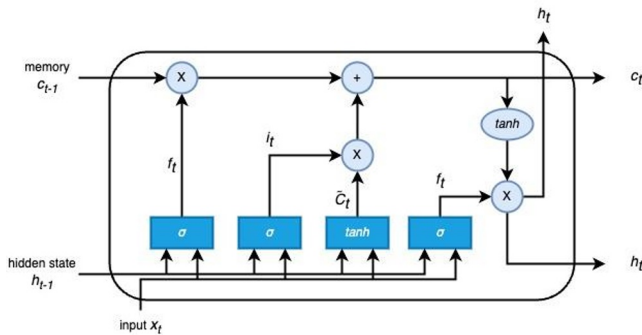


Fig. 2. LSTM mechanism.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\tilde{C}_t = \tanh(W \cdot [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * \tanh(C_t) \tag{7}$$

2.3 Encoder-decoder architecture

In the Encoder-Decoder architecture, there are two different inputs, the past feature values and the known current feature values. The Encoder part compresses the information from the first input sequence into a vector, which is generated from the sequence of the LSTM hidden states [9]. The encoder hidden states and also the second input feed into the decoder part and generate the output sequences. In addition, some dense layers were provided before the output layer to give a better prediction sequence. Fig. provide better explanation for the encoder-decoder architecture. This architecture also help learn the time-dependent characteristics of the sequence, give better prediction for future value [10].

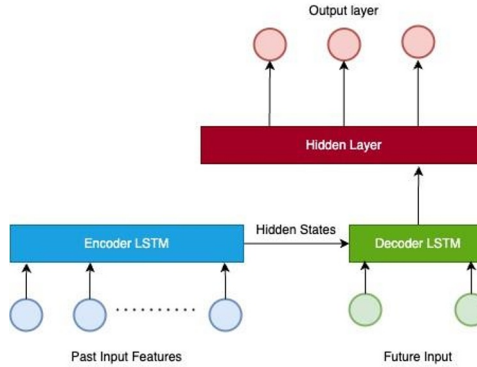


Fig. 3. Encoder-Decoder Architecture

2.4 Evaluation metrics

The models' performance in the validation dataset was assessed to determine the optimal model architecture for our research objective. The metrics used in this study as follows.

2.4.1 Mean squared error (MSE)

Mean Squared Error, is a fundamental metric in regression analysis that calculates the average of the squared differences between actual observation and predicted value. It provides insight into the precision of a predictive model by quantifying the average squared error across all data points. Smaller MSE values are desirable as they indicate a model that predicts closer to the actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

2.4.2 Mean absolute error (MAE)

Mean Absolute Error, is another widely used metric in regression analysis that measures the average of the absolute differences between predicted values and actual observations. MAE provides a straightforward way to understand the average magnitude of errors made by a predictive model. Unlike Mean Squared Error (MSE), MAE does not penalize large errors as heavily, making it more robust to outliers.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (9)$$

2.4.3 Nash-sutcliffe efficiency (NSE)

Nash-Sutcliffe Efficiency, assesses the goodness of fit between observed and simulated data. With values ranging from negative to 1, NSE quantifies the performance of a model in replicating observed data. An NSE of 1 represents a perfect match between model predictions and observed data, while values greater than 0 indicate that the model is outperforming the basic mean-based of the observed data. Conversely, negative NSE values show that the model performs worse than a basic mean-based estimate.

$$NSE = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (10)$$

3 Dataset

3.1 Dataset overview

The dataset for this study was acquired from one of the mining locations in Indonesia. Data was gathered between 09 February 2021 and 02 September 2022 every 7 days. This data contains 7 parameters, i.e. pH, redox potential (ORP), conductivity, total dissolved solids (TDS), SO₄, Fe, and Mn. Fig. 4 shows the graph visualizations of the collected data.

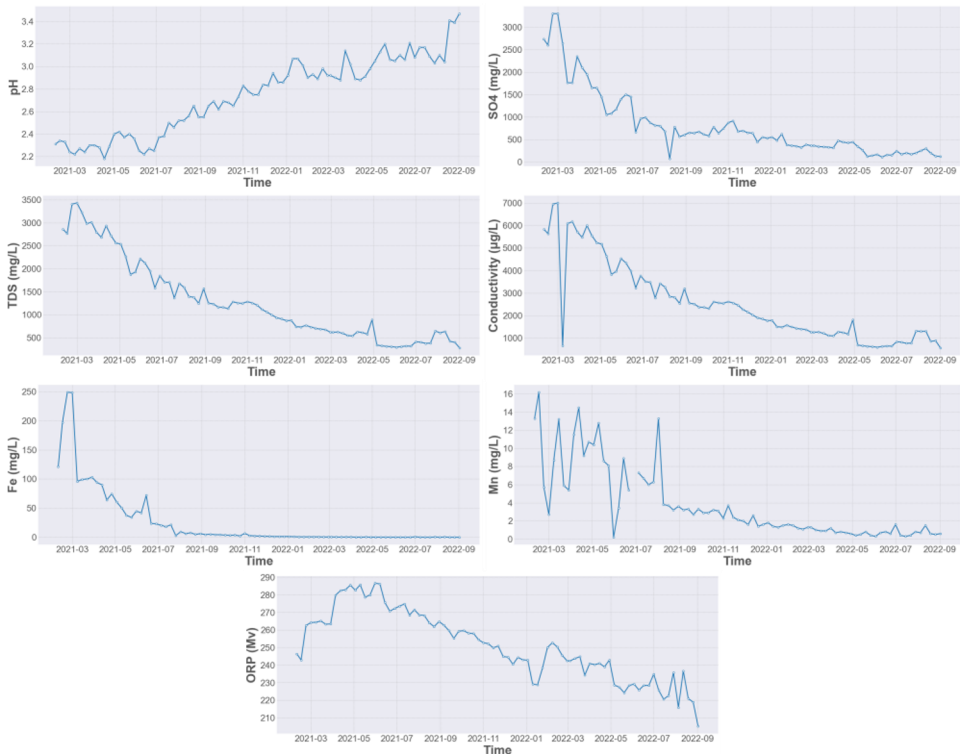


Fig. 4. Dataset visualization.

3.2 Stationarity test

Stationarity test is an important process in time series analysis and forecasting. A stationary condition of a time series data is when the characteristics of the data are unaffected by the observation time. Stationary data does not have trends and seasonality, thus making it easier to analyze and forecast [11]. Therefore, the Augmented Dickey-Fuller (ADF) test was used to test the stationarity of the dataset. The test was developed using a highly significant p -value (0.05) (Table 1).

According to the results, pH and ORP show non-stationary behavior as indicated by their p -values exceeding the critical threshold of 0.05. On the other hand, conductivity, TDS, SO₄, Fe, and Mn indicates stationarity, supported by their p -values below 0.05. This means that these parameters showcased consistent properties over time, devoid of trends or seasonality, making them more amenable to analysis and forecasting. Thus, the ADF test result indicates that the dataset is stationary.

Table 1. ADF test for stationarity

Parameter	ADF statistic	p -value
pH	-0.1143	0.9479
ORP	0.2394	0.9743
Conductivity	-4.3152	0.0004
TDS	3.9183	0.0019
SO ₄	-4.0869	0.0010
Fe	-6.0566	0.0000
Mn	-3.9184	0.0019

3.3 Anomaly detection

Anomalies are patterns in data that have different characteristics from expected conditions and often give bias to the data [12]. These anomalies can occur due to several factors, e.g., contamination, device calibration, human error, or other factors. Detecting anomalies has significant relevance and often improves performance during analysis and forecasting. Isolation forest, a model-based approach, is used for detecting anomalies, which constructs an ensemble of tree structures where anomalies are closer to the root, while normal points are deeper. This approach effectively detects anomalies with a small number of trees and minimal sub-sampling size, thus quick to convergence [13]. The isolation forest was implemented with contamination parameter of 0.2 (Fig. 5).

As indicated by their indices (0, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 16, 17, 18, 25, 82), anomalies were successfully identified using the isolation forest model with chosen contamination parameter of 0.2. These anomalies are characterized by different patterns characteristic from the expected conditions due to several factors. This successful anomaly detection is pivotal for improving the reliability and accuracy of our analysis and forecasting processes, ensuring a more robust and precise interpretation of the underlying patterns in the data.

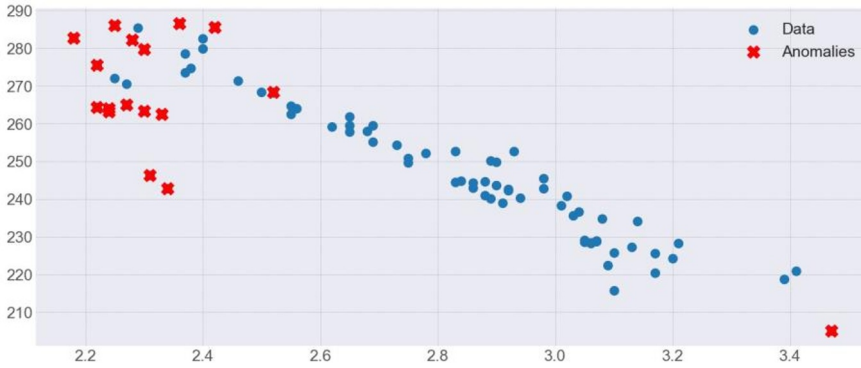


Fig. 5. Anomaly detection using isolation forest. Feature 1 (x-axis), feature 2 (y-axis).

3.4 Data interpolation and transformation

3.4.1 Data interpolation

Huge amounts of data are usually needed when training ANN or deep learning [14]. However, this dataset only contains 83 data points and gathered between 09 February 2021 and 02 September 2022 every 7 days. Predictive interpolation using random forest and tree-based gradient-boosting regression models were used to interpolate the missing data between the time intervals. Every model for each parameter were developed based on the time component with sine and cosine transformation. The data were split into training (80%) and testing (20%) sets. Each parameter is calculated based on the average of the top-3 interpolation results from the best-performing model (Table 2). Except for Mn parameter, where the value calculated from the result of random forest, XGBoost, and ExtraTrees.

Table 2. Predictive interpolation model performances.

Parameter	Model type							
	Random Forest		XGBoost		LigthGBM		ExtraTrees	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
pH	0.0006	0.0244	0.0008	0.0286	0.0015	0.0393	0.0008	0.0285
ORP	0.0004	0.0204	0.0005	0.0224	0.0010	0.0311	0.0004	0.0201
Conductivity	0.0041	0.0644	0.0025	0.0500	0.0109	0.1044	0.0069	0.0832
TDS	0.0042	0.0647	0.0025	0.0504	0.0112	0.1060	0.0070	0.0834
SO ₄	0.0073	0.0854	0.0057	0.0755	0.0130	0.1138	0.0107	0.1034
Fe	0.0015	0.0381	0.0014	0.0376	0.0030	0.0545	0.0018	0.0424
Mn	0.0127	0.1129	0.0117	0.1084	0.0084	0.0916	0.0158	0.1257

3.4.2 Data transformation

ANN or deep learning model usually prefer to receive inputs on the same scale. That is because ANN is just stacks of linear transforms with non linear activation function [1]. Thus, building forecasting models with untransformed data usually leads to in worst performance. Therefore, the data need to be transformed to close to normal distribution. The time component is also transformed using cosine transformation to give the information about time to the model.

4 Model development and evaluation

In this study, three types of model architecture, FNN, LSTM, and encoder-decoder LSTM were develop. Each of the types were trained with three different window size or the number of known past time-steps data the model needs to use to predict the current time-step, except for FNN where also trained without past data. Each of the types and window sizes were trained and tested with each independent set of data. The data were split into training (70%) and testing (30%) sets. All of the model are also trained to forecast all of the parameters. The models were evaluate based on MSE, MAE, and NSE values. The models were trained on the same batch size of 4, ReLU activation function for the output layer, MAE loss, and adaptive moment estimation (Adam) optimizer. The result were shown on Table 3. All model's hyper-parameters were tuned and optimized, and callbacks were also used to create the best performing model.

Table 3. ANN model structure and performances.

Model type	Structure variation		Performance metrics			
	Window size	Epochs		MSE	MAE	NSE
FNN	-	120	Train	0.0005	0.0193	0.9952
			Validation	0.0011	0.0289	0.9889
	7	80	Train	0.0006	0.0220	0.9945
			Validation	0.0014	0.0337	0.9861
	14	90	Train	0.0006	0.0212	0.9945
			Validation	0.0011	0.0310	0.9892
	28	120	Train	0.0004	0.0180	0.9961
			Validation	0.0017	0.0369	0.9837
LSTM	7	125	Train	0.0005	0.0188	0.9955
			Validation	0.0012	0.0296	0.9883
	14	210	Train	0.0006	0.0212	0.9943
			Validation	0.0010	0.0291	0.9903

Encoder-Decoder LSTM	28	225	Train	0.0004	0.0163	0.9959
			Validation	0.0013	0.0305	0.9871
	7	200	Train	0.0003	0.0150	0.9969
			Validation	0.0005	0.0213	0.9953
	14	250	Train	0.0004	0.0168	0.9961
			Validation	0.0005	0.0206	0.9955
	28	250	Train	0.0003	0.0164	0.9966
			Validation	0.0005	0.0206	0.9956

Each model was also tested to forecast the future parameter values for the next 60 days. To determine which which model is the best model of the others, the forecast result and plot given in Fig. 6, Fig. 7, and Fig. 8 can be used, also by given the known condition of train and validation loss and also by comparing to measured data. Generally, lower training and validation loss implies good model fit on the train data and new unknown validation data. Aside from lower training and validation loss, the good and fit model is defined by the distance between training loss and the validation loss. If the distance is close enough, the model is probably good fitted, if the distance is quite far, there a possibility of overfitted model, which is not a good model, and if the validation loss is lower than the training loss, that implies the possibility of underfitted model, which also not a good model.

5 Result and discussion

From the training and validation performance, encoder-decoder LSTM displayed the overall best performance compared to other types of models and structure variation. It shows a good model fit without overfitting to the dataset, Furthermore, the variation in window size shows some overall improvement, meaning the longer the sequence of past values, give better information to the model when predicting the current value. In addition to the model's performances, the results show that ANN models can be implemented in AMD forecasting and analysis.

The models were also evaluated by forecasting all features for 60 days (Fig. 6, Fig. 7, and Fig. 8). Because of the kinetic test measurement and observation were not carried out daily, the measured data contains only 9 observations, while the forecasting period was for 60 days, the calculated performance and error were based on the availability of measured data. Also, based on the performance of the encoder-decoder LSTM on training and validation, we only calculate the performance of that models, for all window size (Table 4). Based on that result, the encoder-decoder model with 7 days of past value show better performance. This was shown by the lower MSE and MAE values compared to other window size variation.

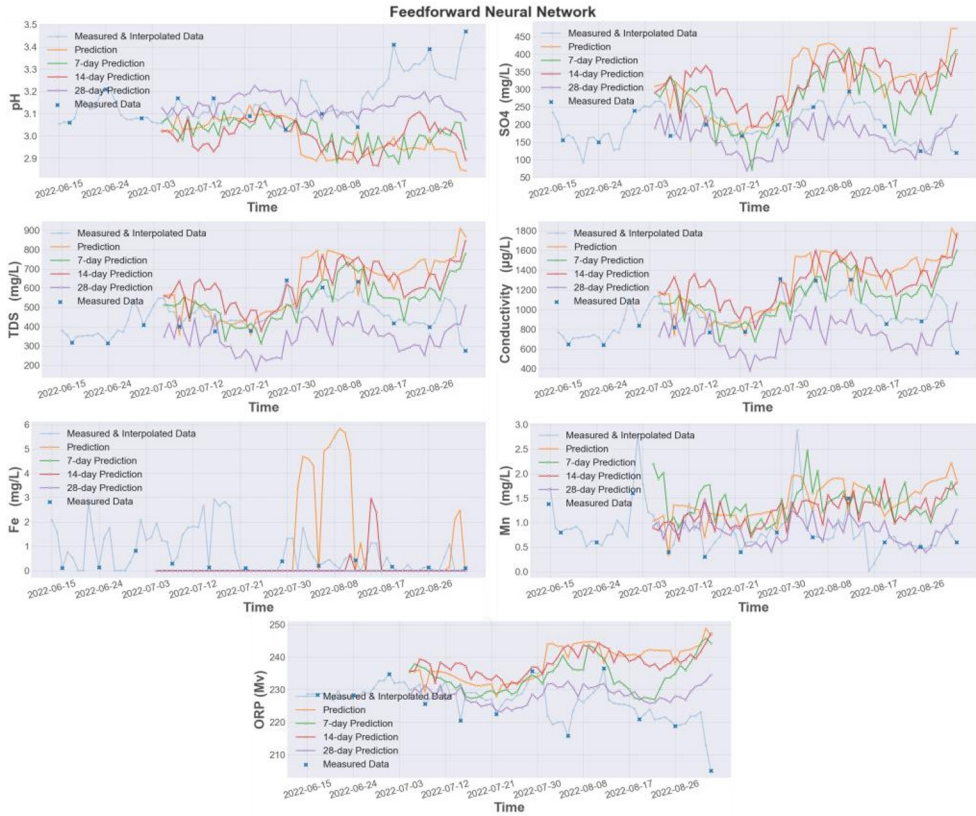


Fig. 6. Forecasted concentrations of all parameters for 60 days of Feedforward Neural Network. Historical data was used between 15 June 2022 to 02 September 2022 for better visualization.

Table 4. MSE and MAE for forecasted data.

Parameter	Window Size					
	7		14		28	
	MSE	MAE	MSE	MAE	MSE	MAE
pH	0.07	0.27	0.09	0.30	0.09	0.29
ORP	305.57	17.48	488.37	22.10	423.44	20.58
Conductivity	223985.57	473.27	387940.83	622.85	342765.42	585.46
TDS	56781.07	238.29	96171.42	310.12	85157.99	291.82
SO ₄	16202.92	127.29	15768.66	125.57	19956.18	141.27
Fe	0.06	0.24	0.06	0.24	0.06	0.24
Mn	0.66	0.81	1.17	1.08	1.31	1.14

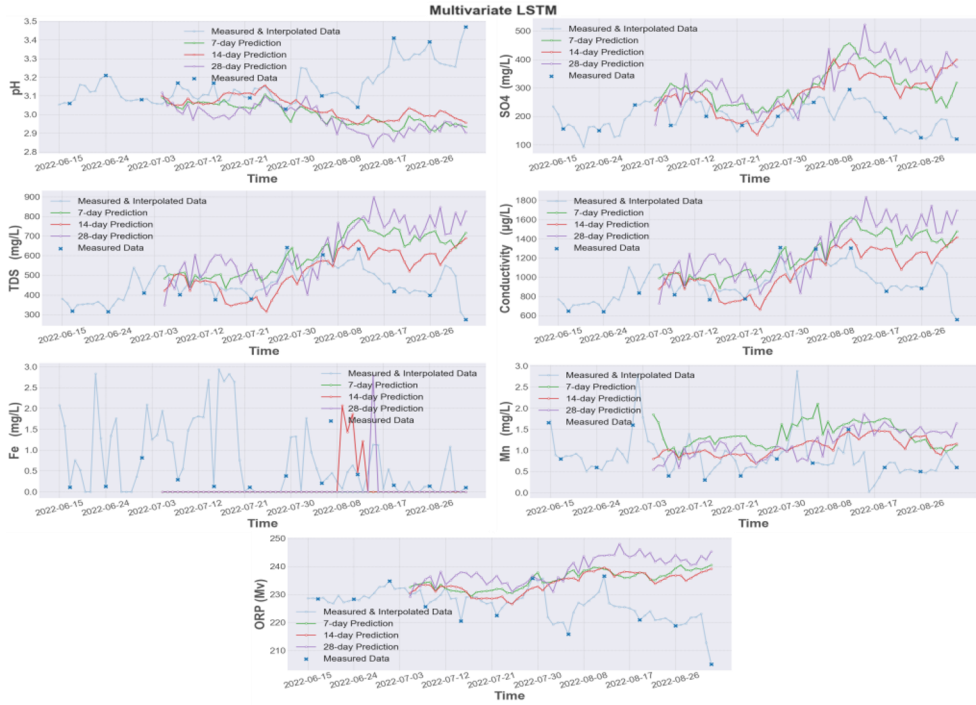


Fig. 7. Forecasted concentrations of all parameters for 60 days of Multivariate LSTM. Historical data was used between 15 June 2022 to 02 September 2022 for better visualization.

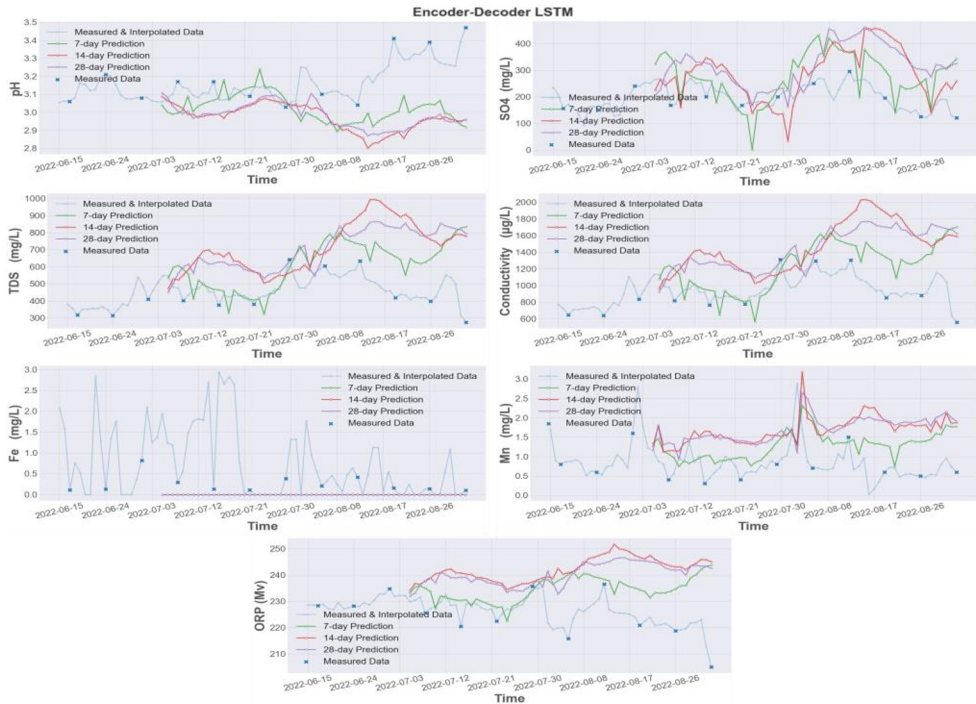


Fig. 8. Forecasted concentrations of all parameters for 60 days of Encoder-Decoder LSTM. Historical data was used between 15 June 2022 to 02 September 2022 for better visualization.

6 Conclusion

The forecasted data show low error measurements from MAE and MSE metrics and also follow the historical trend and pattern. Given this condition, it is safe and confident to implement the best-performing model from the proposed methodology to forecast AMD. This ANN approach shows that this technique is a powerful and good mechanism to model and forecast the future AMD data since the computer can learn patterns, trends, and seasonality of previous data in order to forecast the future.

This also can be concluded that by applying ANN approach is a relevant contribution and addition to solve AMD problems. In addition, these methods outperform conventional time series analysis and statistical techniques in terms of accuracy and performance. Finally, this study shows the ability to predict the actual lab-scale kinetic test in order to predict the AMD with a shorter time, high accuracy, and cost efficiency.

Acknowledgments

This study was supported by the Environment Department of PT Kaltim Prima Coal, Indonesia.

References

1. A. K. Jain, J. Mao, K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, 29, 31-44 (1996)
2. G. S. Simate, S. Ndlovu, "Acid mine drainage: Challenges and opportunities," *J. Environ. Chem. Eng.*, 2(3), 1785-1803 (2014)
3. L. Sibarani, "Prediction of mine drainage quality by kinetic tests," *Indones. Min. J.*, 9(3), 32-39 (2006)
4. A. Jain, A. M. Kumar, "Hybrid neural network models for hydrologic time series forecasting," *Appl. Soft Comput.*, 7, 585-592 (2007)
5. K. S. More, C. Wolkersdorfer, "Application of machine learning algorithms for nonlinear system forecasting through analytics-A case study with mining influenced water data," *Water Resour. Ind.*, 29, 100209 (2023)
6. G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, 50, 159-175 (2003)
7. S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, 9(8), 1735-1780 (1997)
8. A. Graves, "Supervised sequence labelling with recurrent neural networks," *Neural Networks*, 22, 37-45 (2012)
9. Bo Zhang et al., "A novel Encoder-Decoder model based on read-first LSTM for air pollutant prediction," *Science of the Total Environment*, 765, 144507 (2021)
10. Z. Wang, X. Su, Z. Ding, "Long-Term Traffic Prediction Based on LSTM EncoderDecoder Architecture," in *IEEE Trans. Intell. Transp. Syst.*, 22(10), 6561-6571 (2021)
11. R. J. Hyndman, G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed. (OTexts, Melbourne, 2018)
12. V. Chandola, A. Banerjee, V. Kumar, "Anomaly detection: A survey," in *ACM Comput. Surv.*, 41(3), 1-72 (2009)

13. F. T. Liu, K. M. Ting, Z. -H. Zhou, "Isolation Forest," in Proc. IEEE Int. Conf. Data Min.,Pisa, Italy, 2008, pp. 413-422
14. B. Barz, J. Denzler, "Deep learning on small datasets without pre-training using cosineloss," in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., pp. 1371-1380 (2020)