

# **Forecasting Ammonia Concentrations and Colour Levels using Machine Learning for Reclaimed Water Treatment Operation and Management**

by

**Ting Hsi LEE**

A Thesis Submitted to  
The Hong Kong University of Science and Technology  
in Partial Fulfillment of the Requirements for  
the Degree of Master of Philosophy  
in the Department of Civil and Environmental Engineering

August 2022, Hong Kong

## **Authorization**

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

---

Ting Hsi LEE

August 2022

# **Forecasting Ammonia Concentrations and Colour Levels using Machine Learning for Reclaimed Water Treatment Operation and Management**

by

Ting Hsi LEE

This is to certify that I have examined the above MPhil thesis  
and have found that it is complete and satisfactory in all respects,  
and that any and all revisions required by  
the thesis examination committee have been made.

---

Prof. Chii SHANG, Thesis Supervisor

---

Prof. Limin ZHANG, Head of Department

Department of Civil and Environmental Engineering  
August 2022

## Acknowledgements

I would first express my enormous gratitude to my thesis supervisor Prof. Chii Shang for giving me the opportunity to start my MPhil degree in this research group. I was given chances to be exposed to different research topics and work with other brilliant students; most importantly, I have learned so much from his mentoring in research and life. He also encouraged me to develop the research area I am interested in. I sincerely appreciate his guidance in the last two years.

I appreciate Prof. Guanghao CHEN and Prof. Yuhsing WANG for serving on my thesis examination committee. Special thanks should go to Dr. Ran YIN for giving critical advice on my research. Dr. YIN has guided me throughout my master's degree and led me to work on a new research area.

I thank my fellow labmates in Prof. SHANG's group: Oriana JOVANOVIC, Gabriela CAS-SOL, Jing ZHAO, and Tao LI, for their support and help. Also, I thank my friends in HKUST: Yude PEI, Yugo SATO, and Kaleong CHENG for their additional help with my research.

Lastly, I would like to thank my family for their love, support, and continuous encouragement.

# TABLE OF CONTENTS

<b>Title Page</b> . . . . .	i
<b>Authorization Page</b> . . . . .	ii
<b>Signature Page</b> . . . . .	iii
<b>Acknowledgements</b> . . . . .	iv
<b>Table of Contents</b> . . . . .	v
<b>List of Figures</b> . . . . .	vii
<b>List of Tables</b> . . . . .	ix
<b>Abstract</b> . . . . .	x
<b>Chapter 1 Methods and Materials</b> . . . . .	1
1.1 Wastewater treatment plant description . . . . .	1
1.1.1 Wastewater treatment process in SWHEPP . . . . .	1
1.2 Data collection and preparation . . . . .	2
1.2.1 On-line data monitoring and collection . . . . .	2
1.2.2 Loss function for model evaluation . . . . .	5
1.2.3 Data cleaning and pre-processing . . . . .	6
1.2.3.1 Data smoothing with Savitzky-Golay and EWMA filter . . . . .	8
1.2.3.2 Outlier Removal . . . . .	10
1.2.3.3 Feature Engineering . . . . .	10
1.2.4 Data transformation . . . . .	17
1.2.5 Feature selection . . . . .	19
1.3 Machine learning models . . . . .	20
1.3.1 Random Forest . . . . .	20
1.3.2 Deep Neural Network . . . . .	20
1.3.3 Recurrent Neural Network . . . . .	21
1.3.4 Long Short-Term Memory . . . . .	21
1.3.5 Gated Recurrent Unit . . . . .	24
1.3.6 Configurations of machine learning models . . . . .	25
<b>Chapter 2 Results and Discussion</b> . . . . .	28
2.1 Baseline performance of the forecasting models . . . . .	28
2.2 Improved performance on forecasting models using data pre-processing techniques . . . . .	33
2.2.1 Models trained by pre-processed datasets . . . . .	33
2.2.2 The effects of window sizes of the data smoothing filters . . . . .	39
2.3 Exploit hidden patterns in the MBR effluent quality to enhance model performance	41
2.3.1 Ammonia forecasting models . . . . .	41

2.3.2	Colour forecasting models . . . . .	42
2.3.3	Model forecasting results on different forecast horizons . . . . .	43
<b>Chapter 3</b>	<b>Conclusions and Recommendations</b> . . . . .	<b>52</b>
3.1	Conclusions . . . . .	52
3.1.1	Machine learning models versus deep learning models . . . . .	52
3.1.2	Data pre-processing techniques . . . . .	52
3.1.3	Feature engineering techniques . . . . .	53
3.2	Recommendations for future research . . . . .	54
<b>References</b>	. . . . .	<b>56</b>
<b>Appendix A</b>	<b>Python codes</b> . . . . .	<b>58</b>
A.1	Python codes for machine learning models . . . . .	58

## LIST OF FIGURES

1.1	Sewage treatment process flowchart at SWHEPP (adapted from Drainage Services Department 2020) . . . . .	1
1.2	Colour levels and ammonia concentrations were measured in the effluent container (i.e., on the right of the image.) A water pump transported MBR effluent to the effluent container continuously in real-time. The black vault on the left of the image contained a laptop and a colour spectrophotometer. . . . .	2
1.3	instruments of on-line ammonium monitoring system. . . . .	3
1.4	Instruments of on-line colour analysis system. . . . .	4
1.5	The dates of manually calibration and colour level measured in the laboratory were plotted as blue crosses and red dots. . . . .	5
1.6	Schematic diagram of the custom-made on-line colour analysis system. . . . .	6
1.7	Ammonia and colour data collected from 23 December 2021 to 22 January 2022.	6
1.8	Training steps of the machine learning models. . . . .	7
1.9	Illustration of the influence of different polynomial degrees in the fitting of SG filter and the weight decay with varied alpha values in EWMA filter. . . . .	9
1.10	Comparisons of the applying different window sizes on ammonia concentration datasets. . . . .	11
1.11	Comparisons of the applying different window sizes on colour level datasets. . .	12
1.12	Illustration of peak analysis. Four important elements were automatic calculated by the function (MathWorks, 2022). . . . .	13
1.13	Sewer system coverage of SHWEPP. The covered areas (i.e., area circled in red boundary) include Fanling/Sheung-Shui new towns and NENT landfill leachate treatment plant. . . . .	13
1.14	Analysis of influent quality composition and the illustration of the positional encoding. . . . .	14
1.15	Observed ammonia concentrations and colour levels in SHWEPP influent. . . .	15
1.16	Hourly water consumption patterns in households (Abu-Bakar et al., 2021). (a) Cumulative pattern and percentage of hourly consumption for households in the “Evening Peak (EP)” cluster (b) Cumulative pattern and percentage of hourly consumption for households in the “Late Morning Peak Peak (LM)” cluster. (c) Cumulative pattern and percentage of hourly consumption for households in the “Early Morning Peak (EM)” cluster. (d) Cumulative pattern and percentage of hourly consumption for households in the “Multiple Peak (MP)” cluster. Consumption is in ( $m^3$ ). . . . .	16
1.17	The daily patterns of ammonia concentrations on 3, 7, 11, and 15 January 2022.	16
1.18	Concept of forecasting models (Liu, 2020). . . . .	18
1.19	Illustration of the training and inference process. . . . .	19
1.20	Illustration of feature selections for model training. . . . .	19
1.21	Illustration of RF and DNN model structure. . . . .	22

1.22 Variant architectures of Recurrent Neural Networks (adapted from Olah (2015)). $x_t$ corresponds to the current input, $h_{t-1}$ to the last hidden state (output), $h_t$ to the current output, $\tanh$ is the tangent activation function, $\sigma$ is the sigmoid activation function, $\times$ is the vector pointwise multiplication, $+$ is the vector pointwise addition. . . . .	23
1.23 Illustration of how different step sizes of learning rate reach the minimum loss (Ritchie Ng, 2019). . . . .	26
2.1 Baseline performance of the ammonia and colour forecasting models. . . . .	29
2.2 Visualization of the baseline ammonia forecasting results. . . . .	31
2.3 Visualization of the baseline colour forecasting results. . . . .	32
2.4 Results of the removed outliers from the training dataset. . . . .	35
2.5 Illustration of the heterogeneity and homogeneity between validation and different testing datasets. . . . .	36
2.6 Baseline performance of the ammonia and colour forecasting models. . . . .	40
2.7 Comparisons of the model performance in forecasting ammonia concentrations. . . . .	42
2.8 Comparisons of model performance in forecasting colour levels. . . . .	44
2.9 Visualization of the ammonia forecasting models at forecast horizon of one. . . . .	45
2.10 Visualization of the ammonia forecasting models at forecast horizon of two. . . . .	46
2.11 Visualization of the ammonia forecasting models at forecast horizon of three. . . . .	47
2.12 Visualization of the colour forecasting models at forecast horizon of one. . . . .	48
2.13 Visualization of the colour forecasting models at forecast horizon of two. . . . .	49
2.14 Visualization of the colour forecasting models at forecast horizon of three. . . . .	50

## LIST OF TABLES

1.1	The selected hyperparameters for SG and EWMA filters. . . . .	10
1.2	Final model configurations. . . . .	27
2.1	Baseline performance of the ammonia forecasting model, evaluated on test dataset from <b>16 to 22 January 2022</b> . Loss values were calculated by MSE. . . . .	33
2.2	Baseline performance of the ammonia forecasting models, evaluated on test dataset from <b>10 to 16 October 2021</b> . Loss values were calculated by MSE. . . .	37
2.3	Baseline performance of the colour forecasting models, evaluated on test dataset from <b>16 to 22 January 2022</b> . Loss values were calculated by MSE. . . . .	38

# **Forecasting Ammonia Concentrations and Colour Levels using Machine Learning for Reclaimed Water Treatment Operation and Management**

by Ting Hsi LEE

Department of Civil and Environmental Engineering

The Hong Kong University of Science and Technology

## **Abstract**

Water scarcity is a global challenge, and one of the promising ways to mitigate the water resource crisis is via wastewater reclamation. Reclaimed water can generate non-potable water to substitute the use of drinking water for irrigation or industrial processes. Water quality and aesthetics are the primary concerns in reclaimed water since undertreated water can pose health risks, and the unpleasant colour is likely to induce public misgiving. Ammoniacal nitrogen ( $\text{NH}_3\text{-N}$ ) and colour substances exist in the reclaimed water and can severely affect the reclaimed water quality in different ways. Chlorine is commonly used for reclaimed water disinfection and requires precise dosing to satisfy endorsed quality standards. However,  $\text{NH}_3\text{-N}$  consumes chlorine and affects the chlorine dosing. Colour substances do not consume chlorine, but it requires additional efforts and strategies to remove them from the reclaimed water. Therefore, the on-line monitoring of  $\text{NH}_3\text{-N}$  concentrations and colour levels are usually practised in reclaimed water facilities to assist in the removal of both substances. However, the conventional on-line analyzers are wet-chemistry-based, and the measurement takes time. The limitation creates a potential issue: there may not be sufficient time for the downstream chlorine dosing system to respond to sudden surges in colour and  $\text{NH}_3\text{-N}$  levels. To tackle this challenge, this thesis work developed time-series models based on machine learning to forecast the  $\text{NH}_3\text{-N}$  concentrations and colour levels in the reclaimed water three hours into the future. For the training dataset, the  $\text{NH}_3\text{-N}$  and colour data were collected by an on-line analyzer and a customized auto-sampling spectrophotometer, respectively. Both are installed in a

reclaimed water treatment facility in Hong Kong. Baseline models for forecasting NH<sub>3</sub>-N concentrations and colour levels were first developed with five machine learning algorithms. Long Short-Term Memory (LSTM) was found to be the most effective algorithm, with the lowest MSE values of 0.0405 and 0.0148 for NH<sub>3</sub>-N and colour forecasting models, respectively. In the training processes, novel data pre-processing methods and feature engineering techniques were implemented to enhance forecasting model performance. The data pre-processing methods were proved to enhance the quality of training datasets and improve the performance of NH<sub>3</sub>-N and colour forecasting models by reducing the MSE values by 4.2% and 8.1%. The feature engineering results supported that the daily fluctuations in NH<sub>3</sub>-N and colour have correlations with the urban water consumption patterns. This finding further enhanced the NH<sub>3</sub>-N and colour forecasting model performance by reducing MSE by 8.9% and 28.6% compared to baseline models. The established models can be used to assist the disinfection control strategies based on the model predictions using traditional process control systems. This research offers novel methods and feature engineering techniques for NH<sub>3</sub>-N concentrations and colour levels forecasting in reclaimed water for treatment optimization.

# CHAPTER 1

## METHODS AND MATERIALS

### 1.1 Wastewater treatment plant description

#### 1.1.1 Wastewater treatment process in SWHEPP

Shek Wu Hui Effluent Polish Plant (SWHEPP) is a secondary sewage treatment plant that treats the municipal wastewater from Sheung Shui/Fanling Districts and the treated leachate effluent from North East New Territories (NENT) leachate treatment plant. The plant was designed for 300,000 population equivalents (PE) in 2001, and in 2009, the daily treatment capacity was expanded from 80,000 m<sup>3</sup>/day to 93,000 m<sup>3</sup>/day. SHWEPP is operated and maintained by Drainage Services Department (DSD), and the plant will be upgraded to a tertiary treatment level to increase the treatment capacity of 190,000 m<sup>3</sup>/day by the end of 2025. As shown in Fig. 1.1, the treatment plant consists of primary sedimentation, secondary biological treatment, and final sedimentation, followed by a membrane bioreactor (MBR), which provides an advanced level of organic and suspended solids removal. A low volume of the MBR effluent was pumped to an effluent container n the MBR location to monitor the effluent quality in real-time. An ammoniacal nitrogen on-line sensor and a colour level on-line analyzer are installed in the effluent container, indicated as (a) and (b) in Fig. 1.1.

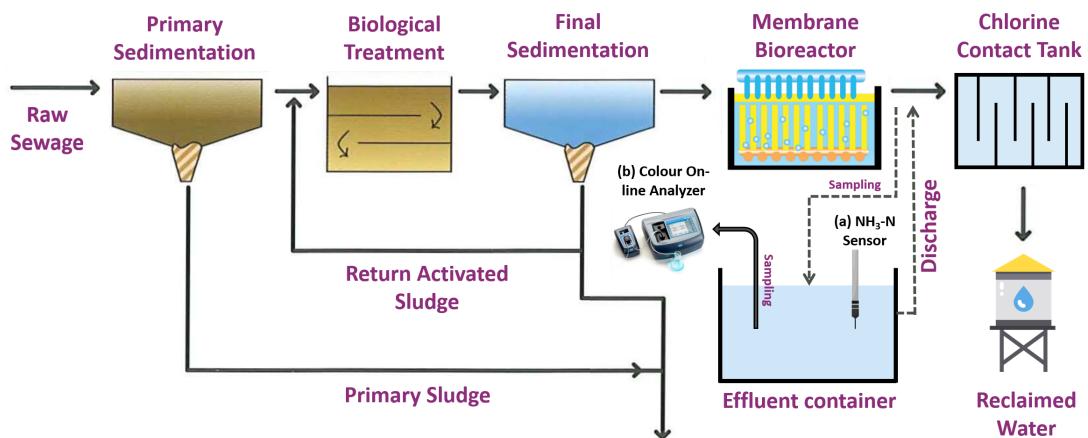


Figure 1.1: Sewage treatment process flowchart at SWHEPP (adapted from Drainage Services Department 2020)

## 1.2 Data collection and preparation



Figure 1.2: Colour levels and ammonia concentrations were measured in the effluent container (i.e., on the right of the image.) A water pump transported MBR effluent to the effluent container continuously in real-time. The black vault on the left of the image contained a laptop and a colour spectrophotometer.

### 1.2.1 On-line data monitoring and collection

To enable us to perform on-line monitoring of ammonium concentrations ( $\text{NH}_3\text{-N}$ ) in the MBR effluent, an Ammonium and Potassium Probe, AmmoLyt® Plus 700 IQ (Xylem Company) was installed as Fig. 1.3a in the effluent container, as shown in Fig. 1.2. The operation was commenced on 27 April 2021 and completed on 27 March 2022. The ion-selective electrode (ISE) probe provides continuous and reagentless monitoring of ammonium and potassium at the configured interval of one measurement per minute. Due to the ISE probe cannot differentiate the potential difference caused by ammonium and potassium ions in the electrodes, the on-line monitoring of ammonium concentrations requires continuous calibration using potassium concentrations.

The instrument records ammonium concentration as  $\text{NH}_4\text{-N mg/L}$ , a form to express the sum of nitrogen found in reduced nitrogen (III) form. Ammonia has a reported pKa of 9.25 (National Center for Biotechnology Information, 2022), meaning ammonium is a primary species under

the pH of 9.25 in water. In WWTPs, the pH in water typically ranges from pH of 7–8, making the NH<sub>4</sub>-N concentrations the dominant species. Both ammonia and ammonium contain one nitrogen atom; 1 mg/L NH<sub>3</sub>-N is the same as 1 mg/L NH<sub>4</sub>-N. Thus, to prevent confusion, in the following paragraph, the unit of NH<sub>4</sub>-N will be expressed by NH<sub>3</sub>-N, which is the unit used in the water quality standard. The collection of on-line ammonia data was achieved through downloading CSV files from the website connected to the IQ Sensor Controller (Xylem Company), as shown in Fig. 1.3b.



(a) AmmoLyt®Plus 700 IQ,  
Xylem.

(b) DIQ/S 284-EF controller,  
Xylem.

Figure 1.3: instruments of on-line ammonium monitoring system.

Hourly monitoring of the colour levels of MBR effluent was conducted from 5 October 2021 to 26 February 2022 by using a custom-made on-line colour analyzer. The default spectrophotometer as Fig. 1.4a and a peristaltic pump as Fig. 1.4b is only capable of initiating a single measurement of colour level by pressing the "READ" button on the DR3900 panel. To achieve continuous sampling and analyzing colour levels without human intervention, an actuator with a programmable time function was mounted on the panel of DR3900, as shown in Fig. 1.4c.

The automatic sampling and analyzing of the colour level begins with the actuator clicking on the "READ" button to initiate the colour analysis at a fixed interval of 30 minutes. 3 mL of sample was collected from the effluent container and delivered to the spectrophotometer cell. After the spectrophotometer analyzed the sample, the data was transmitted to an automatic data acquisition and storage software pre-installed on the laptop. The DR3900 device was connected to a laptop, which receives the real-time data and stores it on data management software from Hach company. To access the real-time data from the laptop, Google Remote Desktop was used to operate the laptop via Internet cloud services using any devices having access to the Internet. The entire process is illustrated in Fig. 1.6. After the measurement, the sample will be

discharged to the effluent container, and the on-line colour monitoring system is left idle until the subsequent measurement.



(a) SIP10 peristaltic pump,  
Hach Company

(b) DR3900 spectrophotome-  
ter, Hach Company

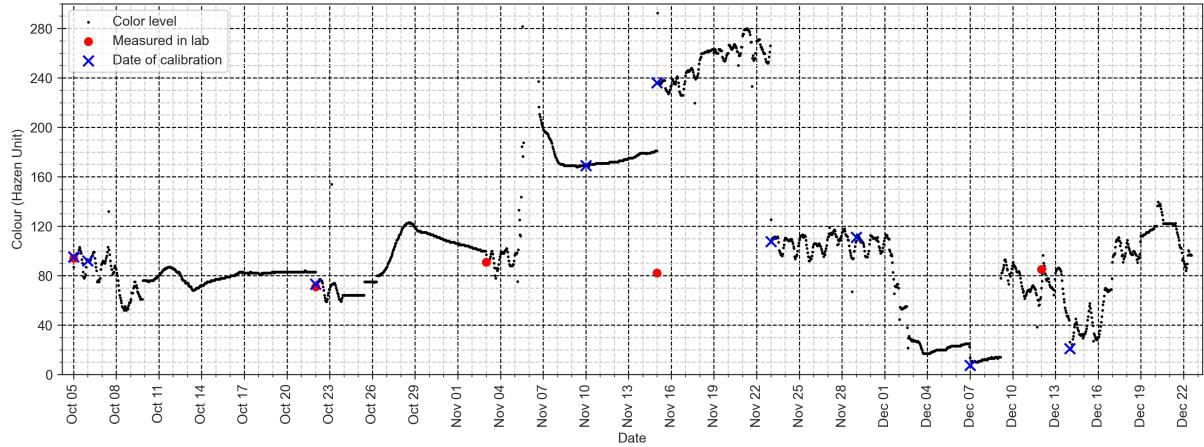


(c) Customized clicker/actuator with programmable timer

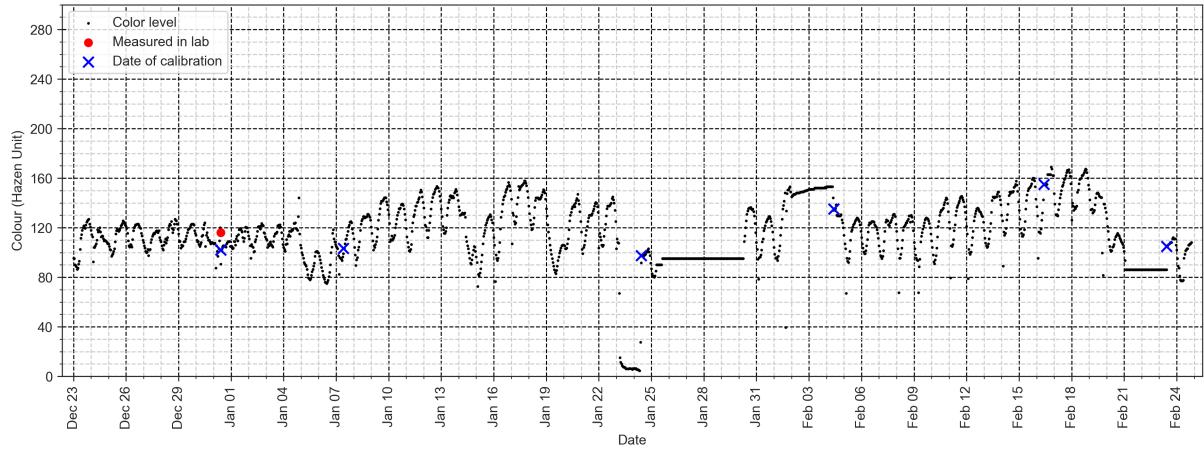
Figure 1.4: Instruments of on-line colour analysis system.

As shown in Fig. 1.5, the maintenance and calibration of the DR3900 spectrophotometer are performed on a weekly basis. During the maintenance, the DR3900 device was shut off, and 100 mg/L chlorine solution was pumped into the sampling tubes and the plastic cuvette for disinfection and cleansing. The cleansing of the tubes and cuvette were manually inspected with eyes to make sure no foreign objects were stuck inside. De-ionized water was brought to the site to perform the spectrophotometer calibration after the reboot of DR3900.

In the proposed model training methods, ammonia and colour data are input into the training forecasting models. Thus, the colour and ammonia data as features should be collected from the same period of time with the same dataset size. In addition, abnormal data caused by sensor downtime should also be excluded. Thus, we chose the ammonia and colour data from 23 December 2021 to 22 January, as shown in Fig. 1.7.



(a) Data collected from 5 October 2021 to 22 December 2021.



(b) Data collected from 23 December 2021 to 24 February 2022.

Figure 1.5: The dates of manually calibration and colour level measured in the laboratory were plotted as blue crosses and red dots.

### 1.2.2 Loss function for model evaluation

Loss functions are used to determine the error between the model outputs (i.e., prediction or forecasting values) and the given target value (DeepAI, 2022). The bigger the difference between the ground truth  $y$  and the model outputs  $\hat{y}$ , the higher the value of the loss function is, meaning the model performed poorer. A low value for the loss means the model performed well. The selection of the types of loss function is essential for training the model to perform specific tasks. This study uses Mean Squared Error (MSE) to evaluate the regression models. The values of MSE will never be negative and are formally defined by the following equation:

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n} \quad (1.2.1)$$

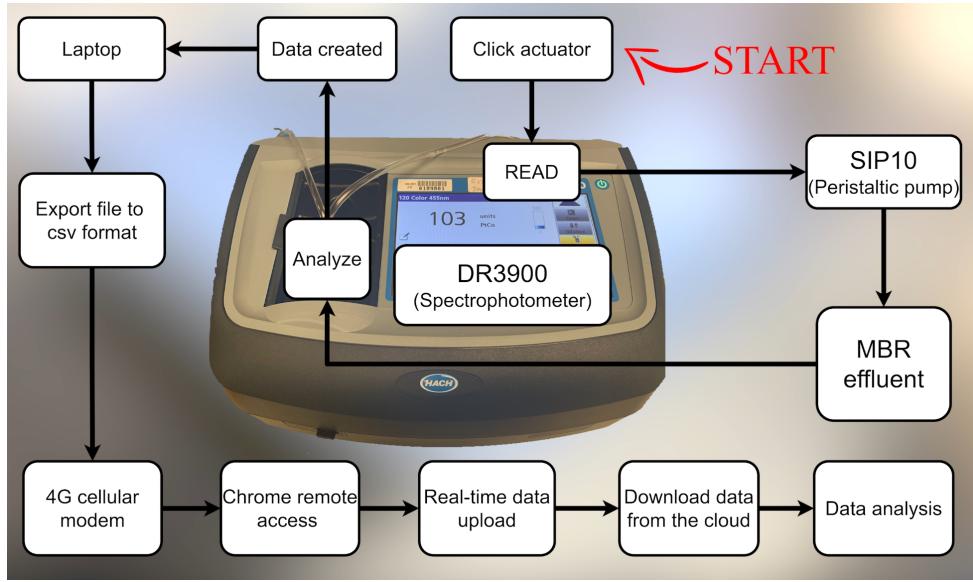


Figure 1.6: Schematic diagram of the custom-made on-line colour analysis system.

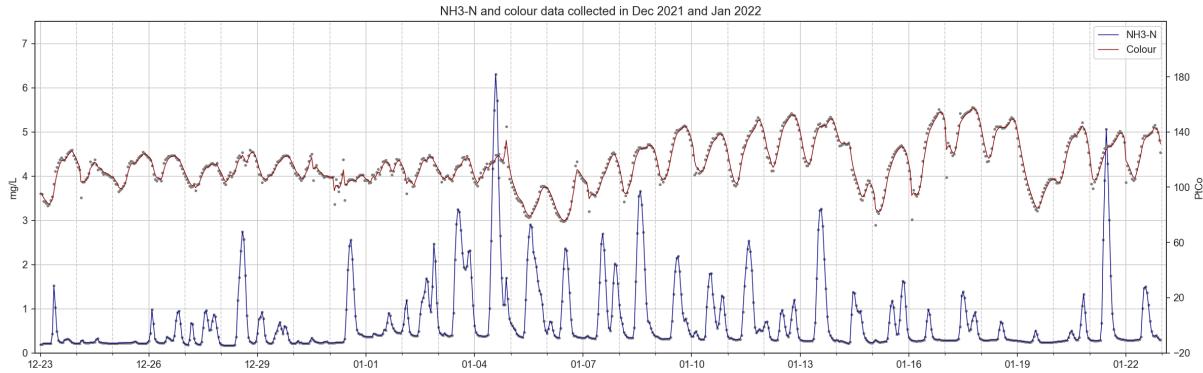


Figure 1.7: Ammonia and colour data collected from 23 December 2021 to 22 January 2022.

### 1.2.3 Data cleaning and pre-processing

In this study, ammonia concentrations and colour levels forecasting models will be trained, and the model training steps are shown in Fig. 1.8. The training processes are split into two sections; one is the baseline model training steps, and the other is the proposed model training steps. The training steps of the first section used cleaned data to train forecasting models and generated baseline model performance, which will be further compared with the model performance generated in the second section. The second section includes using pre-processed datasets (i.e., data smoothing) and feature engineering enhanced datasets to train the forecasting model.

The raw data embedded in the original CSV files has many problems, such as missing values, extreme low or high values, unreadable texts, etc. Thus, data cleaning and pre-processing

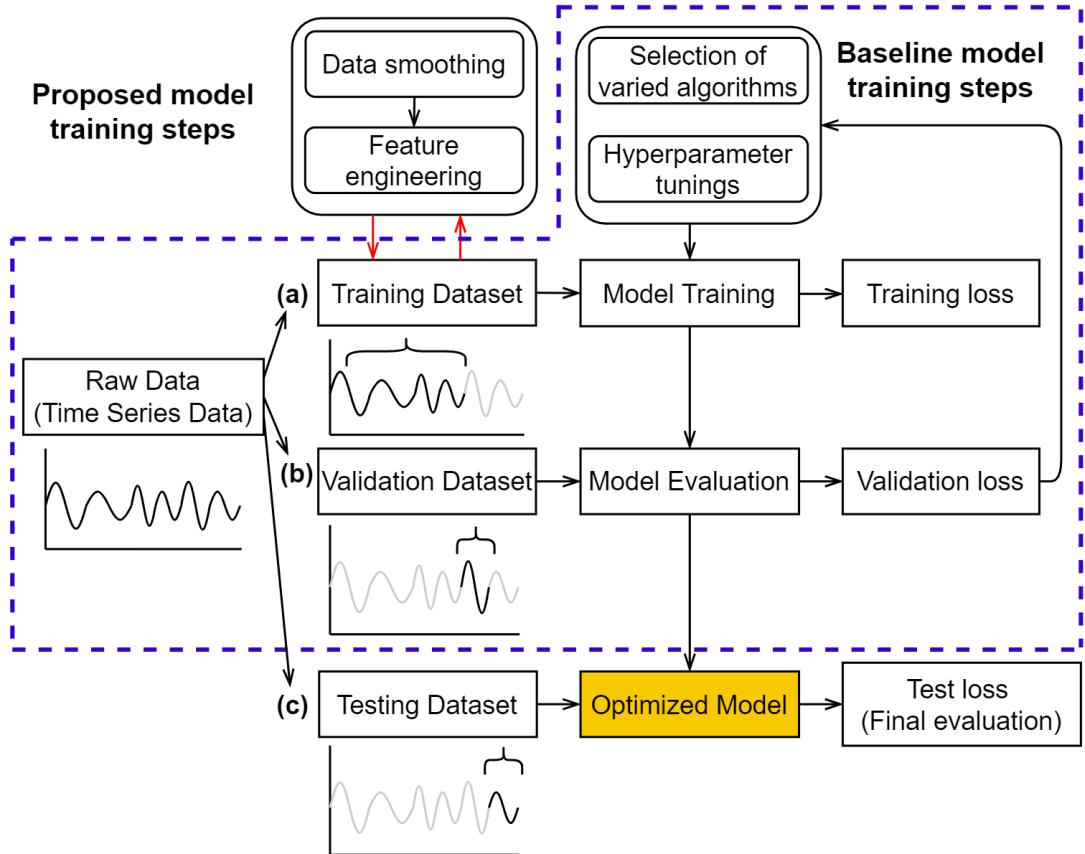


Figure 1.8: Training steps of the machine learning models.

are necessary for a more effective model training process. Python programming language and related libraries such as Numpy and Pandas were used to clean and pre-process the raw dataset for further usage. The raw ammonia dataset collected from the instrument contained 44,640 samples (data points) with eight variables, giving a matrix size of 44,640 x 8, and the samples were collected in time series at 1-minute intervals. The colour level raw dataset collected from the laptop contained 1488 samples with 34 variables, giving a matrix size of 1488 x 34, and the samples were collected in time series at 30-minute intervals.

Extreme values were manually removed before the colour and ammonia datasets were averaged into time-series data at 1-hour intervals. For the ammonia dataset, we replaced the values higher than 7.0 mg/L with NaN (i.e., Not a number), and further interpolation was used to fill up the NaN along with the missing values in the dataset. For colour dataset, we manually took out the relatively low data points on the days when the maintenance and calibration tasks were performed; extremely values higher than 300 Hazen Unit were also replaced by NaN. Same as the data cleaning method used for the ammonia dataset, the missing values and NaN were filled up with interpolation.

### 1.2.3.1 Data smoothing with Savitzky-Golay and EWMA filter

Data smoothing was performed using the same methods on ammonia concentrations and colour levels datasets. One of the effective ways to remove the noise from the dataset is to apply data smoothing filters. Two filters were applied in this study, Savitzky-Golay (SG) and Exponentially Weighted Moving Average (EWMA) filters.

An SG filter is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data without distorting the data tendency. This is achieved via convolution by fitting successive subsets of adjacent data points with a low-degree polynomial using linear least-squares (Wikipedia, 2022b). The illustration is shown in Fig. 1.9a, and the procedures of how data points are smoothed are presented in the following steps:

- 1) Extract short-time window (i.e., blue dots in Fig.1.9a)
- 2) Determine polynomial degree (e.g., different polynomial degree is compared in Fig. 1.9a).
- 3) Find the smoothed data point (i.e., at center of the window).
- 4) Repeat for shifted window (e.g., similar to moving average).

The equation to describe the smoothed value of  $Y_j$  can be expressed in Eq. 1.2.2:

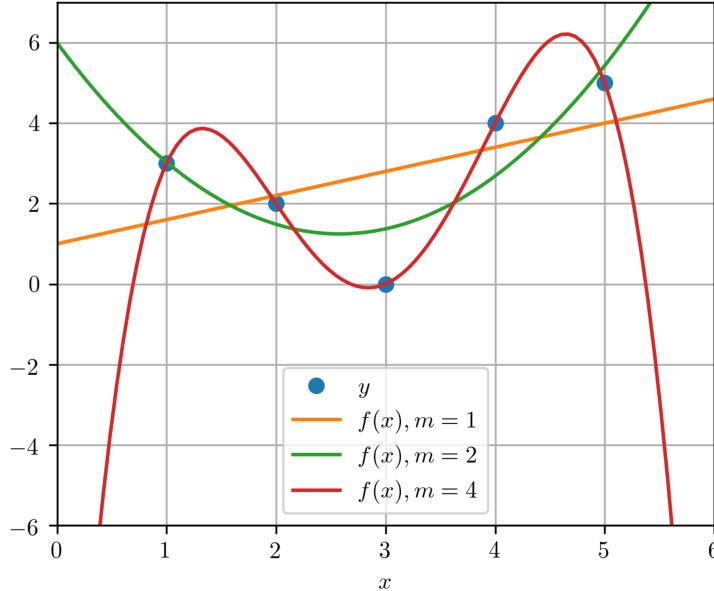
$$Y_j = (C \otimes y)_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i y_{j+i}, \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \quad (1.2.2)$$

where  $Y_j$  corresponds to the  $j^{th}$  smoothed data point,  $m$  to the window size (i.e., numer of data points intended to smooth out) and  $C_i$  to the convolution coefficients (i.e., determined by Savitzky and Golay (1964)).

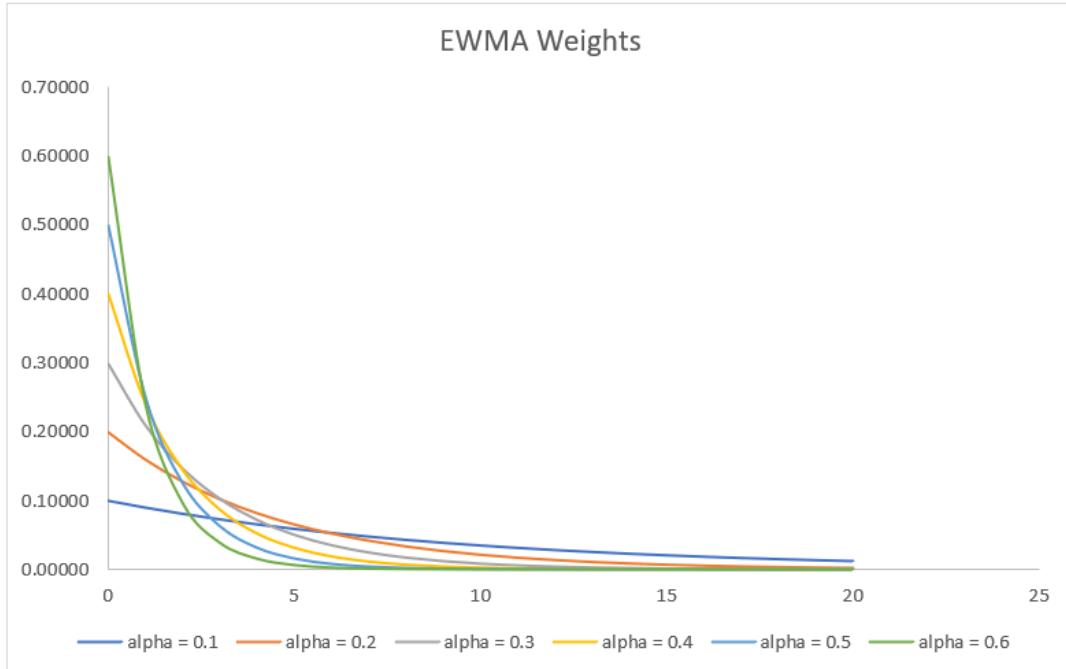
Exponentially weighted moving average (EWMA), also known as autoregressive (AR) filtering, is a technique that filters measurements. An EWMA filter smoothes a measured data point by exponentially averaging that particular point with all previous measurements. The EWMA equation can be expressed in Eq. 1.2.3:

$$\begin{aligned} \alpha &= \frac{2}{span + 1} \\ y_0 &= x_0 \\ y_t &= (1 - \alpha)y_{t-1} + \alpha x_t \end{aligned} \quad (1.2.3)$$

where  $\alpha$  corresponds to the decay parameters,  $x_t$  to the value at a time period,  $y_t$  to the value of the EWMA at any time period t, span to the window size.



(a) SG filter with different polynomial degree (Taal, 2017).



(b) Examples of weights with exponential decay at varied alpha values (CFI, 2022).

Figure 1.9: Illustration of the influence of different polynomial degrees in the fitting of SG filter and the weight decay with varied alpha values in EWMA filter.

Both SG and EWMA filters are required to select the hyperparameters, the selected values are presented in Table. 1.1.

Table 1.1: The selected hyperparameters for SG and EWMA filters.

Group Name	Window size	Polynomial degree
SG-5	5	2
SG-7	7	2
SG-9	9	2
EWMA-2	2	-
EWMA-3	3	-
EWMA-4	4	-

Fig. 1.10 and Fig. 1.11 show the influences of different windows sizes of SG and EWMA filters on ammonia concentrations and colour levels datasets.

### 1.2.3.2 Outlier Removal

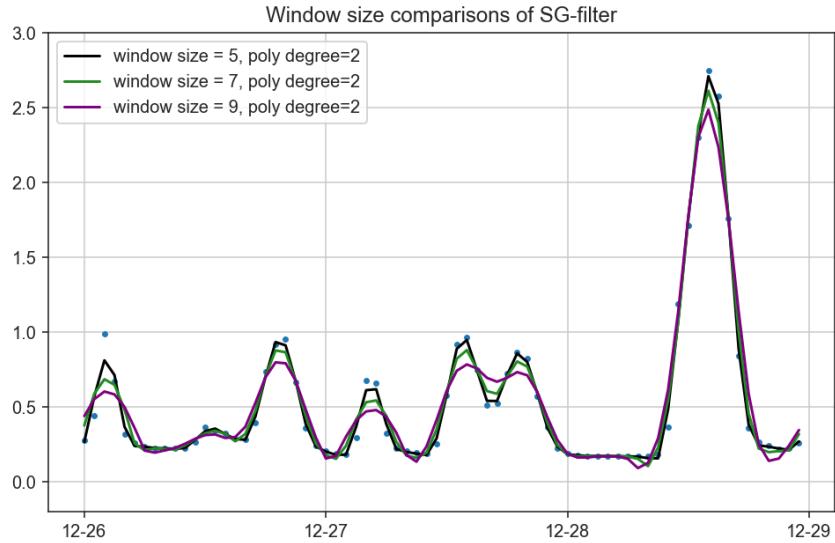
Although the extreme values in the raw ammonia dataset were removed based on basic rules (i.e., concentrations higher than 7.0 mg/L), the ammonia sensor can still collectively capture unideal data points. In the outlier removal process, we intended to identify the collective faults of ammonia data in the unit of an entire day. Two abnormal conditions were defined to determine whether the ammonia data on a specific day shows collective fault:

- 1) NH<sub>3</sub>-N fluctuation  $\leq 0.1$  (i.e., lower than the sensor resolution).
- 2) No diurnal fluctuation (i.e., Fluctuation = peak value – bottom line value).

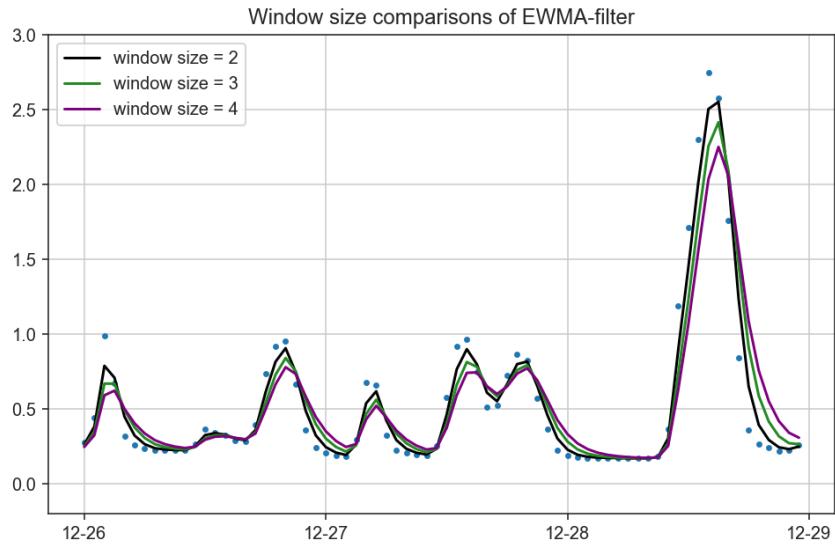
Peak analysis was performed on the daily ammonia data to automatically identify normal or abnormal signals. The analysis takes a one-dimension array (i.e., the data form of ammonia in a day) and finds all local maximum values by comparing neighbouring values. This function will also provide information such as width and prominence, as in Fig. 1.12 to help us identify whether the diurnal fluctuation exists.

### 1.2.3.3 Feature Engineering

We have carefully observed and analyzed the SWHEPP influent to create new features from the raw datasets based on our domain knowledge. We discovered that the SWHEPP influent consists of treated landfill effluent from NENT landfill leachate site and municipal wastewater, as shown in Fig. 1.13. We observed that with a higher blending ratio, which was calculated from the daily volume of treated leachate effluent divided by the daily inflow volume of SHWEPP,



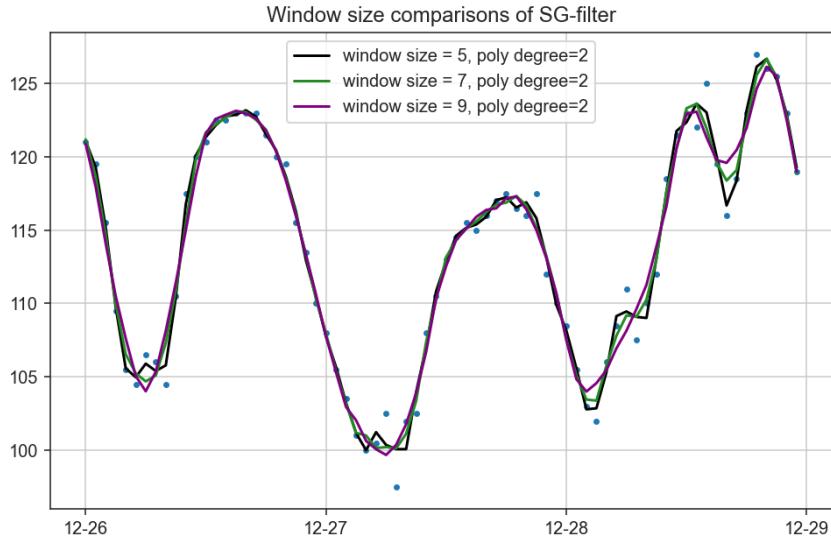
(a) Ammonia data filtered by SG filters with different window sizes.



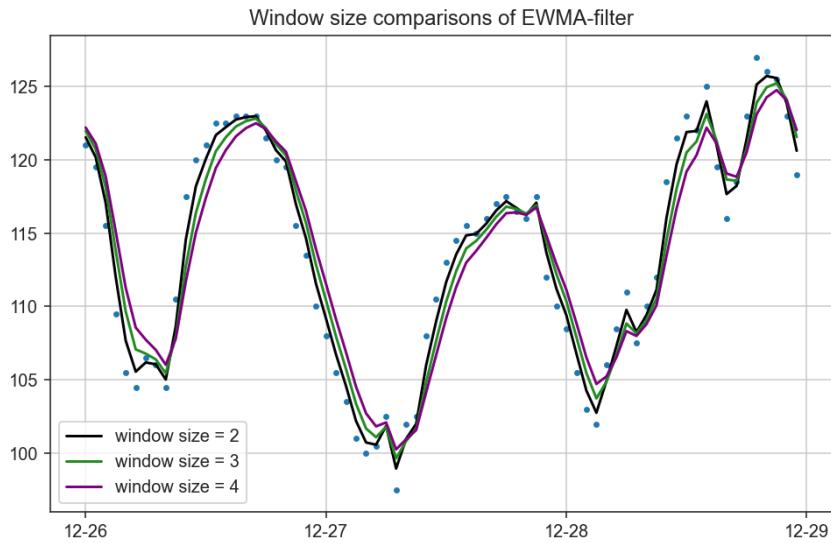
(b) Ammonia data filtered by EWMA filters with different window sizes.

Figure 1.10: Comparisons of the applying different window sizes on ammonia concentration datasets.

the colour level were also higher, as shown in Fig 1.15a. With the Pearson coefficient of 0.68, the increased volume of treated leachate effluent in the public sewage system was proportional to the increase of the colour levels in the SHWEPP influent, while the ammonia concentrations was mainly from the municipal wastewater. During the mixing of both types of wastewater, as in Fig. 1.14a, substances contributing to colour levels were diluted by the municipal wastewater; at the same time, the ammonia concentrations was also diluted by the treated leachate effluent. In Fig. 1.15b, we can observe that the time when the lowest colour level of the day occurred was close to when the highest ammonia concentration was observed. The changes in colour



(a) Colour data filtered by SG filters with different window sizes.



(b) Colour data filtered by EWMA filters with different window sizes.

Figure 1.11: Comparisons of the applying different window sizes on colour level datasets.

levels and ammonia concentrations were interactive. Thus, in feature engineering, colour level data was selected for training the ammonia forecasting model; ammonia data was selected for the training colour forecasting model, as shown in Fig. 1.20.

The new features were inspired by the research work of Abu-Bakar et al. (2021). The author summarized the four types of hourly household water consumption patterns as in Fig. 1.16, which correlates the specific time of the day to the volume of water consumed in households. In other words, as fresh water is consumed, wastewater is generated simultaneously; the wastewater then enters the public sewage system and increases ammonia concentrations. As shown in Fig. 1.17, the peak analysis tool helped us to identify the ammonia concentrations' peak hour,

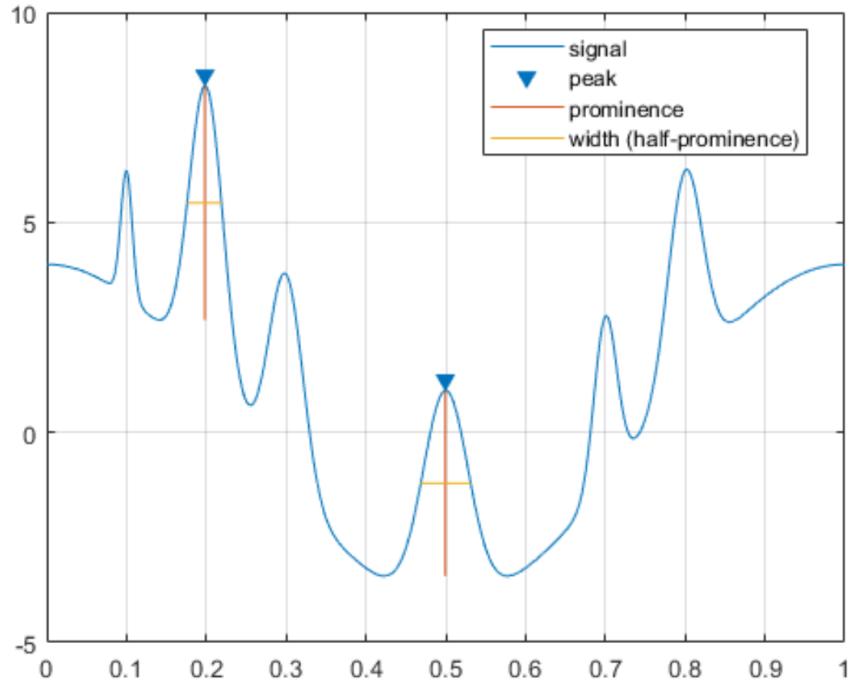


Figure 1.12: Illustration of peak analysis. Four important elements were automatically calculated by the function (MathWorks, 2022).

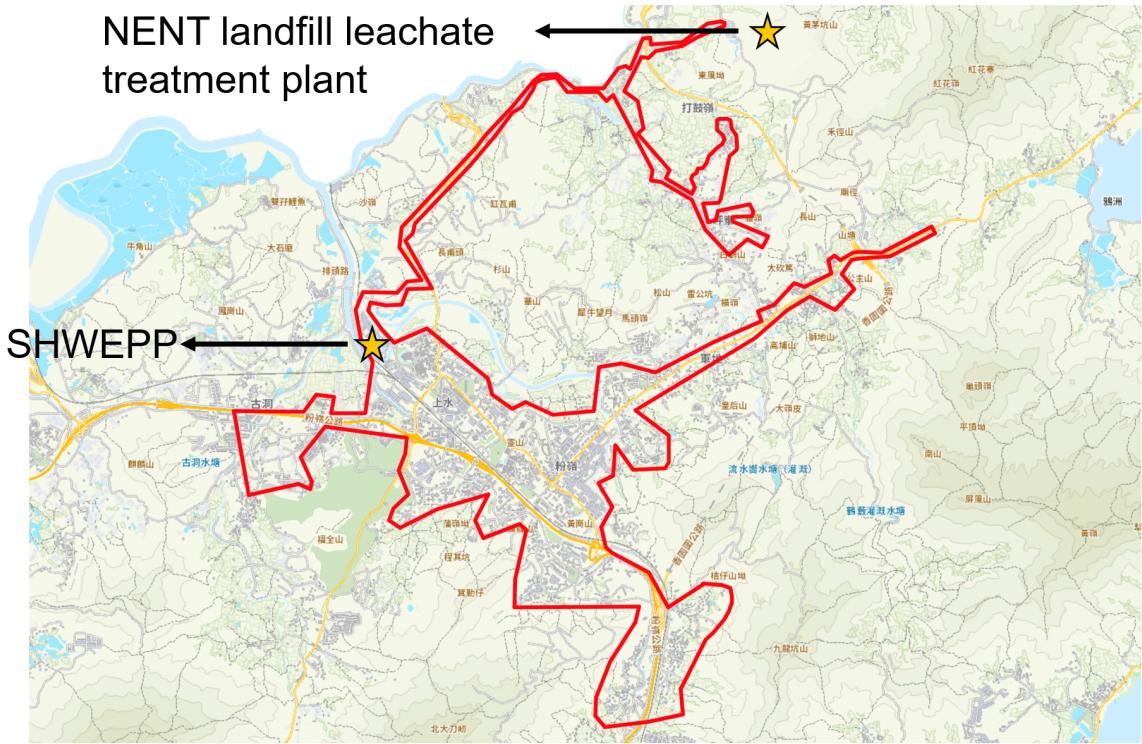
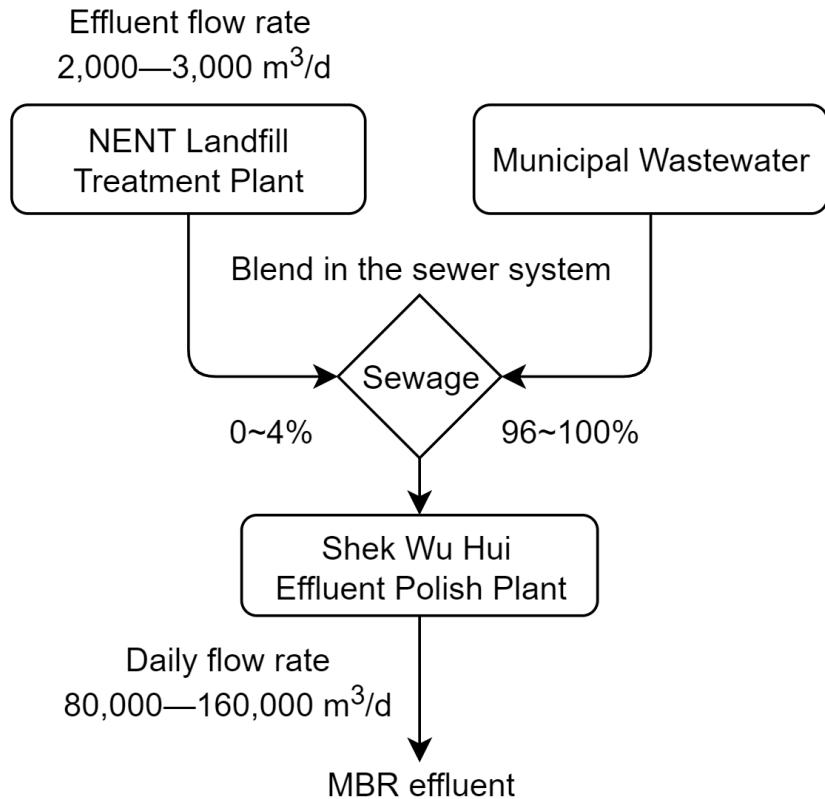
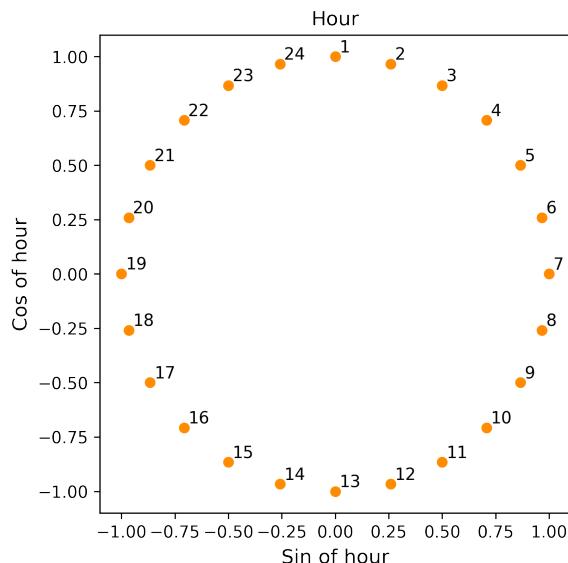


Figure 1.13: Sewer system coverage of SHWEPP. The covered areas (i.e., area circled in red boundary) include Fanling/Sheung-Shui new towns and NENT landfill leachate treatment plant.

which occurred around 13:00 to 14:00, and 20:00 to 21:00. Thus, it is convinced that time features will help the machine learning models correlate better and predict the change of am-



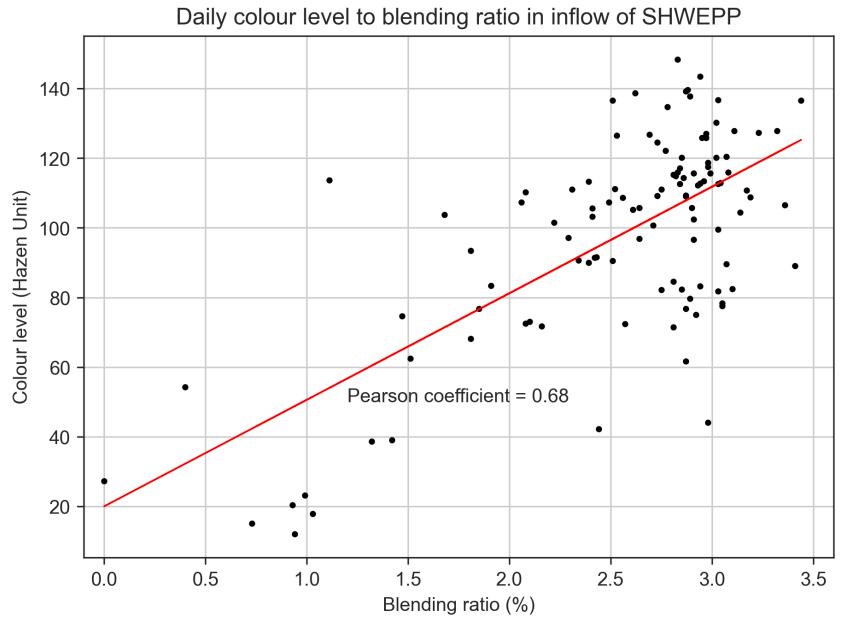
(a) Flowchart showing the blending of treated leachate effluent with municipal wastewater.



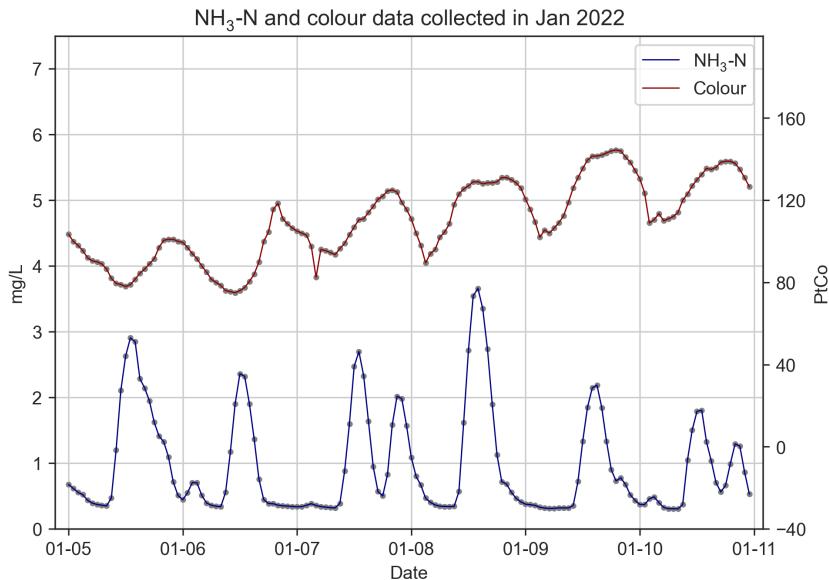
(b) Positional encoding of hour components.

Figure 1.14: Analysis of influent quality composition and the illustration of the positional encoding.

monia concentrations in the wastewater. Time feature was created through a technique called positional encoding (POS). The positional encoded feature was achieved in the following steps:



(a) Coefficient between blending ratio and colour levels.



(b) Trend comparison of ammonia concentrations and colour levels.

Figure 1.15: Observed ammonia concentrations and colour levels in SHWEPP influent.

- 1) The timestamp is represented as three elements—hour, day and month.
- 2) Each element will be decomposed into sine and cosine components.
- 3) Last step is applied to hours and days to make all elements represented cyclically.

Due to the size of the datasets used in this study for training ammonia and colour forecasting model being 31 days, only the hour element was transformed into sine and cosine components as in Fig. 1.14b.

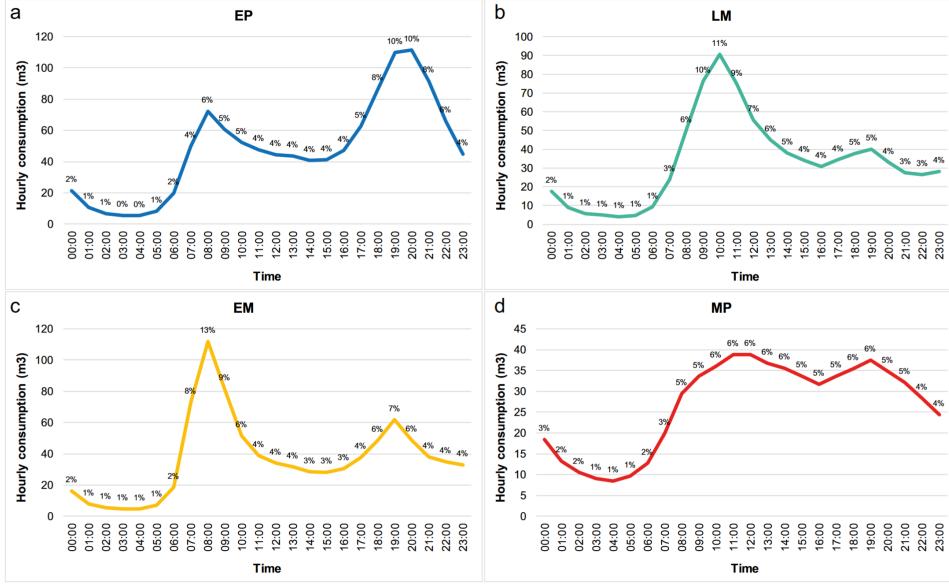


Figure 1.16: Hourly water consumption patterns in households (Abu-Bakar et al., 2021). (a) Cumulative pattern and percentage of hourly consumption for households in the “Evening Peak (EP)” cluster (b) Cumulative pattern and percentage of hourly consumption for households in the “Late Morning Peak Peak (LM)” cluster. (c) Cumulative pattern and percentage of hourly consumption for households in the “Early Morning Peak (EM)” cluster. (d) Cumulative pattern and percentage of hourly consumption for households in the “Multiple Peak (MP)” cluster. Consumption is in ( $\text{m}^3$ ).

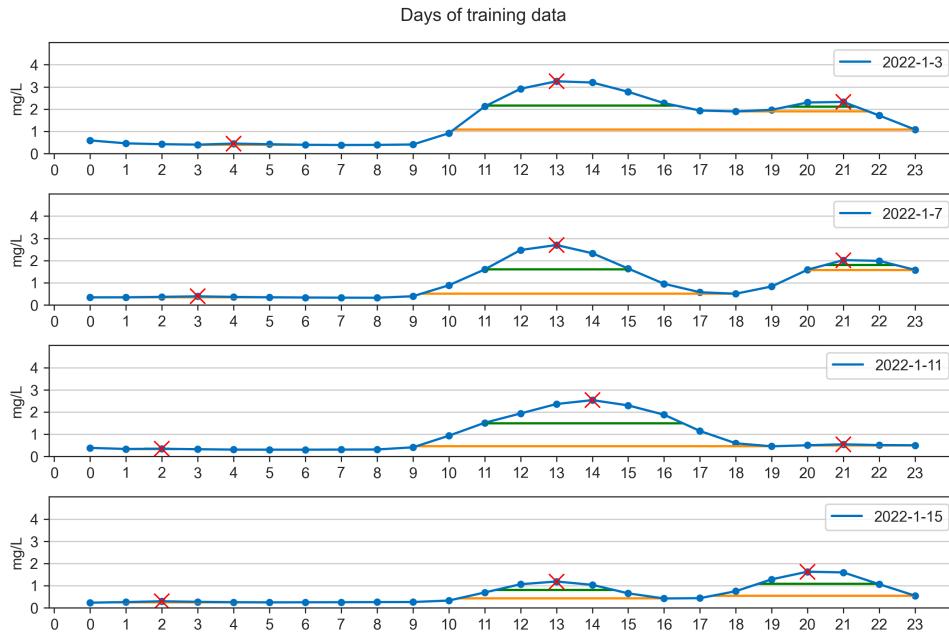


Figure 1.17: The daily patterns of ammonia concentrations on 3, 7, 11, and 15 January 2022.

#### 1.2.4 Data transformation

Before the pre-processed data was fed into the models for training, we need to split the data into three clusters, which were training (60%), validation (20%), and testing dataset (20%). Among each cluster, the data will be further split into input variables  $\mathbf{X}$  and output variables  $\mathbf{Y}$  (i.e., training X/training Y, testing X/testing Y). During the training process, machine learning algorithms will learn a target function  $f$  to best map  $\mathbf{X}$  to  $\mathbf{Y}$ . A training dataset is a set of examples (e.g., historical data) for models to learn the hidden trends and information in the data, shown in (a) in Fig. 1.8. Training loss is calculated by taking the sum of loss for each pair of input and output in the training dataset after every training cycle (i.e., epoch).

In this study, the model is designed to forecast values three hours into the future using the values from the past 24 hours. Fig. 1.18 illustrates a forecasting model's training and forecasting process. The length of the sliding time window in this study is set to be 25 (hours). In training set 1 (i.e., the first 24 hours from the training dataset), the blue block represents the observed values of 24 hours, while the yellow block is the first data point from the testing dataset (i.e., equivalent to the 25<sup>th</sup> hour of the training dataset). The model is required to learn how to map the blue block to the yellow block; the times of model learning is equivalent to the length of the training dataset deducted by the length of the sliding time window (i.e., the second to the last 25<sup>th</sup> hour will be mapped to the last hour of the training dataset). Once the training process is complete, the model will be able to generate a value, known as the inference, prediction, or forecast, given an input of 24 hours of data.

As shown in Fig. 1.19b, to forecast one hour into the future, the model will be input with 24 hours of observed values from the testing dataset, and the model will generate a value known as the forecasted values of the 25<sup>th</sup> hour. For predicting two hours into the future, the model will be input with 23 hours of observed values and the first forecasted values (i.e., the 25<sup>th</sup> hour). For forecasting three hours into the future, the model will be input with 22 hours of observed values and two forecasted values from the last two forecasting processes to generate the value, known as the 26<sup>th</sup> hour. As the sliding time window moves toward the future forecast horizons, the model forecasted results would rely more on the forecasted values instead of the observed values, making the forecasted values less reliable. In this study, a forecast horizon of three was selected for testing the reliability of the model forecasting performance.

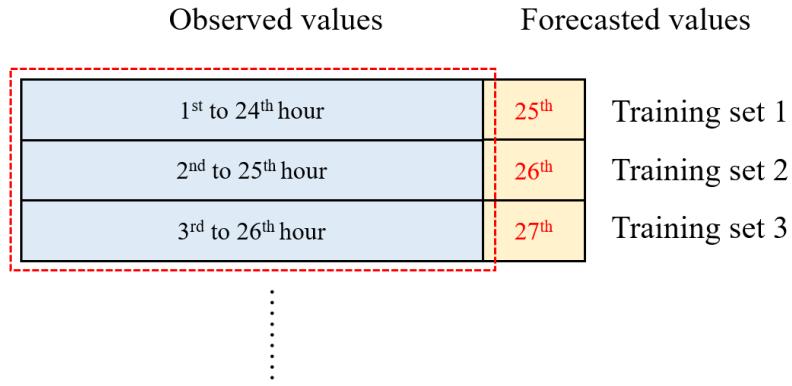
The function of a validation dataset, as in (b) in Fig. 1.8, is used to assess the model performance until we obtain the optimized hyperparameter settings, including the number of neurons



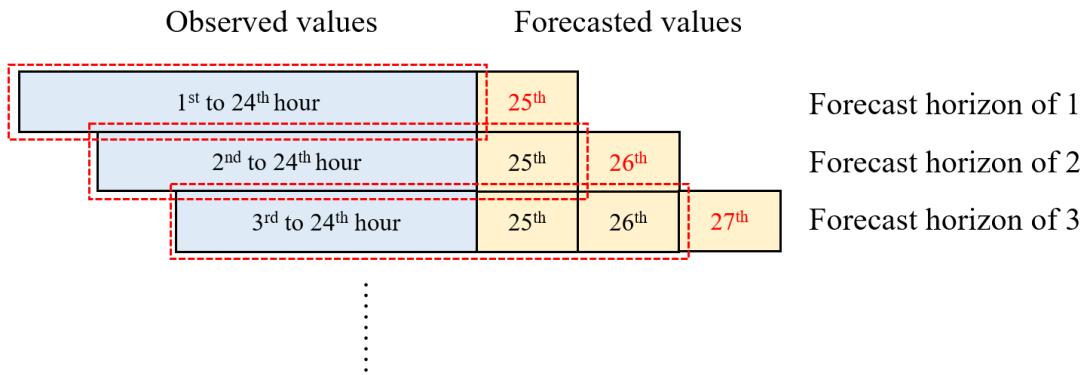
Figure 1.18: Concept of forecasting models (Liu, 2020).

in machine learning models, epoch, etc. The hyperparameter settings for each model will be discussed in the next section. The validation loss plays a vital role during the model training. The adjustments of the hyperparameters will directly reflect on the change of the validation loss; the lower the values, the better the model performance is. As the optimized model is obtained, a testing dataset is used to evaluate the performance of the forecasting model, as shown in (c) in Fig. 1.8. The testing datasets will only be input into the models when the models were tuned to the optimized settings and ready for the final evaluation. The testing datasets are also known as the unseen datasets, which can fairly evaluate the model performance. If the model tuning process was performed on the testing dataset, the model performance would be biased since the hyperparameters were adjusted in favour of the evaluation of the testing dataset.

In Fig. 1.8, the hyperparameters will remain the same once the optimized values were found, thus generating a baseline model performance from different machine learning algorithms. The baseline results will be further compared with the results from the model trained by the proposed model training steps, which include datasets that have been performed data smoothing and feature engineering techniques.



(a) Training process for each training set.



(b) Inference process for different forecast horizons.

Figure 1.19: Illustration of the training and inference process.

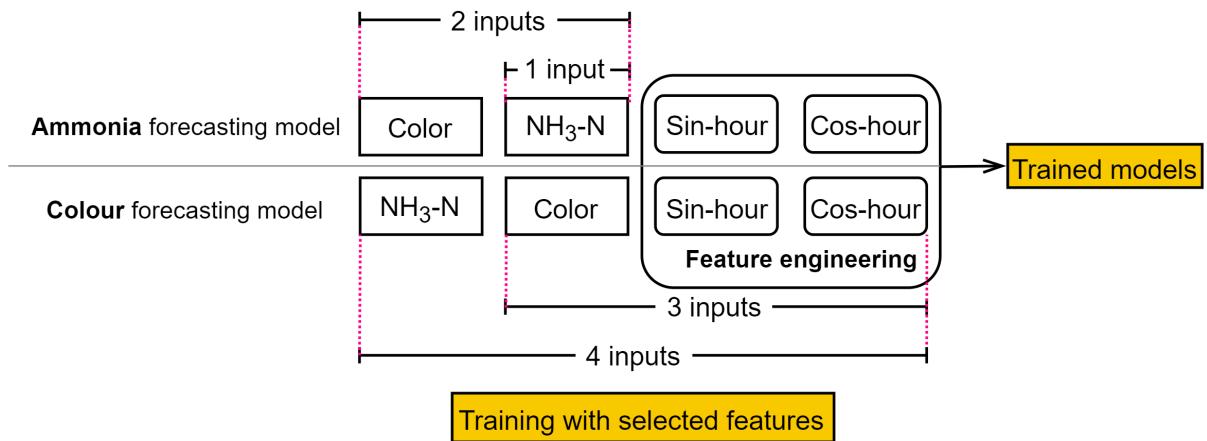


Figure 1.20: Illustration of feature selections for model training.

## 1.2.5 Feature selection

Fig. 1.20 illustrates which features were selected during the model training processes. In baseline model training steps, for both ammonia and colour forecasting models, only one feature was used for training for each model, which was ammonia and colour data, respectively.

Following the baseline model training steps, the model trained by a single feature will generate baseline models. The results from the final evaluation will be defined as the baseline model performance, which will be compared with the model evaluated results from the proposed model training steps. Once the baseline model performance is obtained, more features will be input to the model training processes in the order of two features, three features, and four features.

## 1.3 Machine learning models

### 1.3.1 Random Forest

The machine learning model used in this study (i.e., not deep learning models) is random forest (RF). It is an ensemble method in which the final output is obtained by averaging the results from multiple tree learners (Wang et al., 2021), as shown in Fig. 1.21a. The training algorithm applies the general technique of bootstrap aggregating, also known as bagging, to tree learners. Given a training set  $X = x_1, \dots, x_n$  with targets  $Y = y_1, \dots, y_n$ , bagging repeatedly ( $B$  times) selects a random sample with replacement (i.e., not putting the samples back to the population) of the training set and fits trees to these samples (Wikipedia, 2022a), RF generate outputs through the following steps:

For  $b = 1, \dots, B$  :

- 1) Sample (with replacement)  $n$  training examples from  $X, Y$ , call these  $X_b, Y_b$ .
- 2) Train a regression tree  $f_b$  on  $X_b, Y_b$ .
- 3) Predict unseen samples  $x'$  by averaging the predictions from all the regression tree learners on  $x'$  as in Eq. 1.3.1:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^B f_b(x') \quad (1.3.1)$$

### 1.3.2 Deep Neural Network

Artificial Neural Network (ANN) is a broad term that encompasses any form of Deep Learning model. A typical ANN consists of input, hidden, and output layers, and each layer comprises multiple neurons (i.e., nodes). The connected neurons simulate the human brain by processing

and transmitting input signals to the next nodes (Mohseni-Dargah et al., 2022). What sets it apart from an ANN model and a DNN model is that the former contains only one hidden layer while the latter has more than one, as shown in Fig. 1.21b. The DNN models are nonlinear, which finds the correct mathematical manipulation to turn the input into the output (Bangaloreai, 2018).

### 1.3.3 Recurrent Neural Network

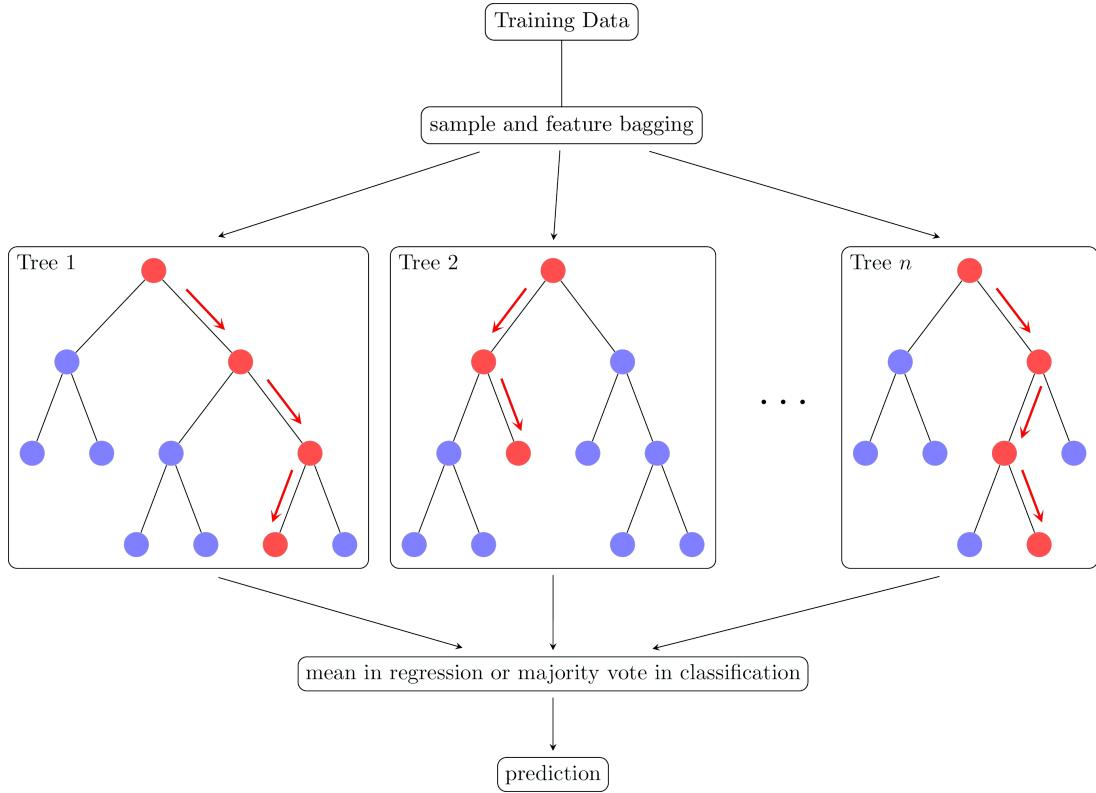
A recurrent neural network (RNN) is a type of Artificial Neural Network designed to work with sequence data. For instance, sequence data are time series, DNA, language, speech, sequences of user actions data, etc. The ammonia concentrations and colour levels data were time-series data, a series of data points listed in minute orders (Donges, 2021). A distinguishing characteristic of RNN is that they share parameters across each layer of the network by allowing information to be passed from the last step of the network to the next. Unlike RNN, feedforward networks like DNN have different weights across each node. The reuse of previous information for making the decision on RNN makes it capable of "learning" from the previous inputs. The realization of the memorizing function is through a memory unit called hidden state (i.e., a vector contains weights) in RNN architecture, which enables RNN to persist data, thus capturing short-term dependencies. The RNN architecture is presented in Fig. 1.22a. The general formulation of a RNN is expressed in Eq. 1.3.2 (Mamandipoor et al., 2020):

$$h_t = \sigma(W^h h_{t-1} + W^x x_t + b) \quad (1.3.2)$$

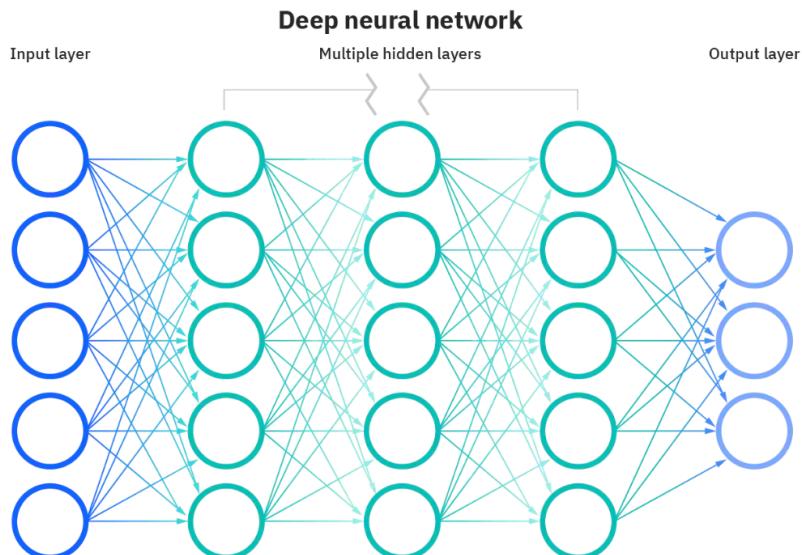
where  $x_t$  is the current input,  $h_t$  is the current hidden state (output),  $h_{t-1}$  is the previous output,  $W^x$  is the weights of the hidden state,  $W^h$  is the weight of the input,  $b$  is the bias,  $\sigma$  is the sigmoid activation function.

### 1.3.4 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a deep recurrent neural network (RNN), an advanced and improved version of RNN. The advent of LSTM solves problems requiring long-term temporal dependencies that RNN cannot learn due to the simple model architecture. The fundamental LSTM network is built on memory blocks called "cells", which are responsible for transferring and receiving the states (i.e., vectors) recording the information from the previous



(a) Random Forest (RF) (Riebesell, 2022).



(b) Deep Neural Network (DNN) (IBM, 2022).

Figure 1.21: Illustration of RF and DNN model structure.

cells. In a cell block, there is an input gate, a forget gate, and an output gate. The function of these three gates is to control the movement of the information into and out of the cell via the sigmoid function. The inputs of the cell will first go through a forget gate ( $f_t$ ) as Eq. 1.3.3a,

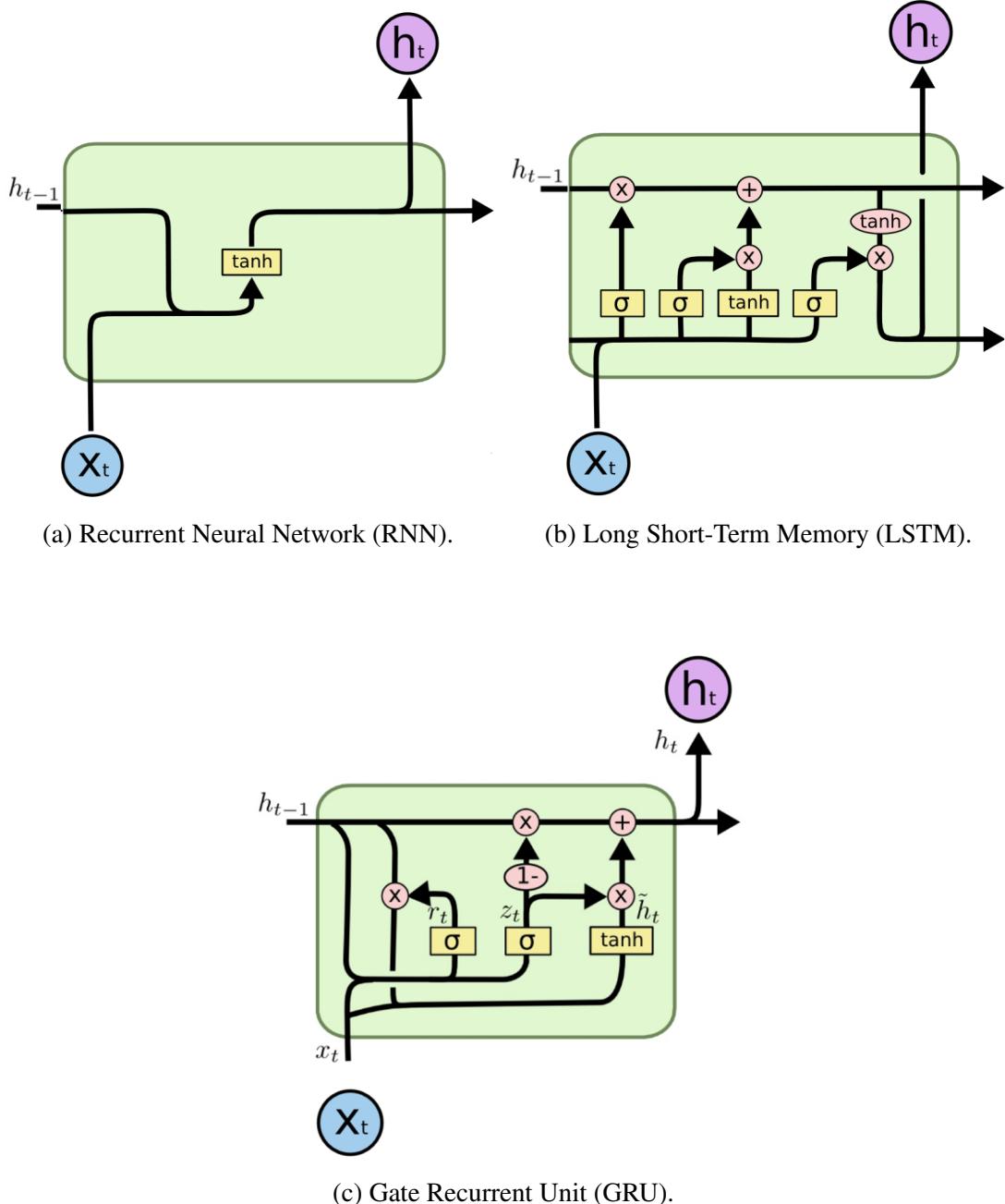


Figure 1.22: Variant architectures of Recurrent Neural Networks (adapted from Olah (2015)).  $x_t$  corresponds to the current input,  $h_{t-1}$  to the last hidden state (output),  $h_t$  to the current output,  $\tanh$  is the tangent activation function,  $\sigma$  is the sigmoid activation function,  $\times$  is the vector pointwise multiplication,  $+$  is the vector pointwise addition.

where the function will multiply each element in the input states by values ranging from 0 to 1 to realize the effect of "forget." Next, an input gate ( $i_t$ ) as in Eq. 1.3.3b will decide whether the new information should be updated or ignored by the sigmoid function (i.e., 0 or 1), followed by a tangent function giving the weight of importance (i.e., -1 to 1) to the values which passed by as in Eq. 1.3.3c. New memory then is appended to the previous memory  $C_{t-1}$  resulting a

new  $C_t$ . Lastly, output values ( $h_t$ ) is obtained based on output cell state ( $O_t$ ) as in Eq. 1.3.3e and Eq. 1.3.3f (Le et al., 2019). The equations for LSTM structure are shown in Eq. 1.3.3:

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \quad (1.3.3a)$$

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \quad (1.3.3b)$$

$$\tilde{C}_t = \tanh(W_n[h_{t-1}, X_t] + b_n) \quad (1.3.3c)$$

$$C_t = C_{t-1}f_t + \tilde{C}_ti_t \quad (1.3.3d)$$

$$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \quad (1.3.3e)$$

$$h_t = O_t \tanh(C_t) \quad (1.3.3f)$$

where  $f_t$  corresponds to the forget gate,  $i_t$  to the input gate,  $\tilde{C}_t$  to the candidate cell state,  $C_t$  to the current cell state,  $O_t$  to the output cell state,  $h_t$  to the output values,  $\sigma$  to the sigmoid function,  $X_t$  to the current input,  $\tanh$  to the tangent function,  $W$  and  $b$  are the weight matrices and bias of the corresponding output gate, respectively.

### 1.3.5 Gated Recurrent Unit

Gated Recurrent Unit (GRU) model is a variant of the LSTM model; by combining the forget gate and input gate into an update gate as in Fig. 1.22c, GRU has fewer parameters compared to LSTM. The advantage of GRU over LSTM is less computing power required while maintaining a similar model performance compared to LSTM. The inputs of the GRU model first enter the update gate ( $z_t$ ) as in Eq. 1.3.4a, where the function will help the model determine how much of the past information needs to be passed along to the future via sigmoid functions, and then followed by the reset gate ( $r_t$ ) as in Eq. 1.3.4b, which is used to decide how much of the past information to forget. Although Eq. 1.3.4a and Eq. 1.3.4b have the same inputs of  $X_t$  and  $h_{t-1}$ , the usages of the gates are different. The outputs of the reset gate will be used to determine the candidate hidden state ( $\tilde{h}_t$ ) as in Eq. 1.3.4c, where the tangent function will determine the importance of the current input ( $X_t$ ), reset gate output, and previous hidden state ( $h_t$ ). At the last step, the output values ( $h_t$ ) is calculated from the candidate hidden state ( $\tilde{h}_t$ ), previous hidden state ( $h_{t-1}$ ), and the outputs of update gate as in Eq. 1.3.4d. The equations of GRU structures are presented in Eq. 1.3.4 (Cheng et al., 2020):

$$z_t = \sigma(X_t W_{xz} + h_{t-1} W_{hz} + b_z) \quad (1.3.4a)$$

$$r_t = \sigma(X_t W_{xr} + h_{t-1} W_{hr} + b_r) \quad (1.3.4b)$$

$$\tilde{h}_t = \tanh(X_t W_{xh} + (r_t \circ h_{t-1}) W_{hh} + b_h) \quad (1.3.4c)$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h}_t \quad (1.3.4d)$$

where  $z_t$  corresponds to the update gate,  $r_t$  to the reset gate,  $\tilde{h}_t$  to the candidate hidden state,  $h_t$  to the output values,  $\sigma$  to the sigmoid function,  $\tanh$  to the tangent function,  $X_t$  to the current input,  $W$  and the  $b$  are the weight matrices and bias of the corresponding output gate, respectively.

### 1.3.6 Configurations of machine learning models

Hyperparameters are variables that we need to set before applying a learning algorithm to a dataset (Agrawal, 2019). For different tasks and datasets, the optimized hyperparameters vary, which makes the seeking of hyperparameters challenging. For RF models, only one hyperparameter needs to be selected—the number of estimators. As shown in Fig. 1.21a, each estimator, known as the tree in the forest, makes a decision. Therefore, we need to set the number of estimators for making a forecast. In this study, we tried different numbers of estimators and selected 500 estimators ultimately.

For training neural networks (NNs), the selection of hyperparameters is much more. The hyperparameters in NNs can be split into two categories, as shown in the followings:

#### Optimized hyperparameters

- 1) Learning rate
- 2) Number of epochs
- 3) Mini batch size

#### Model-specific hyperparameters

- 1) Number of hidden units (neurons)
- 2) Number of layers

The learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function. An iteration describes

the number of times a batch of data passed through the algorithm. In our study, the training data has a length of 432, with a batch size of one; the model will iterate 432 times to complete one epoch. There is a trade-off between the rate of convergence and overshooting when determining an optimal learning rate. A too high learning rate leads to a learning step jump over minima as in Fig. 1.23c, yet a too low learning rate will either be too slow to converge or get stuck in a local minimum loss as in Fig 1.23a. A good size of learning rate should reach the minimum loss at a reasonable time, as in Fig. 1.23b. However, searching for the most optimal learning rate can be time-consuming and a waste of computing power. In this study, we used a learning rate scheduler to achieve the same effect of using a decent learning rate. The scheduler can be set to reduce the learning rate as the epoch increases. When the algorithm detects the test loss is not reducing during the training within a designated epoch time, the learning rate will be multiplied by a customized factor. A factor of 0.5 and a patience of 10 were used in this study. The effect of using a learning rate scheduler is shown in Fig. 1.23d.

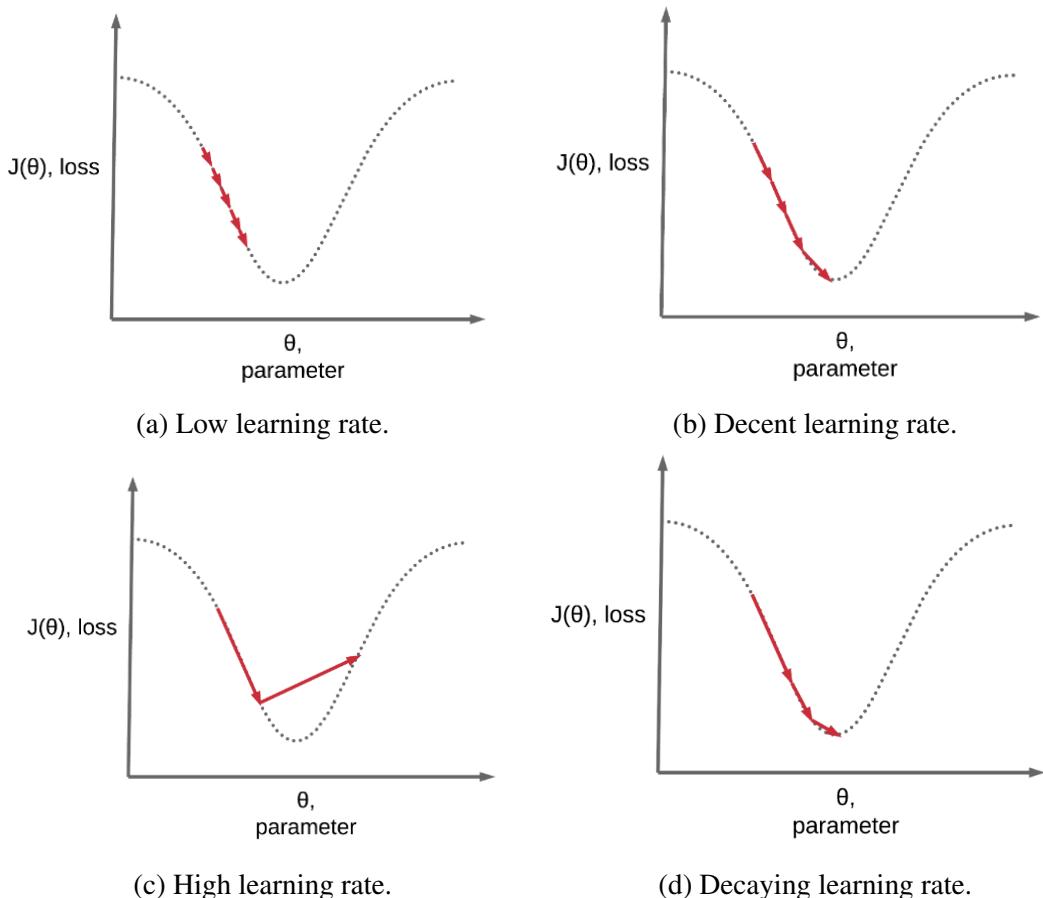


Figure 1.23: Illustration of how different step sizes of learning rate reach the minimum loss (Ritchie Ng, 2019).

In model-specific hyperparameter tuning, the number of neurons and the number of layers

need to be determined based on the complexity of our training dataset. The ammonia and colour datasets are considered simple and small datasets. In the hyperparameter tunings of the deep learning models, we simplified the model structure by lowering the number of layers to 1 except for the DNN model. If the number of hidden layers decreased to one, the DNN models would be called the ANN models according to the definition. The number of neurons was set to 10 to maintain simple deep learning models to prevent overfitting.

The settings of the optimized hyperparameters are listed in the followings in the final iteration of model hyperparameter tuning:

### **Optimized hyperparameters**

- 1) Learning rate: 5e-05
- 2) Number of epochs: 100
- 3) Batch size: 1

Table 1.2: Final model configurations.

Model	Input	h.d <sup>a</sup>	Output	Num. of Exp <sup>b</sup>	Comments
RF	24 <sup>c</sup>	-	3	3	Estimators = 500
DNN	24	2	1	3	h.d = 10 neurons
RNN	24	1	1	3	h.d = 10 neurons
GRU	24	1	1	3	h.d = 10 neurons
LSTM	24	1	1	3	h.d = 10 neurons

<sup>a</sup> Hidden layer.

<sup>b</sup> The times the experiments were repeated.

<sup>c</sup> 24 hourly data points were input into the models for training.

## CHAPTER 2

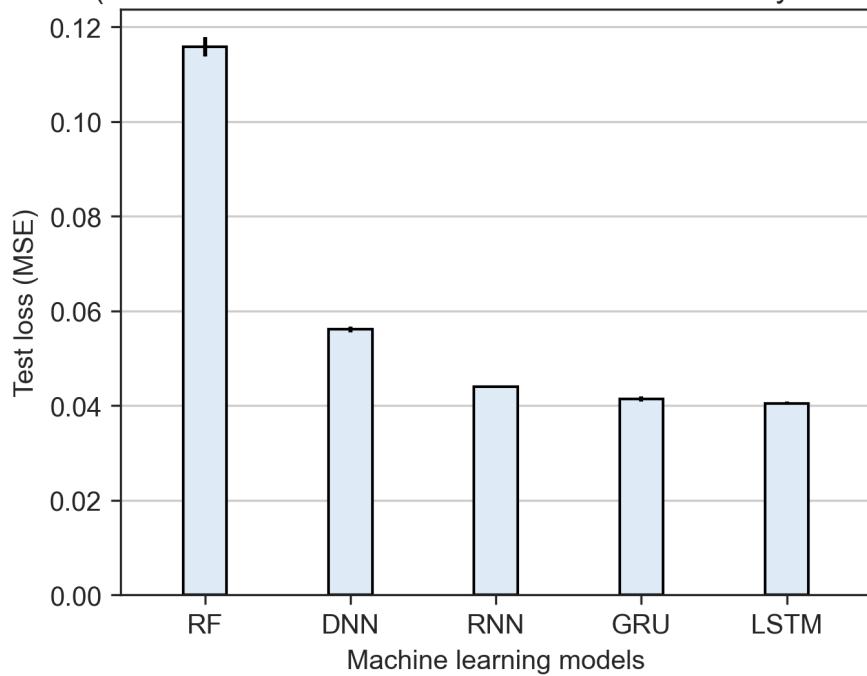
### RESULTS AND DISCUSSION

#### 2.1 Baseline performance of the forecasting models

In this study, five machine learning algorithms were trained with univariate datasets to predict the ammonia concentrations and colour levels in the reclaimed water system. All baseline models are trained by training datasets which were not applied with data pre-processing and feature engineering techniques. The forecasting model performance is presented in Fig. 2.1. As shown in Fig. 2.1a, the test loss values of RF, DNN, RNN, GRU, and LSTM models are 0.1158, 0.0561, 0.0440, 0.0414, and 0.0405, respectively. RF model is the least capable model in forecasting ammonia concentrations, given that its test loss is significantly higher than all the other four deep learning models. The cause of poor RF model performance can be attributed to its simple model structure. RF model generates results based on the averaging results from each decision tree (i.e., each decision tree will generate a prediction based on entropy and information gain). There is only one available hyperparameter for tuning RF models: the estimators (i.e., the number of the decision tree). Therefore, throughout the entire model tuning process. We observed the RF model had the lowest test loss at the beginning among all the models, and the increased estimators did not help lower the test loss values. Meanwhile, several iterations of hyperparameter tunings help the deep learning models to reduce the test loss values to critical values, which were lower than the test loss of the RF model. The gradual reductions of test loss values for deep learning models can be attributed to the nature of their complex model architectures (i.e., a good quantity of neurons, neurons are designed to perform unique functions) and the available hyperparameters for tuning. For instance, the number of hidden layers, number of neurons, learning rate, and epoch are adjustable. The customizable hyperparameters in the deep learning models allow the researchers to fully explore the possibilities of training better models, and the superior performance is reflected in the values of test loss obtained from the optimized hyperparameter settings.

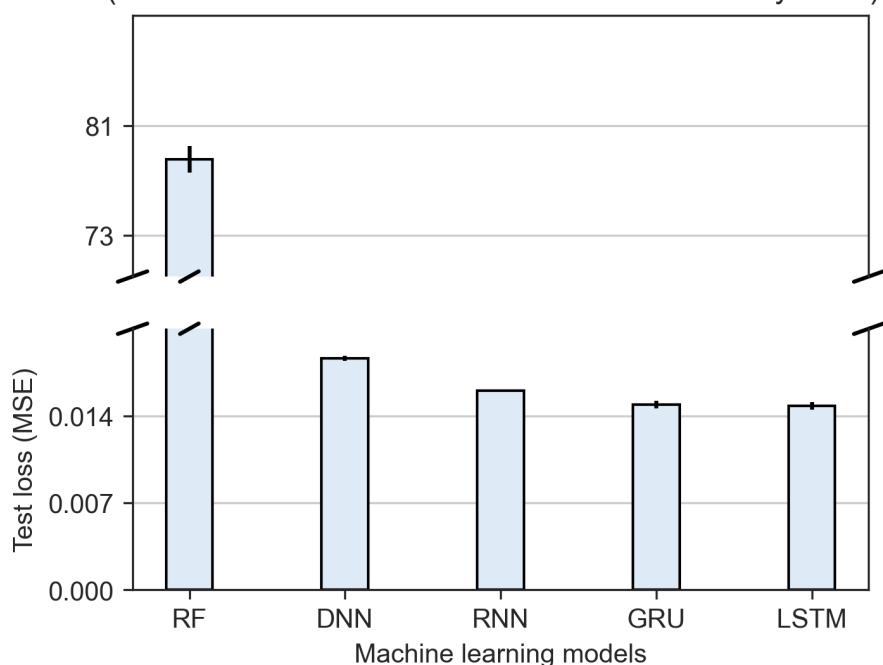
GRU and LSTM models learn the data in similar ways by utilizing memorizing cells to pass and receive critical information from the previous memorizing cells, known as the architecture

Baseline model performance in forecasting NH<sub>3</sub>-N.  
(Evaluated on test dataset from 16 to 22 January 2022)



(a) Test loss values from five ammonia forecasting models.

Baseline model performance in forecasting colour.  
(Evaluated on test dataset from 16 to 22 January 2022)



(b) Test loss values from five colour forecasting models.

Figure 2.1: Baseline performance of the ammonia and colour forecasting models.

of recurrent neural network. Compared to RNN models, both models contain more "gates" in the architectures to help control the flow of information, enabling the models to capture more details. The number of gates in RNN, GRU, and LSTM is one, three, and four; theoretically, GRU and LSTM can learn more information from the data based on a greater number of gates. The results in Fig. 2.1a showed good agreement with our understanding that LSTM performed better than GRU, followed by RNN models based on the values of test loss. For DNN models, the lack of memorizing cells in the model architecture relates to the poorer capability of learning information hidden in time-series datasets. In other words, DNN models cannot comprehend the information hidden in each datapoint in sequence, making the time-series dataset merely a common set of data. The DNN model with a test loss of 0.0440, higher than the 0.0414 of the RNN models, fully justifies the need to use the architecture of recurrent neural networks for training ammonia forecasting models.

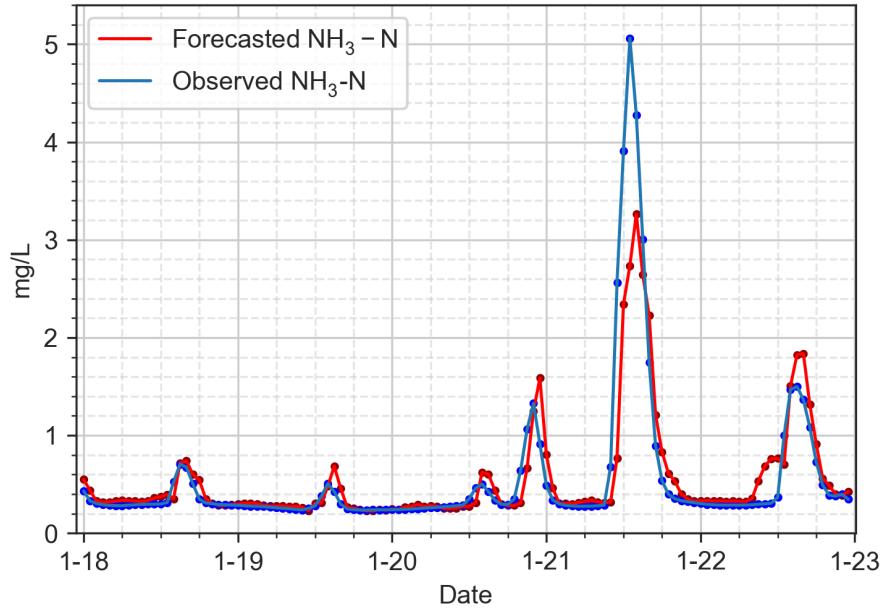
In Fig. 2.1b, the test loss from colour forecasting models are 78.5296, 0.0186, 0.0160, 0.0149, and 0.0148 for RF, DNN, RNN, GRU and LSTM models, respectively. We first noticed the highest test loss value of 78.5296 in the RF model compared to the other four, making RF model the worst model in forecasting colour levels. The extremely high MSE values were caused by the colour levels fluctuating in a wider range of 80 to 160 Hazen Units. The large discrepancy between the actual and predicted colour levels increases the error values, which are further amplified as the MSE values are calculated by the average of the squares of the errors. As shown in Fig. 2.3a, on 20 January 2022, the errors between the ground truth and forecasted values are up to around 30 Hazen Units, which contribute to a large increase of MSE values in the test loss. RF model is regarded as an inferior model for forecasting colour levels using the data collected in SWHEPP.

The performance of DNN, RNN, GRU, and LSTM models, from the best to the least, are identical to what we observed in the results of ammonia forecasting models. LSTM model has the lowest test loss of 0.0148, followed by the GRU, RNN, and DNN models. In colour forecasting models, the model performance of LSTM is very close to GRU, with a difference of less than 0.0001 (i.e., less than 1%). However, the lowest test loss generated from the LSTM model in all the experiment runs (i.e., three runs) is 0.0143, which is lower than 0.0146 from the GRU model. Indicating LSTM model has more potential in forecasting time-series data.

The significantly higher test loss of RF models compared to other models can be visualized by plotting the forecasted values with the ground truths (i.e., observed values). In Fig. 2.2

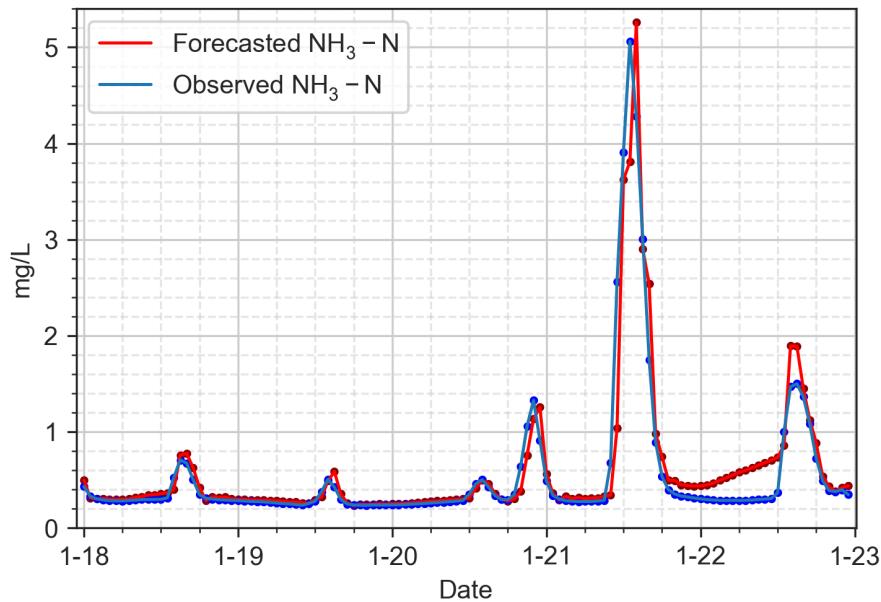
and Fig. 2.3, one-step-ahead forecast horizon of ammonia concentrations and colour levels are plotted by RF as in Fig. 2.2a and Fig. 2.3a and LSTM models as in Fig. 2.2b and Fig. 2.3b. It is easier to observe that the RF models are less capable of predicting the water quality parameters.

The ammonia forecasting results.  
(R-squared=0.7743)



(a) Baseline RF model forecasting ammonia concentration.

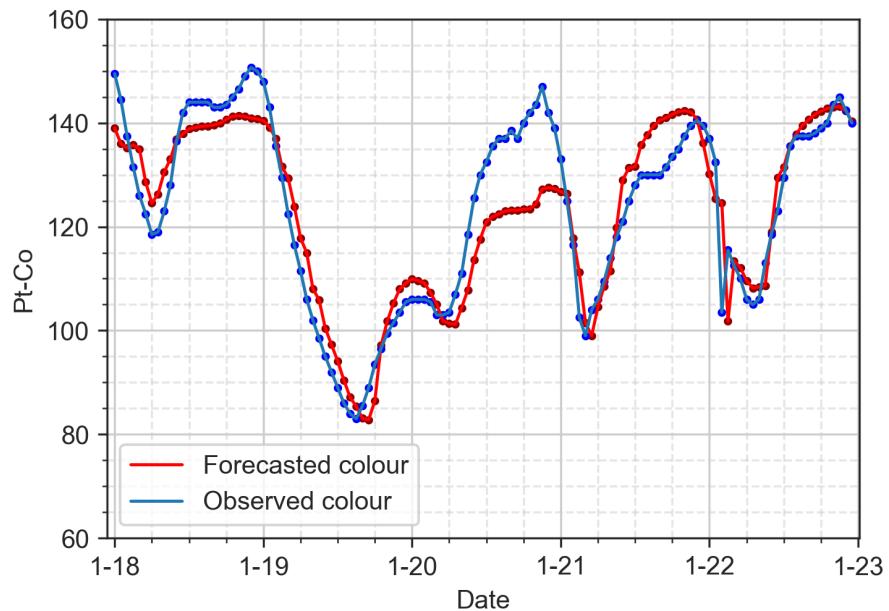
The ammonia forecasting results.  
(R-squared=0.8847)



(b) Baseline LSTM model forecasting ammonia concentration.

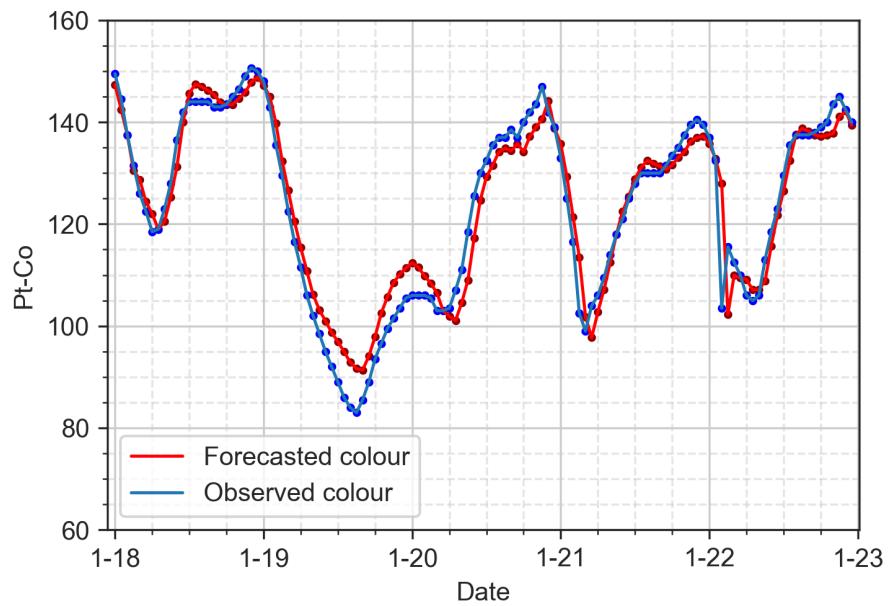
Figure 2.2: Visualization of the baseline ammonia forecasting results.

The colour forecasting results.  
(R-squared=0.8295)



(a) Baseline RF model forecasting colour levels.

The colour forecasting results.  
(R-squared=0.9311)



(b) Baseline LSTM model forecasting colour levels.

Figure 2.3: Visualization of the baseline colour forecasting results.

## 2.2 Improved performance on forecasting models using data pre-processing techniques

### 2.2.1 Models trained by pre-processed datasets

In this study, we investigate whether the datasets treated by the proposed data pre-processing techniques can improve the baseline model performance using the same hyperparameter settings. As shown in Table. 2.1 and Table. 2.3, we listed all the test loss values of five machine learning algorithms trained with each proposed pre-processed technique for ammonia concentrations and colour levels forecasting. The machine learning algorithm trained by datasets that were applied with SG filters at different window sizes is denoted as model-sg5, model-sg7, and model-sg9. The naming rule applies the same to EWMA filtered dataset; the method of outlier removal for ammonia data is denoted as model-or; models trained with the raw datasets are denoted as model-obs (i.e., observed dataset).

Table 2.1: Baseline performance of the ammonia forecasting model, evaluated on test dataset from **16 to 22 January 2022**. Loss values were calculated by MSE.

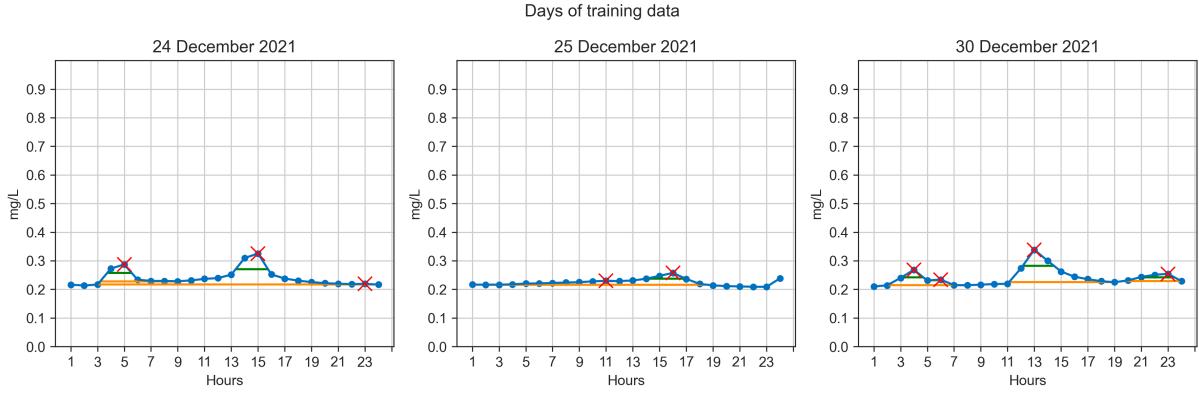
Model-Dataset	Test loss	Valid loss	Model-Dataset	Test loss	Valid loss
GRU-sg7	0.0383	1.2508	RNN-or	0.0432	1.6345
GRU-sg5	0.0385	1.2644	RNN-ew3	0.0434	1.6041
LSTM-ew3	0.0388	1.0796	RNN-obs	0.0440	1.6734
LSTM-sg5	0.0388	1.2346	RNN-sg9	0.0442	1.7046
LSTM-sg7	0.0388	1.1804	DNN-obs	0.0561	3.2383
GRU-ew2	0.0389	1.1891	DNN-sg5	0.0562	3.2170
GRU-ew4	0.0391	1.2390	DNN-ew2	0.0563	3.1677
GRU-ew3	0.0392	1.2199	DNN-ew3	0.0569	3.2317
LSTM-ew2	0.0392	1.0969	DNN-sg7	0.0570	3.2014
LSTM-ew4	0.0395	1.1219	DNN-ew4	0.0571	3.2188
GRU-sg9	0.0396	1.3097	DNN-or	0.0572	3.1972
LSTM-or	0.0398	1.2612	DNN-sg9	0.0574	3.2484
LSTM-obs	0.0405	1.3993	RF-obs	0.1158	-
GRU-or	0.0405	1.2366	RF-sg9	0.1196	-
LSTM-sg9	0.0410	1.3076	RF-ew2	0.1286	-
GRU-obs	0.0414	1.3638	RF-or	0.1294	-
RNN-sg5	0.0415	1.5088	RF-sg5	0.1298	-
RNN-ew2	0.0421	1.5425	RF-ew3	0.1313	-
RNN-sg7	0.0423	1.6267	RF-sg7	0.1409	-
RNN-ew4	0.0432	1.5992	RF-ew4	0.1441	-

The improvements in the performance of ammonia forecasting models are most significant

with models trained by SG filtered datasets. Training GRU models with an sg7 filtered dataset reduced the test loss of GRU-obs from 0.0414 to 0.0383 (-7.5%). LSTM-sg7 also successfully decreased the test loss value of LSTM-obs from 0.0405 to 0.0388 (-4.2%), while RNN-sg5 reduced the test loss value of RNN-obs from 0.0440 to 0.0415 (-5.7%). Using SG filters on the training datasets improves the performance of LSTM, GRU, and RNN models. However, the DNN and RF models trained by sg filtered datasets did not show a superior model performance compared to the test loss values of 0.0561 and 0.1158 of DNN-obs and RF-obs, respectively. Given that DNN and RF models perceive the data points as clusters of individuals, data smoothing using SG filters is not expected to help improve their model performance. SG filter smoothes the data points by convoluting both previous and subsequent data points, making a series of data points correlated or linked with each other. Such data property is believed to be captured by the memorizing cells in recurrent neural networks, such as RNN, GRU, and LSTM models. From the results in Table. 2.1, all the recurrent neural networks-based models outperformed all the DNN and RF models. It can be concluded that DNN and RF models are poor options for training time-series models, even with the use of the SG filter technique.

The RNN-or, GRU-or, and LSTM-or models, which were trained with datasets applied with outlier removal methods, showed lower test loss values of 0.0432 (-1.8%), 0.0405 (-2.2%), and 0.0398 (1.7%) compared to test loss values of 0.0440, 0.0414, and 0.0405 from RNN-obs, GRU-obs, and LSTM-obs, respectively. We also noticed that the improvements of RNN-or, GRU-or, and LSTM-or are minor compared with the models trained by SG and EWMA filtered datasets. In this method, three days of abnormal data were removed from an 18-day dataset as in Fig. 2.4, which accounts for around 15% of the data. Despite the fact that 15% of the data was removed, the improvement in lowering the test loss values was slight. It is suggested that the deep learning models are smart enough to neglect the noise in the training datasets while performing forecasts from the test dataset.

RNN, GRU, and LSTM models trained by EWMA filtered datasets also showed good improvements in the model performance. RNN-ew2, GRU-ew2, and LSTM-ew3 showed lower test loss of 0.0421 (4.3%), 0.0389 (6.0%), and 0.0388 (4.2%) compared to RNN-obs, GRU-obs, and LSTM-obs of 0.0440, 0.0414, and 0.0405, respectively. EWMA filters modified the data points by averaging the value of the current data points with previous ones, making the data property almost identical to the SG filtered data. Both SG and EWMA filters similarly influenced the baseline models, in which LSTM obtained the lowest test loss values, followed by GRU and RNN models. By far, the results only suggest that both filters are robust techniques



(a) Validation dataset from January 2022.

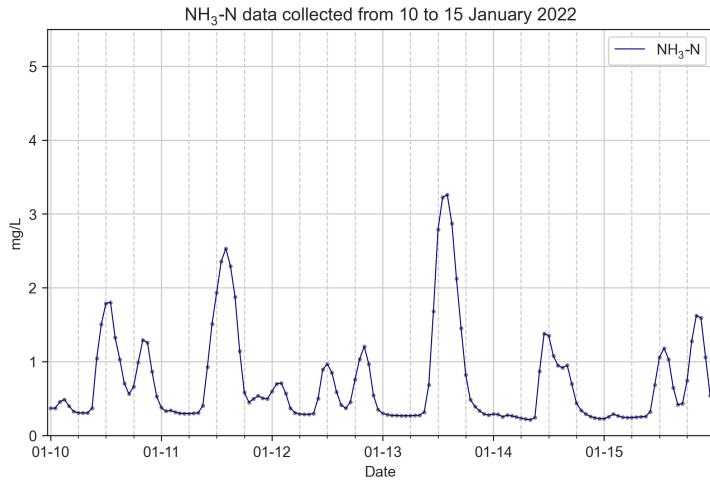
Figure 2.4: Results of the removed outliers from the training dataset.

in terms of lowering the test loss, yet we cannot draw conclusions about which filter is more effective in improving the model performance. In addition, we discovered our test loss values to be abnormal when inspecting models' validation loss and the test loss values.

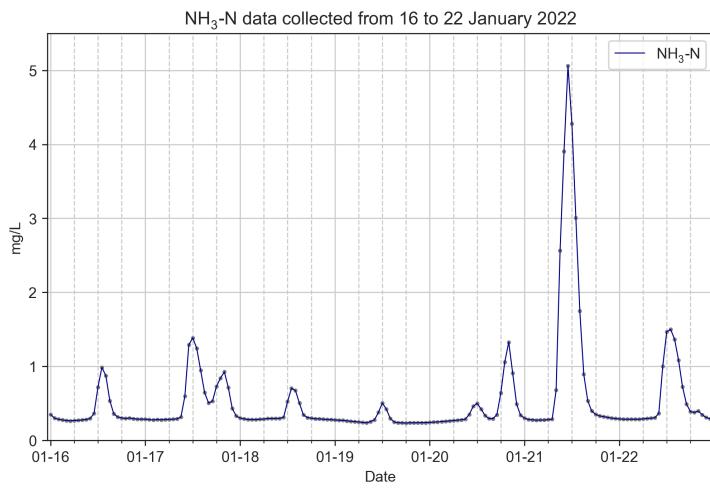
Empirically, the best-performed Model-Dataset combination should match the lowest test with the lowest validation loss values when using the same testing dataset to evaluate a group of models. For instance, the GRU-sg7 model in forecasting ammonia has the lowest test loss of 0.0383, yet the validation loss of 1.2508 only ranks tenth among the validation loss values. The top three lowest validation loss models are LSTM-ew3, LSTM-ew2 and LSTM-ew4, yet the top three lowest test loss models are from GUR-sg7, GRU-sg5, and LSTM-ew3 models. This finding points to the potential heterogeneity between the validation and testing datasets. The limitation of this study's validation and testing datasets is the small dataset size, resulting in specific daily fluctuation patterns of ammonia may only occur in the testing dataset. In all the available ammonia data, we selected the data from October 2021 as the second testing dataset for its high similarity to the validation dataset in January 2022.

As shown in Fig. 2.5, the fluctuation patterns of NH<sub>3</sub>-N in validation dataset as in Fig. 2.5a is much resemble to the testing dataset from Fig. 2.5c compared to testing dataset from Fig. 2.5b. Further tests were carried out using a testing dataset from October to re-evaluate the model performance from Table. 2.1. It is expected that the Model-Dataset ranks of test and validation loss values from the lowest to the highest will change. To the best of my understanding, the comparisons between testing and validation loss are not discussed in the currently available research papers in the modelling of the wastewater treatment industry.

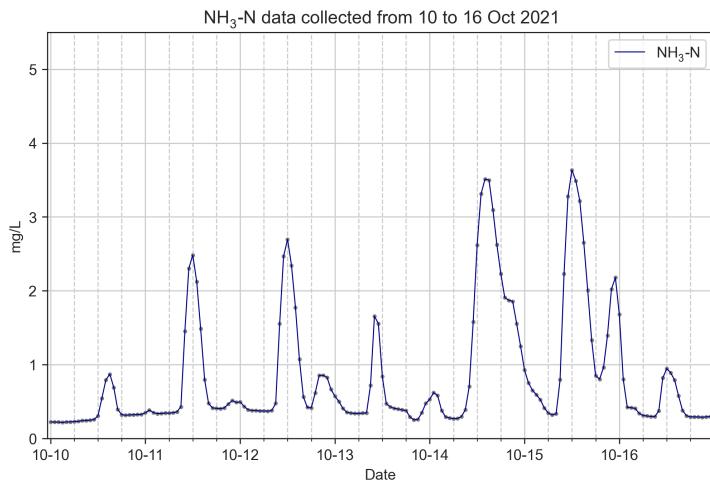
As shown in Table. 2.2, the models with the top lowest test loss values are 0.0158, 0.0161,



(a) Validation dataset from January 2022.



(b) Testing dataset from January 2022.



(c) Testing dataset from October 2021.

Figure 2.5: Illustration of the heterogeneity and homogeneity between validation and different testing datasets.

0.0163 for LSTM-ew3, LSTM-ew2, and LSTM-ew4, which match the top three lowest validation loss values of 1.0796, 1.0969, and 0.1219. This is in good agreement with how the heterogeneity of the datasets can impact the model performance. The evaluations of the ammonia forecasting models in October 2021 showed completely different outcomes compared to those in January 2022. Instead of GRU, LSTM becomes the best model for training the ammonia forecasting model. For LSTM models, the top three Model-Dataset combinations are LSTM-ew3, LSTM-ew2, and LSTM-ew4; for GRU models, they are GRU-ew3, GRU-ew4, and GRU-ew2; for RNN models are RNN-ew4, RNN-ew2, and RNN-ew3. It is evident that EWMA filters have a more significant influence on the model performance for all the recurrent neural network models than SG filters. However, given the small dataset size, caution must be taken if the EWMA filter is applied in future works.

Table 2.2: Baseline performance of the ammonia forecasting models, evaluated on test dataset from **10 to 16 October 2021**. Loss values were calculated by MSE.

Model-Dataset	Test loss	Valid loss	Model-Dataset	Test loss	Valid loss
LSTM-ew3	0.0158	1.0796	RNN-or	0.0197	1.6345
LSTM-ew2	0.0161	1.0969	RNN-sg7	0.0201	1.6267
LSTM-ew4	0.0163	1.1219	RNN-sg9	0.0205	1.7046
LSTM-sg5	0.0166	1.2346	RNN-obs	0.0206	1.6734
GRU-ew3	0.0167	1.2199	DNN-ew3	0.0316	3.2317
GRU-ew4	0.0169	1.2390	DNN-or	0.0316	3.1972
GRU-ew2	0.0170	1.1891	DNN-sg7	0.0316	3.2014
GRU-sg9	0.0174	1.3097	DNN-ew2	0.0318	3.1677
LSTM-obs	0.0175	1.2366	DNN-ew4	0.0319	3.2188
LSTM-or	0.0177	1.2612	DNN-obs	0.0319	3.2383
GRU-sg5	0.0178	1.2644	DNN-sg5	0.0319	3.2170
GRU-sg7	0.0180	1.2508	DNN-sg9	0.0319	3.2484
LSTM-sg7	0.0180	1.1804	RF-sg9	0.1307	-
GRU-or	0.0187	1.3993	RF-sg7	0.1311	-
LSTM-sg9	0.0188	1.3076	RF-sg5	0.1343	-
GRU-obs	0.0189	1.3638	RF-ew2	0.1346	-
RNN-ew4	0.0190	1.5992	RF-ew3	0.1368	-
RNN-ew2	0.0191	1.5425	RF-obs	0.1443	-
RNN-ew3	0.0193	1.6041	RF-ew4	0.1451	-
RNN-sg5	0.0195	1.5088	RF-or	0.1477	-

The test loss values of the colour forecasting models are presented in Table. 2.3. The top six lowest test loss models are LSTM-ew4, LSTM-ew2, LSTM-ew3, GRU-ew3, GRU-ew2, and GRU ew4 with the values of 0.0136, 0.0138, 0.0138, 0.0140, 0.0142, and 0.0143, respectively. LSTM models are shown to be the best-performed model in forecasting colour levels.

The results also suggest that all the top lowest test loss models are trained by EWMA filtered datasets. We found that LSTM, GRU, and RNN models trained by EWMA filtered datasets generated the top lowest test loss values compared to the same models trained by SG filtered datasets. Interestingly, in both colour and ammonia forecasting models, LSTM models trained by EWMA filtered dataset showed the most superior performance, as shown in Table. 2.2 and Table. 2.3. LSTM models trained with EWMA filtered datasets are proved to be the best model and pre-processing techniques for training colour forecasting models in this study.

In the investigation of how a small dataset can influence the model results, we found that the top three lowest validation loss values are LSTM-sg9, LSTM-sg7, and LSTM-ew4, which rank the 7<sup>th</sup>, 20<sup>th</sup>, and 1<sup>st</sup> as the lowest test loss values. In this study, there is no extra colour testing dataset we can retrieve from the historical dataset, despite the fact that we were keen to investigate the homogeneity and heterogeneity of the colour validation and testing dataset. Compromises have to be made during the analysis of colour forecasting models.

Table 2.3: Baseline performance of the colour forecasting models, evaluated on test dataset from **16 to 22 January 2022**. Loss values were calculated by MSE.

Model-Dataset	Test loss	Valid loss	Model-Dataset	Test loss	Valid loss
LSTM-ew4	0.0136	0.7515	RNN-obs	0.0160	1.0623
LSTM-ew2	0.0138	0.8011	LSTM-sg7	0.0161	0.7439
LSTM-ew3	0.0138	0.7547	LSTM-sg5	0.0168	0.8355
GRU-ew3	0.0140	0.8068	DNN-sg5	0.0180	1.4702
GRU-ew2	0.0142	0.8330	DNN-sg7	0.0180	1.4823
GRU-ew4	0.0143	0.7694	DNN-sg9	0.0180	1.4574
LSTM-sg9	0.0143	0.7137	DNN-ew4	0.0181	1.4632
RNN-ew3	0.0144	0.8492	DNN-ew3	0.0182	1.4716
RNN-ew4	0.0147	0.8476	DNN-ew2	0.0183	1.4946
RNN-sg9	0.0147	0.8363	DNN-obs	0.0186	1.5397
LSTM-obs	0.0148	0.9744	RF-sg9	63.6847	
GRU-obs	0.0149	0.9927	RF-sg7	73.8263	
RNN-ew2	0.0150	0.9083	RF-ew3	75.1974	-
GRU-sg9	0.0151	0.7575	RF-ew4	77.8829	-
RNN-sg5	0.0158	0.8846	RF-obs	78.5296	-
RNN-sg7	0.0158	0.8755	RF-ew2	78.8753	-
GRU-sg7	0.0159	0.7791	RF-sg5	81.0696	-
GRU-sg5	0.0160	0.8080	-	-	-

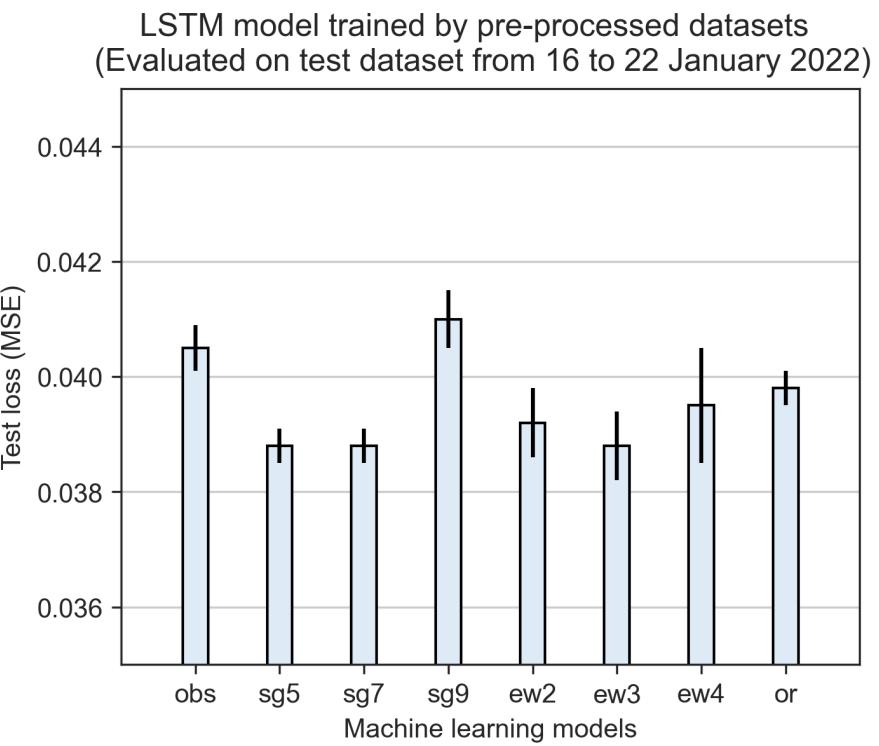
By comparing the baseline performance and the influences of data pre-processing techniques on machine learning models, our findings appear to be well substantiated by using LSTM models for training ammonia and colour forecasting models due to their outstanding

model performance evaluated by test loss values. Although EWMA filters showed surprising effects on improving the performance of most models, the conclusions of determining which pre-processing techniques are the optimum option should be treated with caution. Thus, the testings of the proposed model training processes will include all the pre-processing techniques for model training, and LSTM will be used as the only machine learning model.

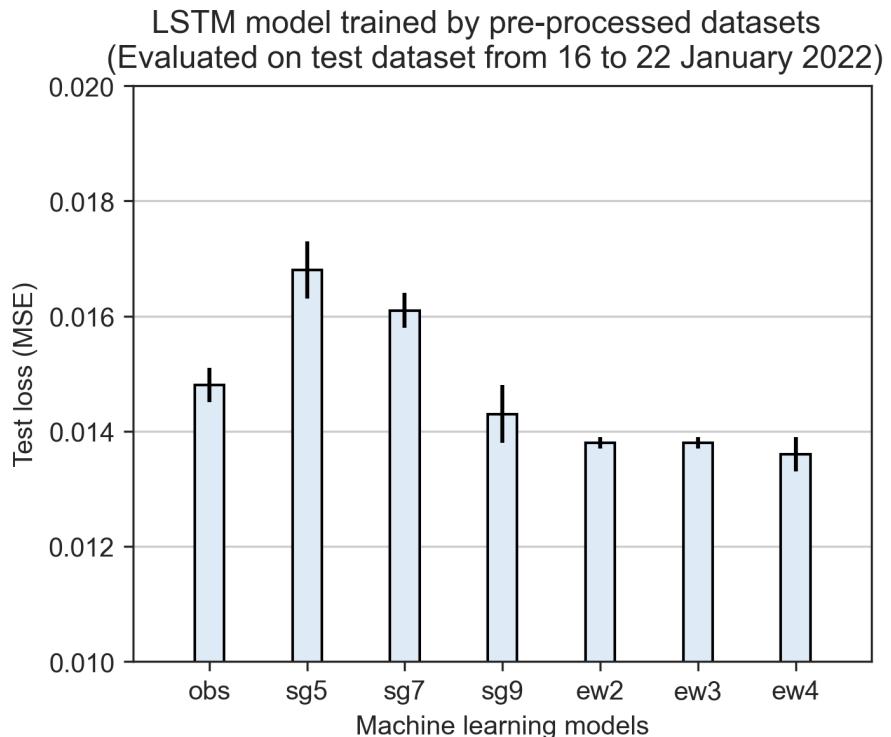
### **2.2.2 The effects of window sizes of the data smoothing filters**

The influences of window sizes in the data smoothing process are investigated using LSTM models and illustrated in Fig. 2.6. Larger and smaller SG window sizes have different impacts on ammonia and colour forecasting models. In ammonia forecasting models, as shown in Fig. 2.6a, LSTM models trained with SG filtered datasets with window sizes of 5, 7, and 9 have the test loss values of 0.0388, 0.0388, and 0.0410. The results suggested that modifying data points at higher degrees may negatively affect the model training process. The results from models trained by EWMA filtered datasets showed good agreement with this finding. The model trained with EWMA filtered datasets with the windows size of 2, 3, and 4 have the test loss values of 0.0392, 0.0388, and 0.0395. A higher test loss value is observed in LSTM-ew4 compared to LSTM-ew3.

For colour forecasting models, as shown in Fig. 2.6b, LSTM models trained by SG filtered datasets with window sizes of 5, 7, and 9 have test loss values of 0.0168, 0.0161, and 0.0143. LSTM models trained by EWMA filtered datasets with window sizes of 2, 3, and 4 showed test loss values of 0.0138, 0.0138, and 0.0136. From these results, we observed that larger window sizes helped the models achieve lower test loss for colour forecasting models, which does not support what we have concluded for the ammonia forecasting models. One possible explanation for the contradictory results is that ammonia and colour data have different sensitivity toward the data smoothing filters. For instance, ammonia concentrations change between the values of 1.0 to 7.0 mg/L, while colour levels vary from 80 to 160 Hazen Units, making the values of filtered data points less significant in colour data. In other words, if ammonia data points are shifted from the original values after applying data smoothing techniques, the values might be biased considering the fluctuated range of ammonia is small, while the shifted colour level data can be less biased among the sample regarding the fluctuation range of colour level is much larger. By far, we can not conclude how to select the window sizes of the data smoothing filters. The unpredictable influences of applying data smoothing filters on forecasting models impede



(a) Baseline performance of ammonia forecasting models trained by LSTM.



(b) Baseline performance of the colour forecasting models trained by LSTM.

Figure 2.6: Baseline performance of the ammonia and colour forecasting models.

the determination of the optimum data smoothing techniques in the subsequent experiments.

## 2.3 Exploit hidden patterns in the MBR effluent quality to enhance model performance

### 2.3.1 Ammonia forecasting models

In the section of feature engineering, we have introduced the selection and creation of the extra input features for training forecasting models, as shown in Fig. 1.20. In this study, a forecasting model trained by one feature is called an univariate model and denoted as LSTM-1; a forecasting model trained by two features is called a multivariate model and denoted as LSTM-2. For models trained by three and four features are denoted as LSTM-3 and LSTM-4. In Fig. 2.7, the performance of ammonia forecasting models trained by two to four inputs (i.e., LSTM-2, LSTM-3, LSTM-4) is compared with the baseline performance (i.e., LSTM-1-obs) to demonstrate how the feature engineered features influenced on the model outputs.

As shown in Fig. 2.7, LSTM-4-obs, LSTM-3-obs, LSTM-2-obs, and LSTM-1-obs have the test loss values of 0.0432, 0.0426, 0.0411, and 0.0405, respectively. This result indicates that LSTM models trained with more features resulted in poorer model performance. Based on our understanding to the extra features such as color levels and sine/cosine features, models trained with more features are expected lower test values. The model performance from LSTM-sg7 and LSTM-sg9 fits well with what we hypothesized. The test loss values of LSTM-4-sg7, LSTM-3-sg7, LSTM-2-sg7, LSTM-1-sg7 are 0.0369, 0.0373, 0.0379, 0.0388, respectively. For LSTM-4-sg9, LSTM-3-sg9, LSTM-2-sg9, and LSTM-1-sg9, the test loss values are 0.0384, 0.0391, 0.0409, 0.0410, respectively. These findings showed that the test loss values of the LSTM models trained by sg7 and sg9 filtered datasets followed the trends of  $\text{LSTM-4} < \text{LSTM-3} < \text{LSTM-2} < \text{LSTM-1}$ . The most remarkable results are from LSTM models trained by SG filtered dataset at a window size of 7. Comparing to the baseline model performance (i.e., LSTM-1-obs), the test loss values of LSTM-1-sg7, LSTM-2-sg7, LSTM-3-sg7 and LSTM-4-sg7 reduced by 4.2%, 6.4%, 7.9%, and 8.9%, respectively.

Our findings in the ammonia forecasting models suggest that colour level is an indispensable feature for improving the model performance. LSTM-2 models trained by datasets applied with any pre-processing techniques showed lower test loss compared to LSTM-1, except LSTM-2 trained by dataset without applying any methods. Strong evidence leads us to believe that the fluctuation of ammonia concentration is highly correlated with the colour levels in SHWEPP influent even without direct evidence.

The methods of training LSTM models on pre-processed datasets have proved their benefits in improving baseline model performance. Yet, the test loss values were only reduced slightly for those models trained with EWMA filtered datasets. As shown in Fig. 2.7, LSTM-3-ew2, LSTM-4-ew2, LSTM-3-ew4, and LSTM-4-ew4 shared very similar test loss values to LSTM-1-obs, indicating the advantages of enhanced training datasets were not fully reflected on the model performance when LSTM models were trained with EWMA filtered datasets.

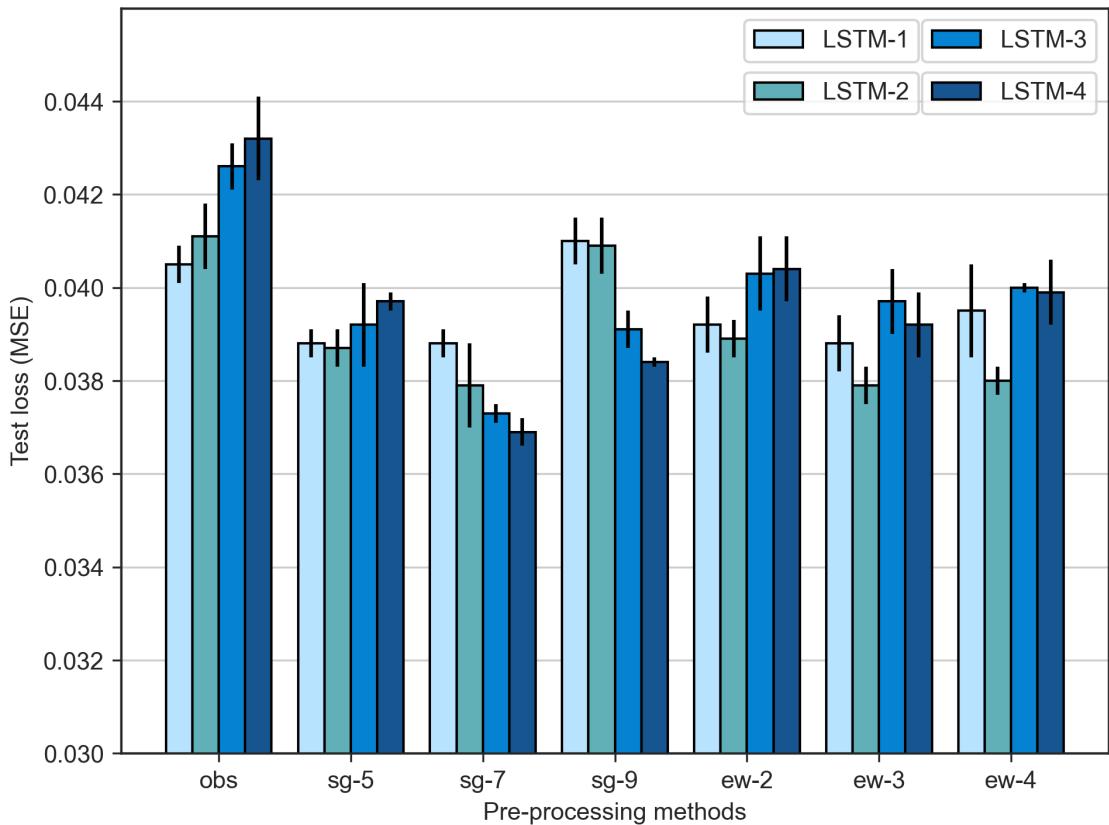


Figure 2.7: Comparisons of the model performance in forecasting ammonia concentrations.

### 2.3.2 Colour forecasting models

As shown in Fig. 2.8, the baseline performance is LSTM-1-obs with test loss value of 0.0148, and many models trained by both SG and EWMA filtered datasets show lower test loss values. The performance of models trained by SG filtered datasets was rather disappointing. In the results of models trained by sg-5 and sg7 filtered datasets, only LSTM-3-sg5, LSTM-3-sg7, and LSTM-4 sg-7 showed lower test loss values of 0.0144, 0.0143, and 0.0136, respectively, compared to LSTM-1-obs. Models trained by sg9 and all the EWMA filtered datasets showed improvement over LSTM-1-obs. In LSTM-3-sg9, we observed the lowest test loss value of

0.0129, which is 28.6% lower than the test loss values of 0.0148 from LSTM-1-obs.

The test loss values of LSTM-4-sg9, LSTM-4-ew2, LSTM-4-ew3, and LSTM-4-ew4 are higher than LSTM-3-sg9, LSTM-3-ew2, LSTM-3-ew3, and LSTM-3-ew4, by 0.0009, 0.0009, 0.0002, and 0.0002, respectively. This finding indicates that training with ammonia and the sine/cosine features deteriorate the model performance for color forecasting models. From what we found in the results of ammonia forecasting models, we concluded that the test loss values increase more when more features were input to the training datasets. In the colour forecasting results, the finding contrasts what we have found previously.

The interpretation for the higher test loss in LSTM-4 models in sg9, ew2, ew3, and ew4 filtered datasets compared to LSTM-3 and LSTM-2 models is that ammonia and sine/cosine features are irrelevant to the development of colour forecasting models. In the process of generating feature engineering, we observed that colour substances are mixed with municipal wastewater at the volume to volume ratio of 1 to 50. Hence, we can infer that the model outputs of forecasted colour levels are highly subject to the input of ammonia concentration. In the training process of the machine learning model, the model treats each input feature with equivalent importance; however, when the model is trained and input with unseen data, the model cannot differentiate which input feature actually influences more on the model outputs. The results suggest that it is best to train features of colour data and sin/cosine features for training color forecasting models.

### 2.3.3 Model forecasting results on different forecast horizons

In this study, ammonia and colour forecasting models were input with data from the past 24 hours to forecast the values three hours into the future. To demonstrate how the proposed model training methods improved the baseline model performance, the forecasted results were visualized for easier comparisons. As shown in Fig. 2.9, the proposed model training methods helped the model to forecast better on 21 January as in Fig. 2.9b during the low ammonia concentration period. On other days, both LSTM-1-obs and LSTM-4-sg7 shared similar accuracy in forecasting ammonia concentration.

In forecasting ammonia concentration in the second hour into the future as in Fig. 2.10, both model showed much higher MSE values of 0.2916 and 0.2351 compared to the MSE values of 0.0647 and 0.0529 from Fig. 2.9. Both models forecasted the ammonia concentration fairly on 17, 18, 19, and 20 January but forecasted poorly on 21 January. During the last two days of forecasting, the patterns of ammonia concentration were quite different compared to the

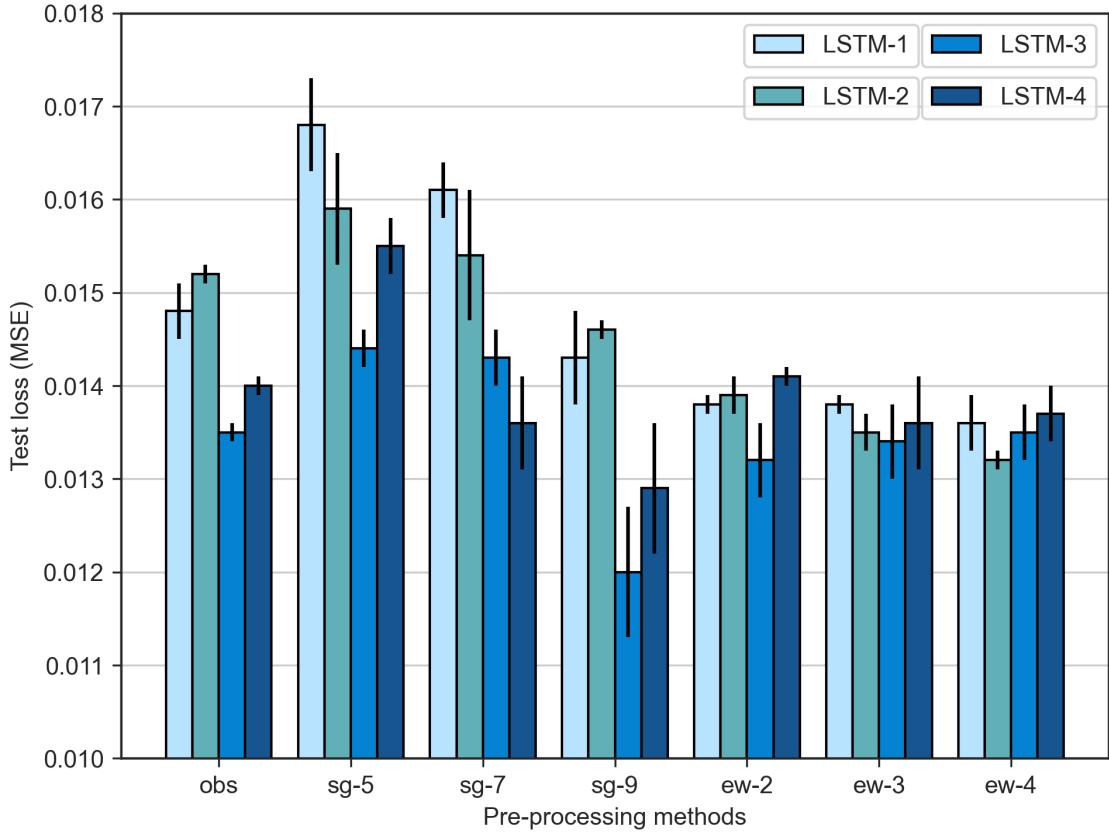
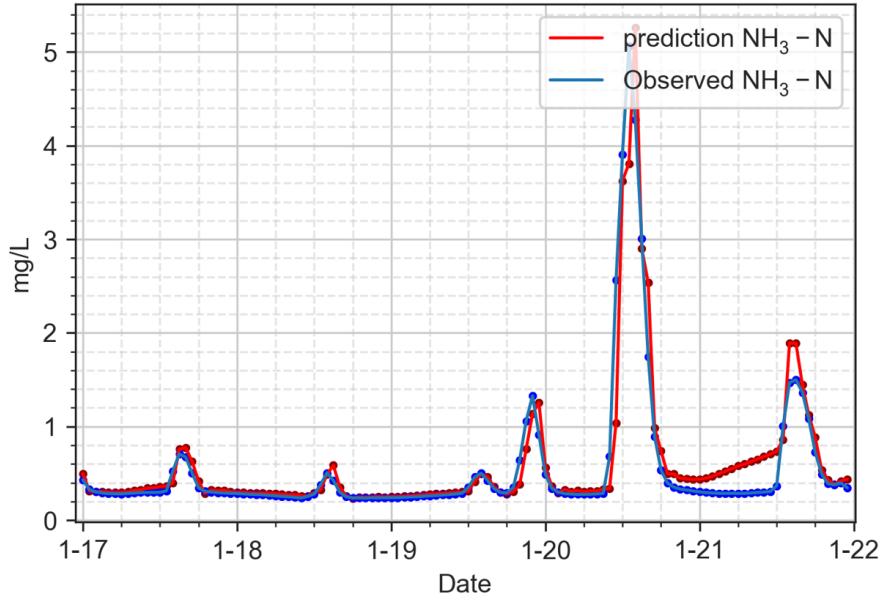


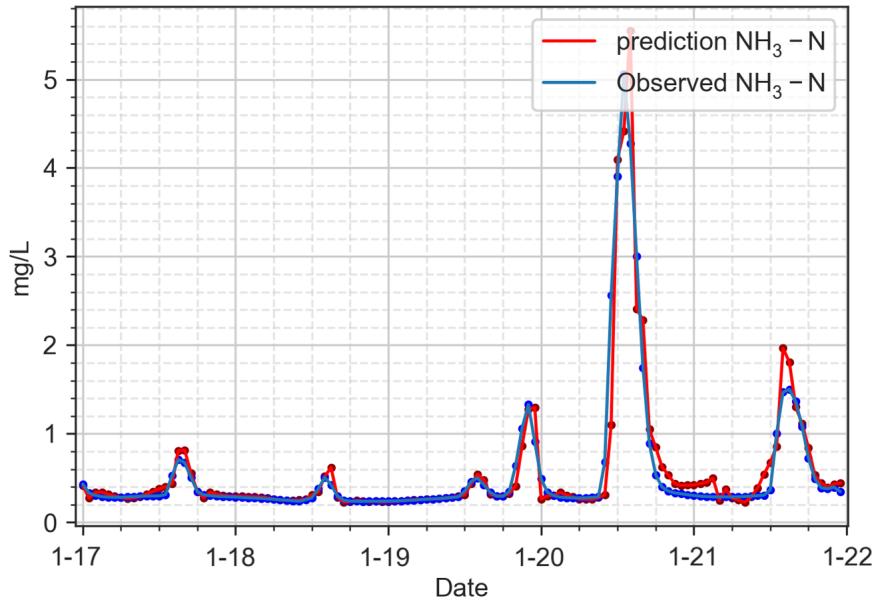
Figure 2.8: Comparisons of model performance in forecasting colour levels.

previous four days. For instance, on the 20 January, the peak concentration of ammonia during the day reached to 5.0 mg/L. Both models seemed unable to precisely forecast the trend of the ammonia concentrations, resulting in overestimated ammonia concentration around noon on 21 January. The proposed model training methods did not seem to forecast better than the baseline model. Forecasting longer time horizons requires an adequate training dataset size in terms of the number of training features and the length of the dataset. The ammonia forecasting model as in Fig. 2.10b was trained with four features with a dataset length of 18 days. Yet, the results suggested that the quantity of training dataset is not sufficient enough for forecasting two hours into the future.

In forecasting ammonia concentration at a forecast horizon of three, although the MSE values of 0.7637 from LSTM-4-sg7 are lower than 0.8025 from LSTM-1-obs, the difference between the two model performance is negligible. For the LSTM-4-sg7 model, we observed ammonia concentrations lower than 0 mg/L were forecasted on 20 January. Both LSTM-4-sg7 and LSTM-1-obs models poorly forecasted the peak ammonia concentration of over 5.0 mg/L on 21 January, which is 3.0 mg/L higher than the actual ammonia concentration on the same day.



(a) LSTM-1-obs, MSE = 0.0647, R-squared = 0.8847

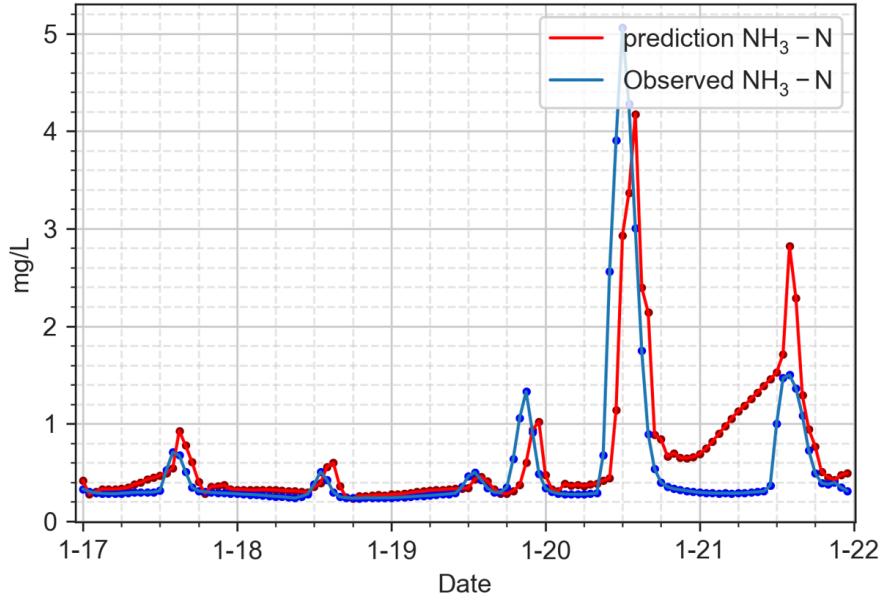


(b) LSTM-4-sg7, MSE = 0.0529, R-squared = 0.9057

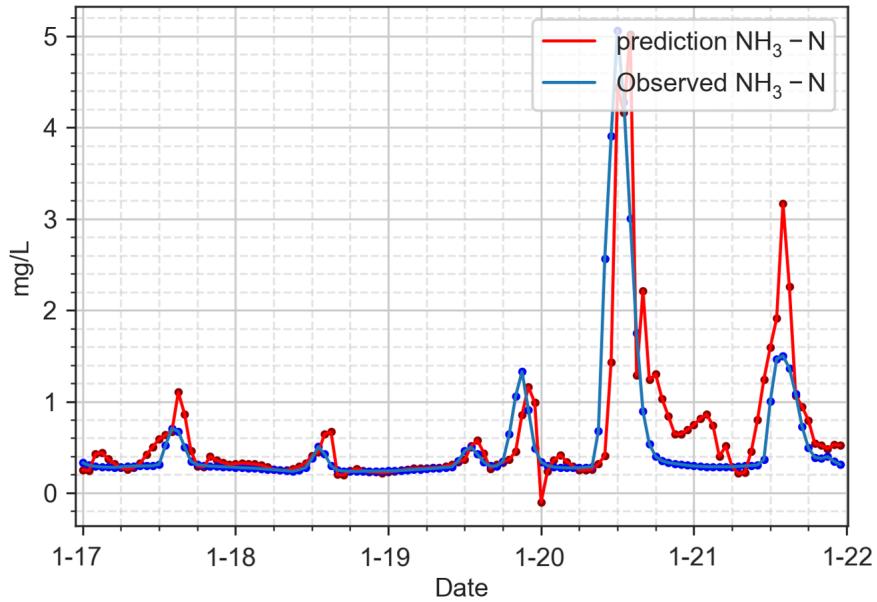
Figure 2.9: Visualization of the ammonia forecasting models at forecast horizon of one.

The results suggest that even with the use of proposed model training methods, the capability of the model performance is still limited due to the limited size of the training dataset.

LSTM-1-obs and LSTM-3-sg9 models forecasted colour levels at a forecast horizon of one with good MSE values of 22.4922 and 17.5955. The errors between the actual and forecasted values are mostly less than 5 Hazen Units. On 18 January, the colour levels dropped to 80 Hazen Units, and both models forecasted colour levels with errors values of up to 5 Hazen Units and



(a) LSTM-1-obs, MSE = 0.2916, R-squared = 0.4805

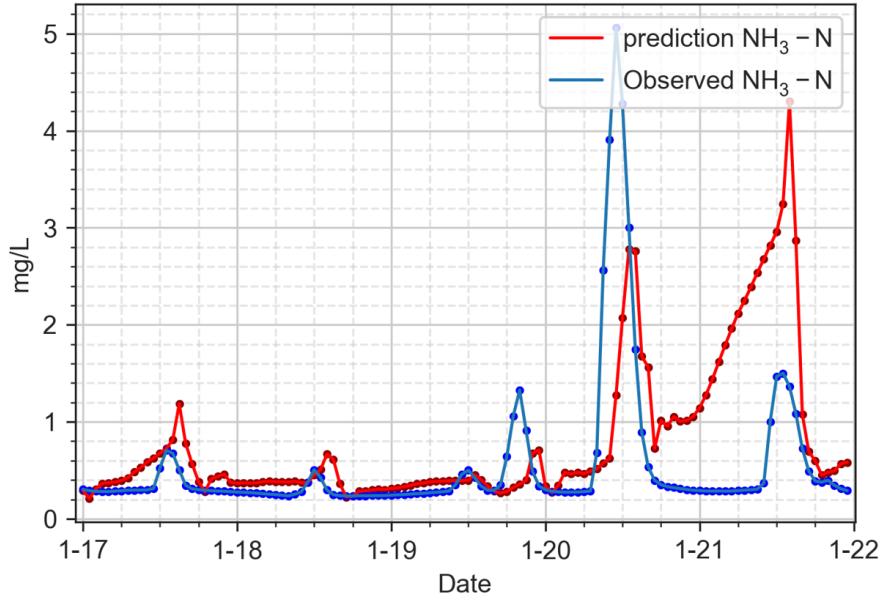


(b) LSTM-4-sg7, MSE = 0.2351, R-squared = 0.5812

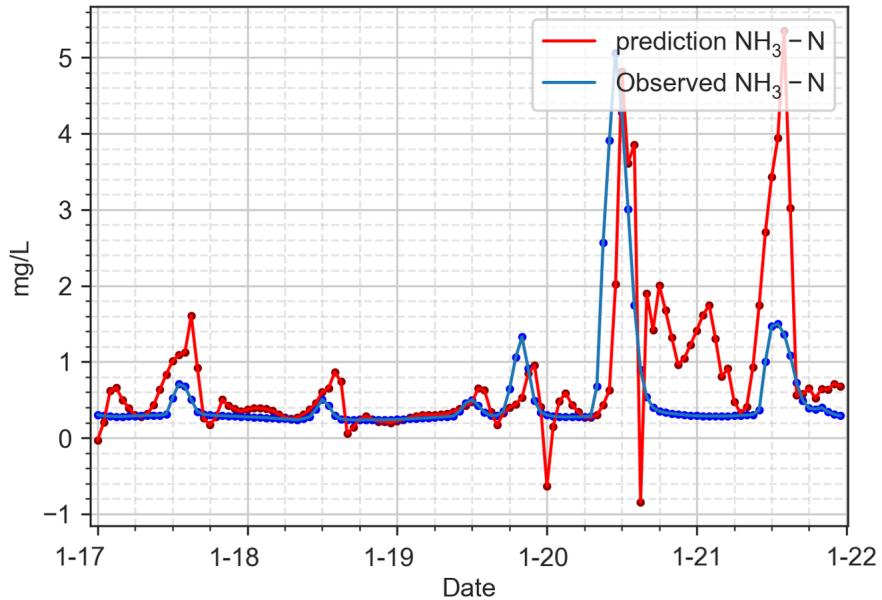
Figure 2.10: Visualization of the ammonia forecasting models at forecast horizon of two.

higher. Although on 22 January, the LSTM-3-sg9 model forecasted the colour level of 92 Hazen Units, which is 10 Hazen Units off from the actual values, the general model performance is satisfactory.

In forecasting colour levels at a forecast horizon of two, the MSE values of LSTM-1-obs and LSTM-3-sg9 increased from 22.4922 and 17.5955 to 62.6678 and 47.4252. The forecasting errors expanded from less than 5 Hazen Units on average to 10 Hazen Units. In Fig. 2.13,



(a) LSTM-1-obs, MSE = 0.8025, R-squared = -0.4291

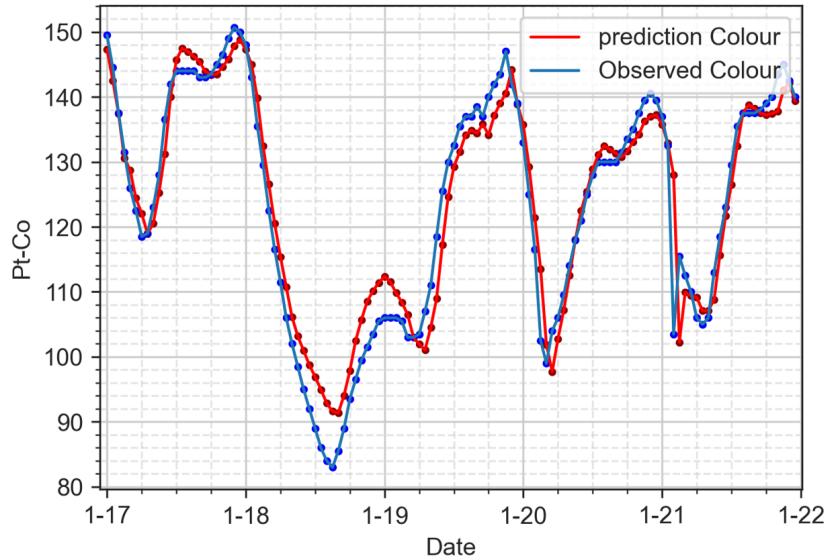


(b) LSTM-4-sg7, MSE = 0.7637, R-squared = -0.3599

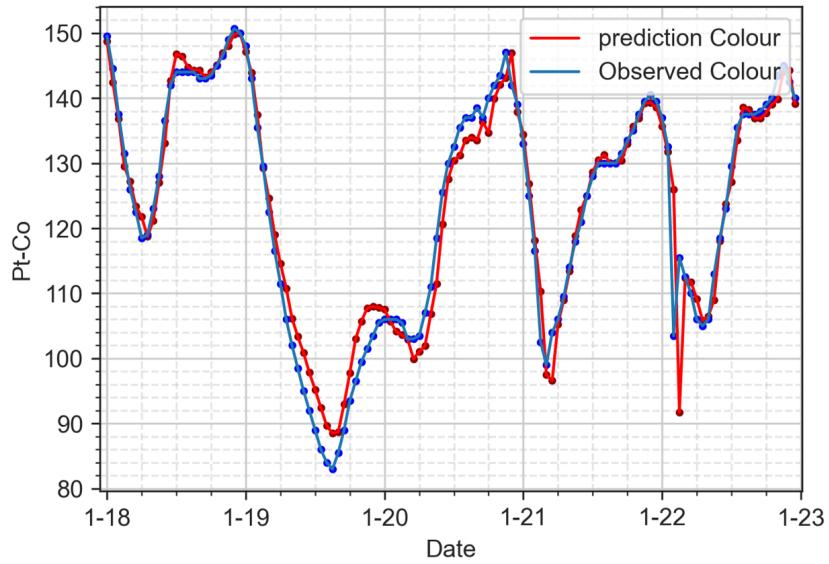
Figure 2.11: Visualization of the ammonia forecasting models at forecast horizon of three.

LSTM-3-sg9 showed more reliable forecasting results compared to LSTM-1-obs by generating minor errors between the forecasted and actual values. However, the lowest forecasted colour level on 22 January has increased from 10 to 24 Hazen Unis, and we can see clearly that the models were getting less reliable in forecasting two hours into the future in forecasting colour levels. The cause of it can also be attributed to insufficient quantity of training dataset.

In Fig. 2.14, the MSE values of LSTM-1-obs and LSTM-3-sg9 have increased to 116.8928



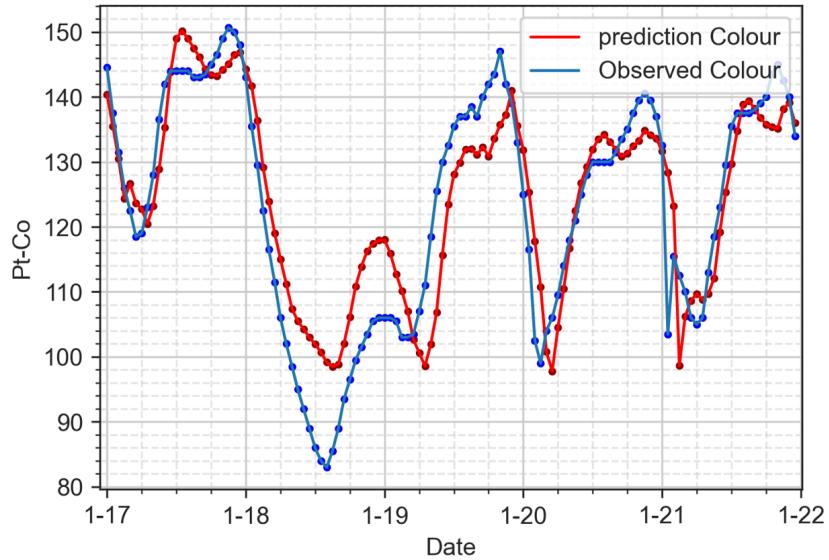
(a) LSTM-1-obs, MSE = 22.4922, R-squared = 0.9311



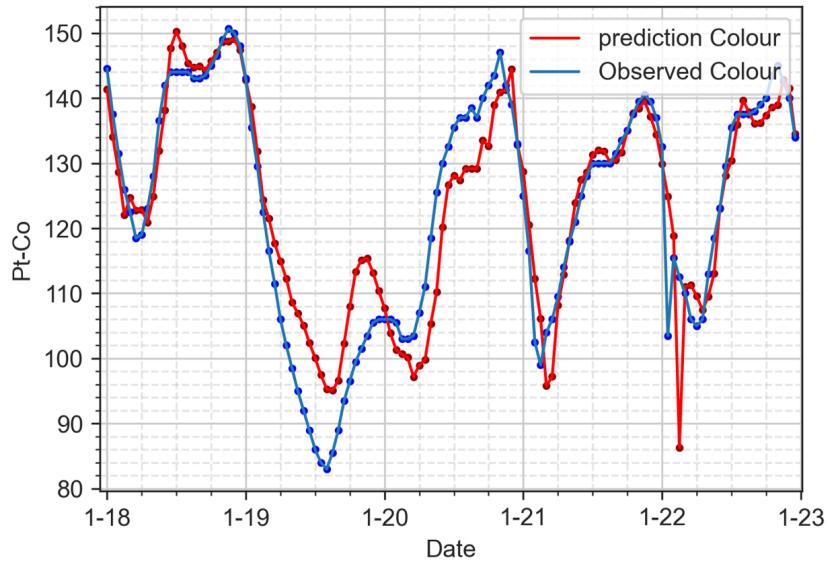
(b) LSTM-3-sg9, MSE = 17.5955, R-squared = 0.9461

Figure 2.12: Visualization of the colour forecasting models at forecast horizon of one.

and 103.4329 in forecasting colour levels at forecast horizons of three. We first noticed that both the models failed to forecast the lowest colour levels on 19 January. The significant drop in colour level can be a rare event in which the model did not learn how to react to such a change of colour levels from historical data. On the following days of 20 January, both the models underestimated the colour levels by forecasting up to 20 Hazen Units lower. The model performance deteriorated even faster than using ammonia forecasting models to forecast ammonia concentration at a forecast horizon of three. The results suggest that with much strong fluctuation of colour levels during the day, it is not reasonable to use colour forecasting models



(a) LSTM-1-obs, MSE = 62.6678, R-squared = 0.8053

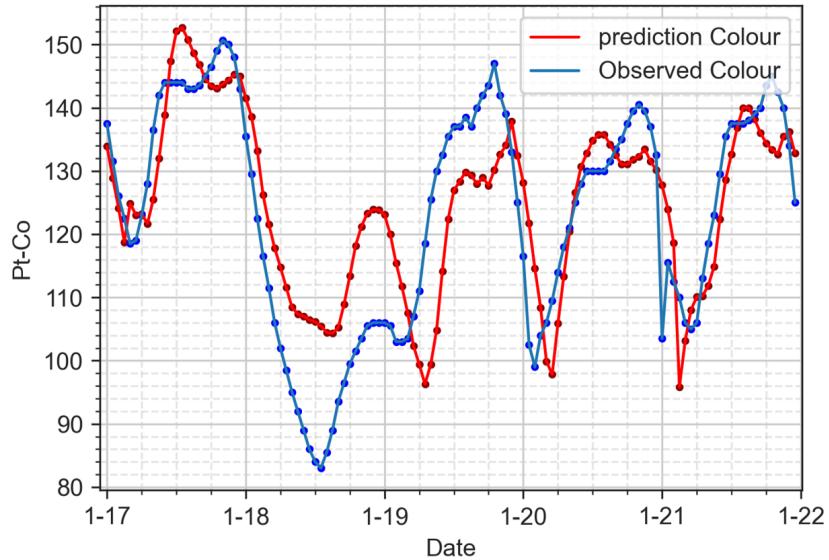


(b) LSTM-3-sg9, MSE = 47.4252, R-squared = 0.8526

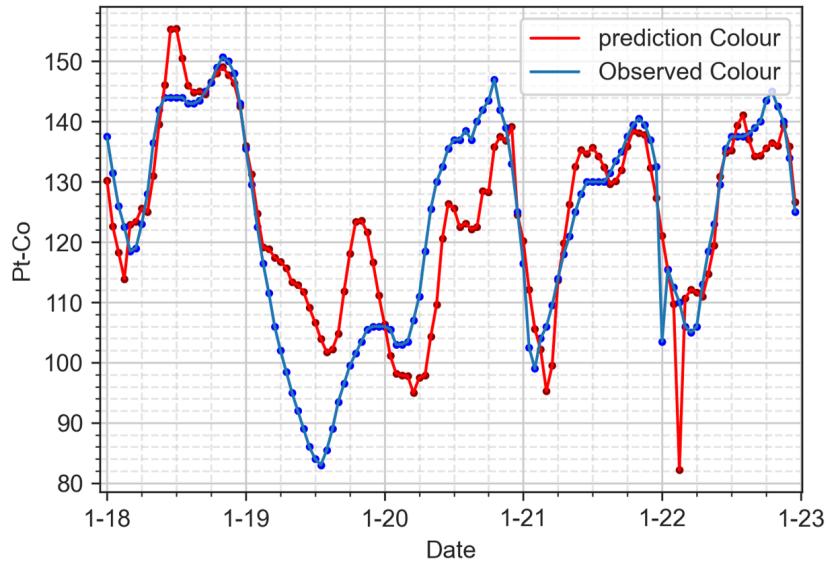
Figure 2.13: Visualization of the colour forecasting models at forecast horizon of two.

trained with only three input features to forecast three hours into the future.

One of the underlying causes of inaccurate forecasting results in longer forecast horizons for both forecasting NH<sub>3</sub>-N concentration and colour levels can be attributed to the physical limitations of the MBR effluent container. When the effluent was sampled in the container, the NH<sub>3</sub>-N and colour substances in the effluent container were not always evenly dispersed throughout the water, resulting in heterogeneity in the effluent container. Thus, the forecasted NH<sub>3</sub>-N concentration and colour levels can deviate from the true values. Another cause of the inaccuracy is derived from how the model inferences the forecasting values. As shown



(a) LSTM-1-obs, MSE = 116.8928, R-squared = 0.6237



(b) LSTM-3-sg9, MSE = 103.4329, R-squared = 0.6750

Figure 2.14: Visualization of the colour forecasting models at forecast horizon of three.

in Fig. 1.19b, the forecast horizon of two was inferred from 23 hours of the observed value and one hour of the forecasted value. Therefore, if the first forecasted value was not with pinpoint accuracy compared to the true values, the second forecasted value, which is inferred from the model based on the input of the previous forecasted value, will only become much more inaccurate. The error will be magnified with the increase in the forecast horizon.

The measurement technology of water quality can also significantly impact the accuracy of model forecasting in the long forecast horizon. In this study, we found models are more capable of forecasting colour levels for one to three hours into the future compared to NH<sub>3</sub>-N forecasting

models. The measurement technology of on-line colour is based on the spectrophotometer, which is a reliable way to measure colour levels in the water. However, the technology used for NH<sub>3</sub>-N data collection is an ion-selective electrode sensor, which is a less robust way to measure NH<sub>3</sub>-N compared to the conventional wet-analyzer. The NH<sub>3</sub>-N sensor inaccuracy can be caused by the interference from other ions, the effect of the ionic strength of the solution, and potential drift during a sequence of measurements. The physical constraints of the NH<sub>3</sub>-N sensor have made the data quality inherently poor, resulting in limited model forecasting accuracy.

## CHAPTER 3

# CONCLUSIONS AND RECOMMENDATIONS

### 3.1 Conclusions

#### 3.1.1 Machine learning models versus deep learning models

The selection of using which machine learning and deep learning models was not widely discussed to the best of our knowledge in modelling forecasting models in the wastewater treatment industry. This study has investigated the model performance of the machine learning model of RF and four other deep learning models of DNN, RNN, GRU, and LSTM on forecasting ammonia concentrations and colour levels in the reclaimed water system for assisting treatment operation and management. The evidence from this study suggested deep learning models are much capable of learning from historical data and generating more accurate forecasting results. In both ammonia and colour forecasting models, the test loss values of RF are much higher than those of the least-performance deep learning model of DNN. Among all the deep learning models, the results indicate that LSTM and GRU models have the lowest test loss of 0.0405 and 0.0414, respectively. However, further research works suggest that LSTM models trained with pre-processing methods generate the lowest test loss compared to GRU, making the LSTM model the most promising recurrent neural network model for training forecasting models in WWTPs.

#### 3.1.2 Data pre-processing techniques

Our research also highlighted how the model performance could be improved by applying data pre-processing and feature engineering techniques. Generally speaking, all the proposed data smoothing and outlier removal methods reduced the test loss values compared to the baseline model performance (i.e., the window sizes of the data smoothing filters need to be carefully selected), as shown in Fig. 2.6. Ammonia and colour forecasting models trained by EWMA filtered datasets showed the lowest test loss values compared with models trained by SG filtered datasets and datasets applied with outlier removal methods. Applying an EWMA filter

on training datasets can reduce the noise and allow the important patterns to stand out more clearly. The information hidden in the convoluted data points then can be further captured by the memorizing cells in the recurrent neural networks such as GRU and LSTM.

### 3.1.3 Feature engineering techniques

This study is the first step towards enhancing our understanding of the potential benefits of using created features for model training. The thorough examinations of the Geomap near the SHWEPP and the investigation of water composition in the public sewage system helped us hypothesize that the change of ammonia concentrations and colour levels depend on each other. With the help of an additional colour/ammonia feature for the ammonia/colour forecasting models, the test loss was reduced by 6.4% (i.e., LSTM-2-sg7 compared to LSTM-1-obs) and 10.8% (i.e., LSTM-2-ew4 compared to LSTM-1-obs), respectively.

Moreover, the similarity between the household consumption patterns and the daily fluctuation of ammonia concentrations have unexpectedly helped us formulate the time features via positional encodings. The influence of the sine and cosine hour features on the model performance showed tremendous improvements in both ammonia and colour forecasting models. In the former, test loss dropped by 8.9% (i.e., LSTM-1-obs compared with LSTM-4-sg7) while the latter reduced by 28.6% (i.e., LSTM-1-obs compared with LSTM-3-sg9). The remarkable use of positional encoding features is that they are not limited to ammonia and colour forecasting models. Any time-series data characterized by daily fluctuation patterns can adopt the use of the features of sine and cosine hour as long as the patterns are based on actual events. In addition, the positional encoding features are not limited to the hour component, we can encode time component features from seconds to weeks, and even years, the application of it is unlimited. However, the feature engineering method comes with limitations. In the results of ammonia forecasting models, LSTM-2-obs, LSTM-3-obs, and LSTM-4-obs showed higher test loss compared to LSTM-1-obs, indicating that when the models were not trained with ammonia feature only, the model performance worsened. Our results suggested that feature engineering needs to be carefully evaluated and experimented with before its real application. Despite the limitations, the combination use of feature engineering in building ammonia and colour forecasting models in this study has fully proved its advantages.

### **3.2 Recommendations for future research**

Due to the insufficient amount of ammonia and colour data, we cannot ensure the proposed methods in this study can also be applied to datasets of larger size and with more complex daily patterns of water quality. Therefore, it is recommended to collect a larger dataset (i.e., data collected from a longer time period) and a dataset including a variety of fluctuation patterns to support the findings of the results in this study. There are three concerns when training forecasting models with insufficient dataset size. First, a small dataset has simple and regular patterns of daily fluctuation in both NH<sub>3</sub>-N and colour datasets. The patterns were similar on a daily basis, and days with irregular patterns were not observed. With such an easy-to-predict pattern, we could not demonstrate the advantages of using deep learning models and data enhancement techniques to forecast water quality. Second, we cannot conclude how the size of the dataset can affect the model performance. The sensitivity analysis was commonly practiced for finding the dataset size influence on model training. It is possible to increase the model performance simply by training on a larger dataset. Third, insufficient data could also lead to unstable performance when the same data smoothing techniques were applied for training different models. For instance, models trained by sg7 filtered dataset (LSTM-4-sg7 and LSTM-3-sg7) have the lowest test loss values; however, LSTM-2-ew4 has a lower test loss than LSTM-4-sg7. We failed to explain why models trained by the sg7 filtered dataset influenced ammonia forecasting models differently among LSTM-2, LSTM-3, and LSTM-4. It is necessary to elucidate the influence of each data pre-processing technique to establish robust strategies for smoothing the training datasets.

More focus should be shifted to the fouling behavior and the accuracy of the on-line sensors in the wastewater treatment plant. Despite the fact that the development of on-line sensors is mature for providing on-line data in short time intervals (e.g., every minute), the fouling behavior due to microorganisms and the accuracy of the on-line sensors subject to the measurement technology (i.e., ion-selective exchange method for measuring NH<sub>3</sub>-N is less accurate than wet-analyzer) are major concerns. We have noticed only mild NH<sub>3</sub>-N sensor fouling behaviors were found and did not influence the collected data quality. In future studies, the data collected from on-line sensors should be carefully analyzed to identify how much the sensor fouling behavior can potentially affect our model performance. Meanwhile, more on-line sensors should be installed in the treatment process, such as turbidity and E. coli. Installing multiple on-line sensors will allow us to develop models which are more accurate. In addition, data collected from

sensors will allow us to build a soft sensor, which can help identify the fouling behavior of the on-line sensors.

Previous research studies have demonstrated using Matlab-Simulink to simulate the improved process control strategies using machine learning model controls compared to PID or other traditional mathematical models. In future works, the study will explore writing the physical and operational characteristics of the water reclaimed system into the Matlab-Simulink. By implementing the models developed in this study on Matlab, we can investigate how the improvements in model forecasting accuracy can help the process control strategy in stabilizing the reclaimed water quality. Several metrics can be used to evaluate the machine learning model control, such as the required time to reach set-point conditions and how much reclaimed water in volume we can generate from the same amount of wastewater effluent recycled.

## References

Halidu Abu-Bakar, Leon Williams, and Stephen H. Hallett. Quantifying the impact of the COVID-19 lockdown on household water consumption patterns in England. *npj Clean Water*, 4(1):13, December 2021. ISSN 2059-7037. doi: 10.1038/s41545-021-00103-8.

Samarth Agrawal. Hyperparameters in Deep Learning, February 2019.

Bangaloreai. Deep neural network (DNN) is an artificial neural network (ANN), March 2018.

CFI. Exponentially Weighted Moving Average (EWMA), January 2022.

Tuoyuan Cheng, Fouzi Harrou, Farid Kadri, Ying Sun, and Torove Leiknes. Forecasting of wastewater treatment plant key features using deep learning-based models: A case study. *IEEE Access*, 8:184475–184485, 2020. doi: 10.1109/ACCESS.2020.3030820.

DeepAI. Loss Function, June 2022.

Niklas Donges. A Guide to RNN: Understanding Recurrent Neural Networks and LSTM Networks, July 2021.

IBM. Neural Networks, June 2022.

Le, Ho, Lee, and Jung. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. *Water*, 11(7):1387, July 2019. ISSN 2073-4441. doi: 10.3390/w11071387.

Jie Liu. Time Series Forecasting 101 – Part 2. Forecast COVID-19 daily new confirmed cases with Exponential Smoothing Forecast and Forest-based Forecast, July 2020.

Behrooz Mamandipoor, Mahshid Majd, Seyedmostafa Sheikhalishahi, Claudio Modena, and Venet Osmani. Monitoring and detecting faults in wastewater treatment plants using deep learning. *Environmental Monitoring and Assessment*, 192(2), 2020. doi: 10.1007/s10661-020-8064-1.

MathWorks. Documentation-Findpeaks, June 2022.

Masoud Mohseni-Dargah, Zahra Falahati, Bahareh Dabirmanesh, Parisa Nasrollahi, and Khosro Khajeh. Chapter 12 - Machine learning in surface plasmon resonance for environmental monitoring. In Mohsen Asadnia, Amir Razmjou, and Amin Beheshti, editors, *Artificial Intelligence and Data Science in Environmental Sensing*, Cognitive Data Science in Sustainable Computing, pages 269–298. Academic Press, January 2022. ISBN 978-0-323-90508-4. doi: 10.1016/B978-0-323-90508-4.00012-5.

National Center for Biotechnology Information. "PubChem Compound Summary for CID 222, Ammonia" PubChem, June 2022.

Christopher Olah. Understanding LSTM Networks, August 2015.

Janosh Riebesell. Random Forest, June 2022.

Jie Fu Ritchie Ng. Deep Learning Wizard. Zenodo, April 2019.

Abraham. Savitzky and M. J. E. Golay. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Analytical Chemistry*, 36(8):1627–1639, July 1964. ISSN 0003-2700. doi: 10.1021/ac60214a047.

Cees Taal. Smoothing your data with polynomial fitting: A signal processing perspective, April 2017.

Dong Wang, Sven Thunéll, Ulrika Lindberg, Lili Jiang, Johan Trygg, Mats Tysklind, and Nabil Souihhi. A machine learning framework to improve effluent quality control in wastewater treatment plants. *Science of The Total Environment*, 784:147138, August 2021. ISSN 00489697. doi: 10.1016/j.scitotenv.2021.147138.

Wikipedia. Random forest, June 2022a.

Wikipedia. Savitzky–Golay filter, June 2022b.

# APPENDIX A

## PYTHON CODES

### A.1 Python codes for machine learning models

Random Forest

```
model = RandomForestRegressor(n_estimators = 500)
```

Deep Neural Network

```
class model_MLP_1(torch.nn.Module):
    def __init__(self, n_input=1, n_hidden=10,
                 n_batch=1, n_output=1):
        super(model_MLP_1, self).__init__()
        self.input_size = n_input
        self.hidden_size = n_hidden
        self.batch_size = n_batch
        self.output_size = n_output
        self.fc1 = nn.Linear(self.input_size, self.hidden_size)
        self.relu = nn.ReLU()
        self.fc2 = nn.Linear(self.hidden_size, self.output_size)

    def forward(self, src, device):
        output = self.fc1(src[:, :, 0])
        output = self.relu(output)
        output = self.fc2(output)
        return output
```

Recurrent Neural Network

```
class RNN(nn.Module):
    def __init__(self, input_size=1, hidden_size=10,
                 num_layers=1, output_size=1):
        super(RNN, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.output_size = output_size
        self.rnn = nn.RNN(input_size=input_size,
                          hidden_size=hidden_size,
                          num_layers=num_layers)
```

```

    self.fc = nn.Linear(hidden_size, output_size)

def forward(self, src, device):
    h_t = torch.zeros(self.num_layers, 1, self.hidden_size)
    out, _ = self.rnn(src[:, :, 0], h_t)
    out = self.fc(out)
    return out

```

### Gated Recurrent Unit

```

class GRU(nn.Module):
    def __init__(self, input_size=1, hidden_size=10,
                 num_layers=1, output_size=1):
        super(GRU, self).__init__()
        self.input_size = input_size
        self.hidden_size = hidden_size
        self.num_layers = num_layers
        self.output_size = output_size
        self.gru = nn.GRU(input_size=input_size,
                          hidden_size=hidden_size,
                          num_layers=num_layers)
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, src, device):
        h_t = torch.zeros(self.num_layers, 1,
                         self.hidden_size)
        out, _ = self.gru(src[:, :, 0], h_t)
        out = self.fc(out)
        return out

```

### Long Short-Term Memory

```

class model_LSTM_1(nn.Module):
    def __init__(self, n_hidden=10):
        super(model_LSTM_1, self).__init__()
        self.n_hidden = n_hidden
        self.n_layers = 1
        self.lstm = nn.LSTM(input_size = 1,
                            hidden_size = self.n_hidden)
        self.linear = nn.Linear(self.n_hidden, 1)

    def forward(self, src, device):
        h_t = torch.zeros(self.n_layers, 1, self.n_hidden)
        c_t = torch.zeros(self.n_layers, 1, self.n_hidden)
        h_t, c_t = self.lstm(src[:, :, 0], (h_t, c_t))
        output = self.linear(h_t)
        return output

```