

# A Machine Learning Approach for Anaerobic Reactor Performance Prediction Using Long Short-Term Memory Recurrent Neural Network

Benjamin Steven Vien<sup>1, a\*</sup>, Leslie Wong<sup>1, b</sup>, Thomas Kuen<sup>2, c</sup>, L. R. Francis Rose<sup>3, d</sup>  
and Wing Kong Chiu<sup>1, e</sup>

<sup>1</sup>Department of Mechanical and Aerospace Engineering, Monash University, Wellington Rd, Clayton, VIC 3800, Australia

<sup>2</sup>Melbourne Water Corporation, 990 La Trobe Street, Docklands, VIC 3008, Australia

<sup>3</sup>Defence Science and Technology Group, 506 Lorimer Street, Fishermans Bend, VIC 3207, Australia

<sup>a</sup>ben.vien@monash.edu, <sup>b</sup>Leslie.Wong@monash.edu, <sup>c</sup>Thomas.Kuen@melbournewater.com.au  
<sup>d</sup>Francis.Rose@dst.defence.gov.au, <sup>e</sup>wing.kong.chiu@monash.edu

**Keywords:** Machine Learning, Data Analysis, Anaerobic Reactor, Data Preparation, Long Short-Term Memory, Artificial Neural Network

**Abstract.** Predictive models are important to help manage high-value assets and to ensure optimal and safe operations. Recently, advanced machine learning algorithms have been applied to solve practical and complex problems, and are of significant interest due to their ability to adaptively ‘learn’ in response to changing environments. This paper reports on the data preparation strategies and the development and predictive capability of a Long Short-Term Memory recurrent neural network model for anaerobic reactors employed at Melbourne Water’s Western Treatment Plant for sewage treatment that includes biogas harvesting. The results show rapid training and higher accuracy in predicting biogas production when historical data, which include significant outliers, are preprocessed with z-score standardisation in comparison to those with max-min normalisation. Furthermore, a trained model with a reduced number of input variables via the feature selection technique based on Pearson’s correlation coefficient is found to yield good performance given sufficient dataset training. It is shown that the overall best performance model comprises the reduced input variables and data processed with z-score standardisation. This initial study provides a useful guide for the implementation of machine learning techniques to develop smarter structures and management towards Industry 4.0 concepts.

## Introduction

Melbourne Water’s self-powered Western Treatment Plant (WTP) at Werribee, Victoria, Australia [1], provides essential sewage treatment and under normal operating condition, its biological digestion process produces 65,000m<sup>3</sup> methane-rich biogas per day. This biogas is harvested and used to generate 7MW renewable electrical energy per day which is worth \$8 million (AUD) per year. The anaerobic treatment lagoons are covered by 2-mm thick high-density polyethylene (HDPE) floating sheets, approximately 450m x 170m, to capture odorous and greenhouse gases produced by the bacteria in the sewage under the covers. The untreated raw sewage content, such as fats, oil, floating solids and buoyed sludge or other fibrous material, may be carried to the surface of the lagoon by small bubbles, potentially forming into a solid mass called ‘scum’. The scum in the anaerobic lagoon is a mixture of floating solids, undigested

sludge and trapped gases formed during anaerobic digestions. This scum can accumulate under the covers, amassing into a large iceberg-like body, which is called ‘*scumberg*’. The formulation of *scumbergs* has the potential to adversely impact the structural integrity of the HDPE floating covers and block the pathway of biogas, thereby affecting the WTP performance. Due to the complex nature of the scum accumulation and *scumbergs* formation, it is very difficult to develop analytical or rule-based computing models for forecasting their effects on the performance of the anaerobic reactors and the structural integrity of the HDPE floating covers. Consequently, the only reliable performance data can only be obtained from the actual measurements at the sewage processing plant as the processes cannot reliably be scaled down to a laboratory-sized simulation. Therefore, there is a need for ‘smart’ real-time monitoring and diagnostic-prognostic capability modelling for the operation and management of this high-value asset based on current and/or historical operational data.

In many practical cases, physical systems for which explicit rules are either unknown or too difficult to determine, cannot be accurately modelled using traditional computing methods. In the last few decades, machine learning (ML) techniques have been extensively employed in engineering applications, which include structural engineering [2-4], water reservoir operations [5, 6], and structural health monitoring [7-9]. Artificial neural networks (ANN) are one of the machine learning-based algorithms that mimic the information processing and knowledge acquisition in the human brain. ANN is highly desirable due to its ability to model nonlinearity and predict accurately even in the presence of noisy and incomplete data of the real-world system. Furthermore, it can adaptively update its model in response to changing environments over time [10, 11] which makes ANNs an ideal candidate for solving complex engineering problems. However, ANNs are unable to provide justifications for their solutions and can be unpredictably inaccurate when extrapolating solutions for problems outside the network training domain.

A recurrence neural network (RNN) is a special type of ANN for sequential data modelling which store some past information. However, traditional RNNs have the problem of gradient vanishing/exploding and lack of long-term memory ability [12]. Long Short-Term Memory (LSTM) network was first proposed by Hochreiter and Schmidhube [13] to overcome the limitation of RNNs. LSTM network can effectively learn long-term dependencies between time steps and is superior to most RNN prediction methods [13]. In short, the difference between the traditional RNN and LSTM is the internal operation of the recurrent cell. In a traditional RNN, only one internal state exists and it is recomputed in every time step, whereas an LSTM cell has an additional self-connected memory cell state in which information can be stored; these memory cells are managed by cell gates, allowing learning of long-term dependencies [13]. There is significant interest in using LSTM network architecture for time-series forecasting for various engineering application including wind-power, solar power and electric load [14-17]. This architecture can also be expected to be highly advantageous for producing data-driven and adaptive predictive models based on ANN techniques for optimal biogas harvesting while ensuring the structural integrity of the floating covers.

In this paper, an LSTM network architecture, which comprises one sequence input layer followed by one LSTM layer, a fully-connected layer (consisting of one neuron) and an output layer, is developed and demonstrated to predict the biogas production of WTP anaerobic reactor. The investigation includes a parametric study to design the LSTM network topology, which includes data pre-processing, optimising the number of hidden units in an LSTM layer and training parameters, refer to Figure 1. A historical data sample from WTP is partitioned to train

the LSTM network input and recurrent weights and biases and to evaluate the trained models' performance. In this study, MATLAB R2020a was used to develop the LSTM network. This study reports on the data preparation of the real-life dataset of the anaerobic reactor and environment readings and the development of LSTM network architecture for biogas production prediction.

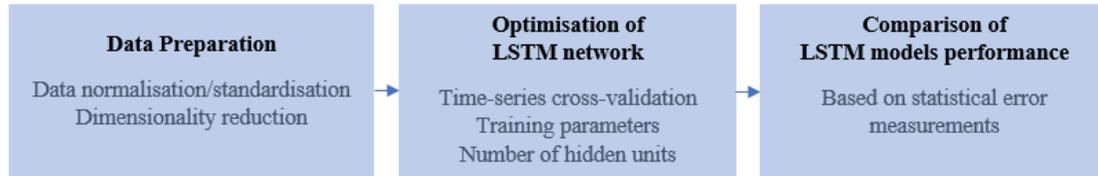


Figure 1: Schematic depiction of the procedure for the development and evaluation of LSTM models.

## Method

### Data Preparation

A real-world historical dataset of the 25W anaerobic reactor at WTP consists of 14 variables with 365 daily readings from November 2018 to October 2019 as collected by Melbourne Water (shown in Table 1), was used to train and develop the LSTM prediction models. The output variable is biogas production, whereas the remaining variables are treated as inputs for the prediction. Data pre-processing is an important stage for designing ML models, which primarily entails transforming raw data into a clean and useable format. Data normalisation/standardisation is a standard pre-processing technique that changes the values in a dataset to a common scale without misrepresenting the difference in the range of variables. For RNN/LSTM models, this is a crucial procedure to ensure stability and improve network training and performance [17]. In this study, two common data scaling techniques are applied and investigated: z-score standardisation (standardised variables), rescales data to have zero mean and standard deviation of 1 and min-max normalisation (normalised variables), linearly transforms data in the range [-1, 1] [18].

Normally, it is advantageous to reduce the number of inputs by removing redundant variables to improve the speed learning algorithm and eliminate overfitting to a degree. A feature selection technique aims to reduce the number of features (inputs) by using a correlation-based method to filter the correlated variables. In this study, Pearson's correlation coefficient is employed to measure the association between the input variables (refer to Figure 2). Based on the correlation matrix, 7 inputs variables, viz. average  $I_{main}$  average and minimum  $I_{s1}$ , average and minimum  $I_{s2}$ , average temperature  $T_A$  and average rainfall  $R$ , are selected for the reduced number of input variables (relative to the original 14 input variables) for the LSTM network model investigation.

Table 1: Historical WTP 25W anaerobic reactor daily data.

MAIN VARIABLES	UNITS	VARIABLES	MEAN	MAX	MIN	STANDARD DEVIATION	QUARTILE 1 (25%)	QUARTILE 2 (50%)	QUARTILE 3 (75%)
Main Inlet $I_{main}$	ML/d	Average	269.1	361.2	151.2	27.9	250.6	266.0	286.3
		<b>MAXIMUM</b>	<b>356.3</b>	<b>422.0</b>	<b>183.0</b>	<b>23.4</b>	<b>347.0</b>	<b>363.0</b>	<b>367.0</b>
		Minimum	120.4	344.0	0.0	47.3	95.0	113.0	139.0
SECONDARY INLET 1 $I_{s1}$	ML/D	<b>AVERAGE</b>	<b>7.6</b>	<b>123.2</b>	<b>0.0</b>	<b>16.3</b>	<b>0.0</b>	<b>1.1</b>	<b>7.2</b>
		Maximum	49.4	353.0	0.0	67.6	0.0	22.0	72.0
		<b>MINIMUM</b>	<b>0.04</b>	<b>15.0</b>	<b>0.0</b>	<b>0.8</b>	<b>0.0</b>	<b>0.0</b>	<b>0.0</b>
Secondary Inlet 2 $I_{s2}$	ML/d	Average	4.3	208.2	0.0	18.2	0.1	1.0	2.7
		<b>MAXIMUM</b>	<b>35.8</b>	<b>591.0</b>	<b>0.0</b>	<b>94.9</b>	<b>1.0</b>	<b>2.0</b>	<b>4.0</b>
		Minimum	0.6	4.0	0.0	1.0	0.0	0.0	1.0
ATMOSPHERIC TEMPERATURE $T_A$	°C	<b>AVERAGE</b>	<b>17.2</b>	<b>33.6</b>	<b>6.6</b>	<b>5.2</b>	<b>12.6</b>	<b>16.4</b>	<b>20.9</b>
		Maximum	25.9	46.0	11.0	6.9	21.0	25.0	31.0
		<b>MINIMUM</b>	<b>11.5</b>	<b>25.0</b>	<b>2.0</b>	<b>4.7</b>	<b>8.0</b>	<b>11.0</b>	<b>15.0</b>
Rainfall $R$	mm	Average	1.4	81.8	0.0	5.2	0.0	0.0	0.6
<b>BIOGAS PRODUCTION <math>g</math></b>	<b>NM<sup>3</sup>/HR</b>	<b>AVERAGE</b>	<b>2868.0</b>	<b>3825.5</b>	<b>35.8</b>	<b>468.8</b>	<b>2548.3</b>	<b>2876.7</b>	<b>3162.7</b>

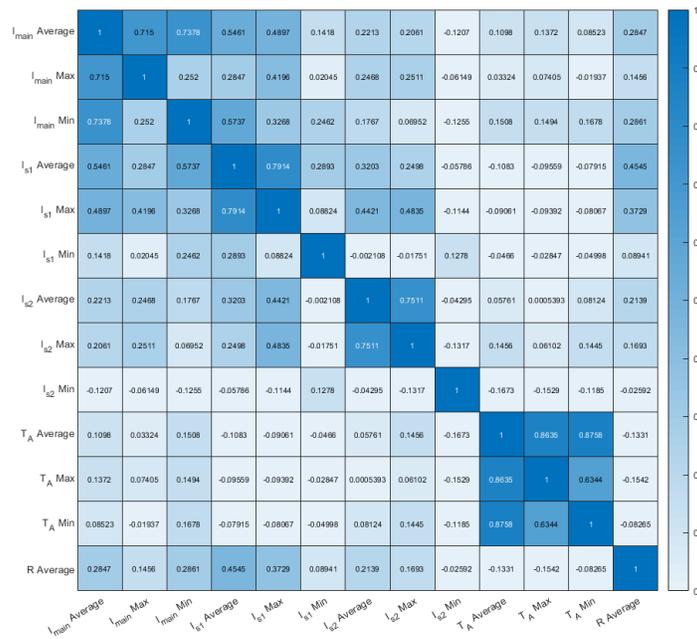


Figure 2: Heatmap of Pearson's correlation matrix on the input variables.

Model Validation

To ensure generalisation from the training data, it is essential that part of the test data set is preserved, unseen by the ML during training. Otherwise, the model is likely to overfit, yielding high accuracy on the training data, but failing to generalise from the training dataset, thereby resulting in poor predictive performance on new data. To evaluate trained ML models, *k*-fold

cross-validation technique is a common resampling procedure that randomly splits the dataset into  $k$  groups then trains the model on all groups except one that is reserved for testing the model [19]. Chronological ordered cross-validation, where training sets consist of observations that occur prior to those that form the testing set, is more suitable for time-series and sequential data modelling [20]. This study employed an expanding window (also known as forward-chaining) cross-validation with datasets partitioned into 6 nearly evenly distributed train-test sets (approximately 60 data points), equating to a 5-split (iteration) procedure, refer to Figure 3. The average error values of the 5 splits were used to evaluate the models' overall performance.

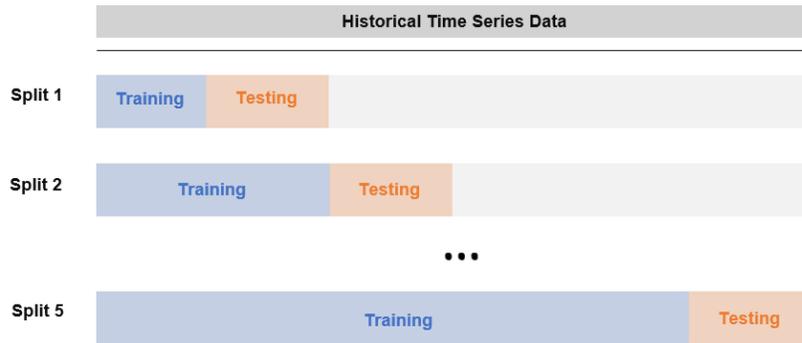


Figure 3: Expanding window 5-split time-series cross-validation.

The performances of LSTM models were compared and assessed based on the statistical error measurements; mean square error (MSE), mean absolute error (MAE) [21] and coefficient of determination ( $R^2$ ). It should be noted that the error measurements are calculated based on the standardised and normalised output variables. The training of the neural network used MSE as the loss function and stochastic gradient descent with momentum as the optimisation algorithm, which introduces learning variables, the learning rates and momentum parameter [22], to overcome problems of slow or non-convergence encountered in traditional gradient descent methods.

In this study, the several LSTM models were developed using standardised data with full and reduced input variables denoted as SFIV and SRIV models, respectively, and normalised data with full and reduced input variables denoted as NFIV and NRIV models, respectively. The study firstly investigated the optimal training parameters and the number of hidden units of a single LSTM layer for each model. Furthermore, the performances based on the error measurements for each best performing model were compared.

## Results and Discussion

Firstly, the optimum number of epoch (the number of times that the whole dataset is passed to the network) was determined by evaluating the average MSE, MAE and  $R^2$  of the 5 splits. In this investigation, a 10-hidden-unit LSTM layer was considered with epoch varying from 10 to 2000. The initial learning rate was 0.01, and the learning rate schedule was set to *piecewise*, where the learning rate was reduced by a factor of 0.2 after half of the number epochs had passed and the gradient threshold/clipping was 1. The optimal epoch for both SFIV and SFIV models are 50, and for NFIV and NRIV models are 100 and 200, respectively, refer to Table 2. Below or beyond these optimal epoch values will likely lead to underfit or overfit the trained models. It should be noted that inspection of learning curves (loss error over epoch/iteration) is necessary to monitor the behaviour of the model and ensure convergence.

Table 2: Optimisation of the number of epochs for SFIV, SRIV, NFIV and NRIV LSTM models.

Number of Epochs	Standardised						Normalised					
	Full Input Variables			Reduced Input Variables			Full Input Variables			Reduced Input Variables		
	MAEave	MSEave	R <sup>2</sup>	MAEave	MSEave	R <sup>2</sup>	MAEave	MSEave	R <sup>2</sup>	MAEave	MSEave	R <sup>2</sup>
10	285.00	1.32E+05	0.10	365.31	2.12E+05	0.10	869.52	9.88E+05	0.09	969.90	1.14E+06	0.08
20	286.40	1.40E+05	0.12	278.83	1.31E+05	0.12	407.43	2.81E+05	0.04	372.13	2.08E+05	0.11
50	<b>234.69</b>	<b>9.07E+04</b>	<b>0.19</b>	<b>243.28</b>	<b>9.86E+04</b>	0.14	312.79	1.66E+05	0.09	286.55	1.37E+05	0.12
100	288.34	1.40E+05	0.15	264.63	1.22E+05	0.18	<b>247.24</b>	<b>1.03E+05</b>	0.15	271.95	1.17E+05	0.16
200	331.70	1.84E+05	0.13	316.57	1.62E+05	<b>0.23</b>	270.04	1.22E+05	0.11	<b>250.78</b>	<b>1.01E+05</b>	0.14
500	387.92	2.59E+05	0.09	299.14	1.53E+05	0.12	285.78	1.33E+05	0.09	253.47	1.11E+05	0.15
1000	357.40	2.18E+05	0.10	404.02	2.66E+05	0.09	274.00	1.24E+05	0.18	265.73	1.16E+05	0.08
2000	413.71	3.11E+05	0.07	451.33	3.35E+05	0.06	343.56	2.09E+05	<b>0.19</b>	337.19	1.89E+05	<b>0.26</b>

Table 3: Error measurements for the different number of hidden units in one LSTM layer for SFIV, SRIV, NFIV and NRIV models. **Underlined** and **bolded** are the best and top three results, respectively, according to the error measurements.

Number of hidden units	Standardised						Normalised					
	Full Input Variables			Reduced Input Variables			Full Input Variables			Reduced Input Variables		
	MAEave	MSEave	R <sup>2</sup>	MAEave	MSEave	R <sup>2</sup>	MAEave	MSEave	R <sup>2</sup>	MAEave	MSEave	R <sup>2</sup>
1	252.08	1.06E+05	0.11	289.21	1.37E+05	0.18	266.65	1.19E+05	<b>0.15</b>	262.72	1.10E+05	<b>0.20</b>
2	236.63	8.47E+04	0.19	301.30	1.46E+05	0.13	251.98	1.04E+05	0.12	249.69	9.84E+04	0.19
3	272.17	1.23E+05	0.12	249.26	1.05E+05	0.15	252.15	9.94E+04	0.13	262.57	1.18E+05	0.14
4	253.41	1.03E+05	<b>0.21</b>	239.56	9.10E+04	0.20	242.12	9.39E+04	0.12	257.48	1.09E+05	0.16
5	264.54	1.16E+05	0.12	269.79	1.22E+05	0.18	255.78	1.06E+05	0.14	262.08	1.15E+05	0.15
6	268.25	1.19E+05	0.12	<b>211.63</b>	<b>7.12E+04</b>	<b>0.23</b>	<b>240.75</b>	<b>9.17E+04</b>	0.14	248.19	1.01E+05	<b>0.24</b>
7	240.02	9.66E+04	0.17	277.97	1.35E+05	0.18	276.55	1.29E+05	0.06	<b>246.79</b>	<b>9.68E+04</b>	0.14
8	243.68	9.36E+04	0.16	256.87	1.12E+05	0.18	245.00	9.93E+04	0.15	<b>246.51</b>	9.82E+04	<b>0.20</b>
9	269.03	1.19E+05	0.16	244.66	1.01E+05	0.17	267.92	1.17E+05	0.09	<b>239.89</b>	<b>9.12E+04</b>	0.17
10	234.69	9.07E+04	0.19	243.28	9.86E+04	0.14	247.24	1.03E+05	<b>0.15</b>	250.78	1.01E+05	0.14
11	<b>220.62</b>	<b>7.86E+04</b>	<b>0.21</b>	254.31	1.07E+05	0.20	246.10	9.74E+04	0.10	254.13	1.09E+05	0.12
12	232.63	9.27E+04	0.12	<b>215.77</b>	<b>7.70E+04</b>	<b>0.22</b>	267.47	1.12E+05	0.13	248.03	9.97E+04	0.16
13	247.47	9.80E+04	0.11	239.92	9.44E+04	0.21	266.36	1.10E+05	0.12	262.21	1.13E+05	0.13
14	<b>226.39</b>	<b>8.46E+04</b>	0.19	273.90	1.20E+05	0.12	<b>241.01</b>	9.70E+04	0.13	249.87	<b>9.68E+04</b>	0.17
15	237.23	8.83E+04	<b>0.22</b>	245.71	9.58E+04	0.16	253.61	9.76E+04	0.08	269.82	1.17E+05	0.16
16	<b>227.06</b>	<b>8.11E+04</b>	0.20	268.90	1.19E+05	0.17	<b>238.66</b>	<b>9.28E+04</b>	<b>0.18</b>	254.71	1.03E+05	0.15
17	249.80	9.94E+04	0.17	<b>219.38</b>	<b>8.10E+04</b>	<b>0.24</b>	251.85	9.88E+04	0.08	249.27	9.79E+04	0.13
18	260.56	1.13E+05	0.15	246.12	1.00E+05	0.15	306.20	1.43E+05	0.06	262.87	1.07E+05	0.12
19	251.33	1.02E+05	0.20	255.63	1.06E+05	0.13	243.84	<b>9.28E+04</b>	0.11	249.16	9.96E+04	0.16
20	281.16	1.26E+05	0.14	235.06	8.63E+04	0.19	258.16	1.11E+05	0.11	258.39	1.03E+05	0.14

Based on the error measurements, the optimal number of hidden units in a one LSTM layer for SFIV and SRIV models are 11 and 6, respectively and for NFIV and NRIV models are 6 and 9, respectively, as indicated in Table 3. The coefficients of determination were compared to evaluate the models with different pre-processed data. It is shown that NFIV and NRIV models performed relatively poorly compared to data pre-processed with z-score standardisation, refer to

Tables 3 and 4, and this is evident in Figures 4 and 5. This is because extreme outliers exist in the historical data and are retained during the learning process. In practice, data standardisation is more robust in handling outliers/extremities or non-uniformly distributed data than min-max normalisation. Furthermore, the models with normalised variables took relatively longer to train.

Table 4: Error measurement of the best performing LSTM models for each split cross-validation.

Best Performing Model	Error	k <sup>th</sup> Split				
		1	2	3	4	5
Standardised Full Input Variables 11-hidden-unit LSTM layer	MAE	245.53	219.31	207.35	221.90	209.04
	MSE	7.58E+04	9.82E+04	6.92E+04	8.46E+04	6.55E+04
	R <sup>2</sup>	0.25	0.31	0.18	0.00	0.30
Standardised Reduced Input Variables 6-hidden-unit LSTM layer	MAE	227.76	228.51	219.14	188.64	194.11
	MSE	6.79E+04	9.99E+04	7.17E+04	6.11E+04	5.55E+04
	R <sup>2</sup>	0.30	0.40	0.06	0.04	0.37
Normalised Full Input Variables 6-hidden-unit LSTM layer	MAE	243.54	265.65	241.49	224.25	228.84
	MSE	7.65E+04	1.26E+05	8.67E+04	9.17E+04	7.75E+04
	R <sup>2</sup>	0.15	0.35	0.04	0.00	0.13
Normalised Reduced Input Variables 9-hidden-unit LSTM layer	MAE	264.45	230.47	227.97	243.22	233.33
	MSE	9.08E+04	1.03E+05	7.89E+04	9.61E+04	8.79E+04
	R <sup>2</sup>	0.11	0.28	0.22	0.17	0.09

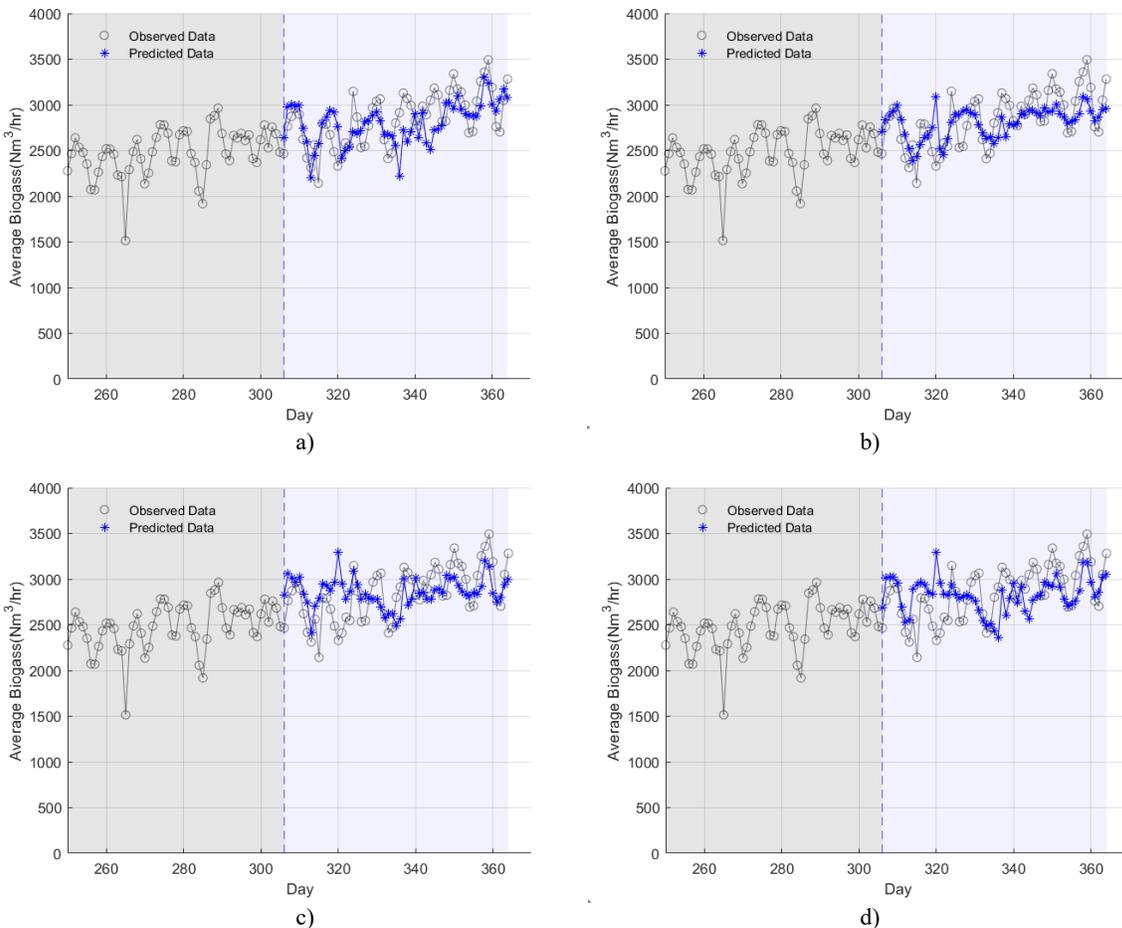


Figure 4: 5<sup>th</sup> split predicted biogas production of the best performing LSTM models: a) SFIV, b) SRIV, c) NFIV and d) NRIV models.

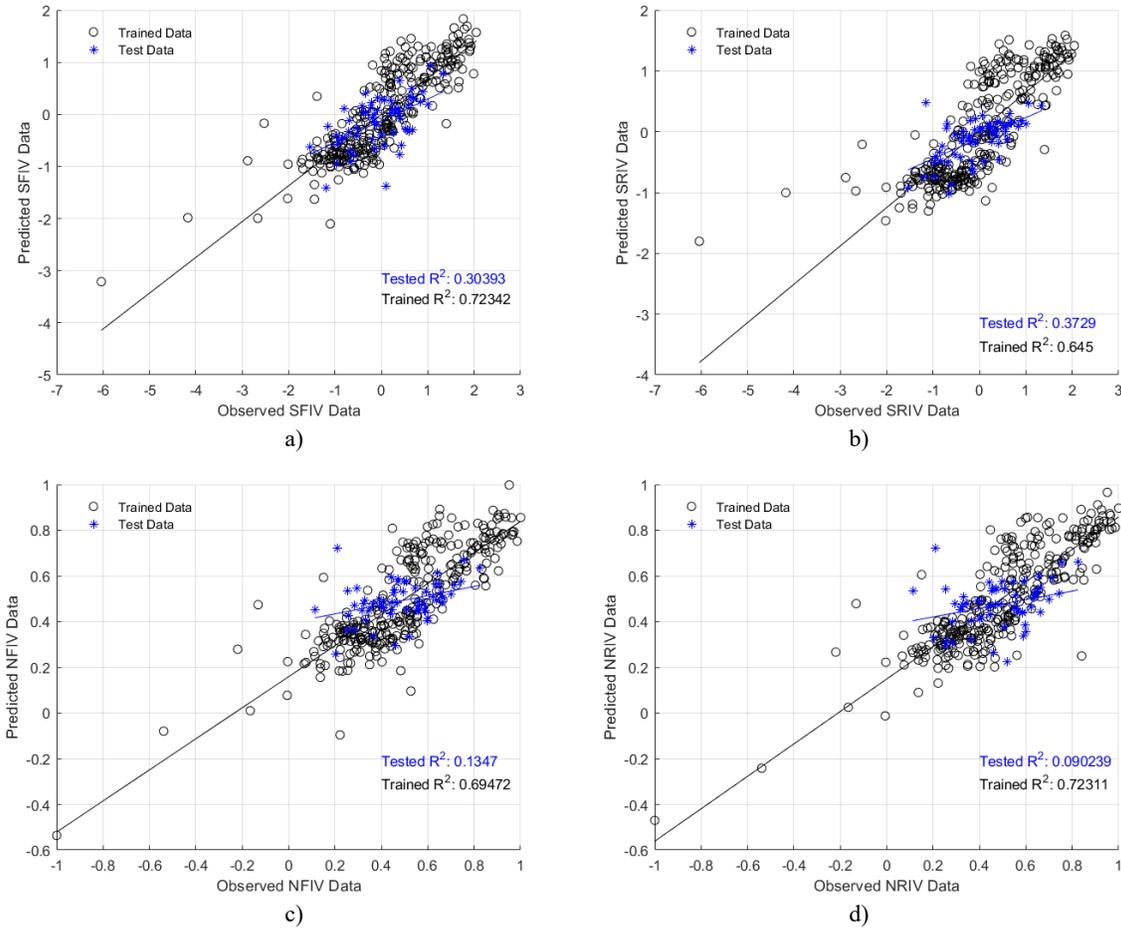


Figure 5: Predicted and observed plots of the best performing LSTM models on the 5<sup>th</sup> split; a) SFIV, b) SRIV, c) NFIV and d) NRIV models.

The best LSTM model is the SRIV model and it is shown that the other best performing models can also generalise and predict with good accuracy after the last split (refer to Table 4). The substantial improvement in the models' performance on the last split suggests most of the dependencies and key features can be learnt in the first four data partition sets.

### Conclusion

This study has reported on the development of an ANN architecture for predicting the performance of WTP anaerobic reactor. With the currently available dataset, the findings have shown that an LSTM network can be utilised to predict biogas production. It is shown that the prediction model with data standardisation yields higher accuracy and outperforms the max-min normalisation for the WTP historical dataset, which included significant outliers. The LSTM model trained by using standardised data with reduced input variables yields the best average performance on all splits. It is shown that the LSTM predicting model with a reduced number of input variables via Pearson's correlation coefficient selection method can achieve good accuracy given sufficient dataset training. Ongoing studies on the data preparation and development of ML algorithms and their architectures for WTP performance forecasting as well as the monitoring of floating cover structural integrity are underway in integrating AI-enable approach for future management and operation of this critical asset.

## Acknowledgments

Financial supports provided by Australia Research Council (Linkage Project Grant) (ARC LP170100108) and Melbourne Water Corporation are gratefully acknowledged. The in-kind contributions from Melbourne Water are also gratefully acknowledged.

## Reference

- [1] Melbourne Water. *Western Treatment Plant Virtual Tour*. 2020 30 September 2020 [cited 2020 30 November]; Available from: <https://www.melbournewater.com.au/water-data-and-education/learning-resources/water-and-sewage-treatment-plants/western-treatment-0>.
- [2] Madhiarasan, M. and S. Deepa, A novel criterion to select hidden neuron numbers in improved back propagation networks for wind speed forecasting. *Applied intelligence*, 2016. 44(4): p. 878-893. <https://doi.org/10.1007/s10489-015-0737-z>
- [3] Moosazadeh, S., E. Namazi, H. Aghababaei, A. Marto, H. Mohamad, and M. Hajihassani, *Prediction of building damage induced by tunnelling through an optimized artificial neural network*. *Engineering with Computers*, 2019. 35(2): p. 579-591. <https://doi.org/10.1007/s00366-018-0615-5>
- [4] Liu, H. and Y. Zhang, Deep learning-based brace damage detection for concentrically braced frame structures under seismic loadings. *Advances in Structural Engineering*, 2019. 22(16): p. 3473-3486. <https://doi.org/10.1177/1369433219859389>
- [5] Ehsani, N., B.M. Fekete, C.J. Vörösmarty, and Z.D. Tessler, *A neural network based general reservoir operation scheme*. *Stochastic environmental research and risk assessment*, 2016. 30(4): p. 1151-1166. <https://doi.org/10.1007/s00477-015-1147-9>
- [6] Maier, H.R., A. Jain, G.C. Dandy, and K.P. Sudheer, Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environmental modelling & software*, 2010. 25(8): p. 891-909. <https://doi.org/10.1016/j.envsoft.2010.02.003>
- [7] Deng, L. and D. Yu, *Deep learning: methods and applications*. *Foundations and trends in signal processing*, 2014. 7(3-4): p. 197-387. <https://doi.org/10.1561/20000000039>
- [8] Liu, H. and Y. Zhang, Deep learning based crack damage detection technique for thin plate structures using guided lamb wave signals. *Smart Materials and Structures*, 2019. 29(1): p. 015032. <https://doi.org/10.1088/1361-665X/ab58d6>
- [9] Agarwal, S. and M. Mitra, Lamb wave based automatic damage detection using matching pursuit and machine learning. *Smart materials and structures*, 2014. 23(8): p. 085012. <https://doi.org/10.1088/0964-1726/23/8/085012>
- [10] Basheer, I.A. and M. Hajmeer, *Artificial neural networks: fundamentals, computing, design, and application*. *Journal of microbiological methods*, 2000. 43(1): p. 3-31. [https://doi.org/10.1016/S0167-7012\(00\)00201-3](https://doi.org/10.1016/S0167-7012(00)00201-3)
- [11] Engelbrecht, A.P., *Computational intelligence: an introduction*. 2007: John Wiley & Sons.

- [12] Bengio, Y., P. Simard, and P. Frasconi, *Learning long-term dependencies with gradient descent is difficult*. IEEE transactions on neural networks, 1994. 5(2): p. 157-166. <https://doi.org/10.1109/72.279181>
- [13] Hochreiter, S. and J. Schmidhuber, *Long short-term memory*. Neural computation, 1997. 9(8): p. 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [14] Gensler, A., J. Henze, B. Sick, and N. Raabe. Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks. in 2016 IEEE international conference on systems, man, and cybernetics (SMC). 2016. IEEE. <https://doi.org/10.1109/SMC.2016.7844673>
- [15] Bouktif, S., A. Fiaz, A. Ouni, and M.A. Serhani, Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches. Energies, 2018. 11(7): p. 1636. <https://doi.org/10.3390/en11071636>
- [16] Yu, R., J. Gao, M. Yu, W. Lu, T. Xu, M. Zhao, J. Zhang, R. Zhang, and Z. Zhang, *LSTM-EFG for wind power forecasting based on sequential correlation features*. Future Generation Computer Systems, 2019. 93: p. 33-42. <https://doi.org/10.1016/j.future.2018.09.054>
- [17] Le, X.-H., H.V. Ho, G. Lee, and S. Jung, Application of long short-term memory (LSTM) neural network for flood forecasting. Water, 2019. 11(7): p. 1387. <https://doi.org/10.3390/w11071387>
- [18] Han, J., J. Pei, and M. Kamber, *Data mining: concepts and techniques*. 2011: Elsevier.
- [19] Rodriguez, J.D., A. Perez, and J.A. Lozano, *Sensitivity analysis of k-fold cross validation in prediction error estimation*. IEEE transactions on pattern analysis and machine intelligence, 2009. 32(3): p. 569-575. <https://doi.org/10.1109/TPAMI.2009.187>
- [20] Hyndman, R.J. and G. Athanasopoulos, *Forecasting: principles and practice*. 2018: OTexts.
- [21] Hajihassani, M., D.J. Armaghani, H. Sohaei, E.T. Mohamad, and A. Marto, Prediction of airblast-overpressure induced by blasting using a hybrid artificial neural network and particle swarm optimization. Applied Acoustics, 2014. 80: p. 57-67. <https://doi.org/10.1016/j.apacoust.2014.01.005>
- [22] Ruder, S., An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747, 2016.