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 CASSOL, 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

LIST OF FIGURES

LIST OF TABLES

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 (NH_3-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, NH_3-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 NH_3-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 NH_3-N levels. To tackle this challenge, this thesis work developed time-series models based on machine learning to forecast the NH_3-N concentrations and colour levels in the reclaimed water three hours into the future. For the training dataset, the NH_3-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_3-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_3-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_3-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_3-N and colour have correlations with the urban water consumption patterns. This finding further enhanced the NH_3-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_3-N concentrations and colour levels forecasting in reclaimed water for treatment optimization.

CHAPTER 1

INTRODUCTION

1.1 Background

Urban water challenge increases as the cities grow larger. The World Bank estimates that the urban population worldwide will double by 2050—with severe implications of escalating water demands in cities by 50–70 percent (TheWorldBank, 2021). Global climate change has primarily affected the amount, distribution, and quality of the available fresh water in the urban water cycle. The report from (UNICEF, 2021) points out that one in four cities is facing challenges in supplying adequate water to inhabitants, and the situation is even worse in cities in the developing world. The rise of urban water usage will generate more wastewater. Thus, converting municipal/industrial wastewater into reusable water has recently drawn much attention. Reuse water increases availability by substituting freshwater for non-potable (drinkable) uses for agricultural irrigation, industrial and urban water reuse, etc. The alternative reuse water can supply many activities and save drinking water for other purposes elsewhere (Adewumi et al., 2010).

The construction of reclaimed water facilities often requires a huge amount of capital investment. Upgrading available wastewater treatment plants with reuse water treatment facilities is an economical solution accompanied by the potential of realizing resource recovery (e.g., nitrogen and phosphorus recovery) (Maryam and Büyükgüngör, 2019; Kehrein et al., 2020). The primary concern of reusing treated wastewater is the potential risks caused to public health. Under unexpected circumstances, the reclaimed water facilities can produce unqualified reclaimed water, which is harmful to the living beings (i.e., as reused water is ingested directly or through irrigated crops) and irrigated soil (Adewumi et al., 2010). In Hong Kong, reclaimed water quality is regulated with up to 10 or more water quality parameters, and any parameters that fail to meet the standard will lead to disqualification. The common practice for controlling the treated water quality is achieved through water quality control strategies. The market controllers have evolved from a simple on-off logic controller called Programmable Logic Controller (PLC), to a more advanced multi-step response controller called proportional-integral-derivative (PID), and finally to the controller consists of machine learning models.

The uses of machine learning models in the water quality controllers for assisting water quality control strategy are ground-breaking applications. Many research papers have proposed various machine learning models for replacing the PLC and PID controllers and demonstrated the benefits of machine learning models. From the study of (Librantz et al., 2018), PID and machine learning-based controllers were deployed to compare the operational costs of dosing the chlorine to the setpoint concentration in a drinking water treatment plant. The results showed that the Artificial Neural Network-based model has a more satisfactory cost reduction in a chlorination dosing control system than the PID controller. Another research finding suggests using a Support Vector Regression (SVR) model as the controller required less time to reach the setpoint concentration of free chlorine residual compared to the PID controller in both simulation and experimental conditions Wang et al. (2020). Incorporating machine learning models in traditional process control systems has also been practiced by Santín et al. (2015) for avoiding violations of total nitrogen in the effluent using the decisions made by Artificial Neural Networks. Long Short-Term Model was also used to predict which process control strategy should be selected for eliminating violations of total nitrogen concentrations in the effluent Pisa et al. (2019). Forecasting water quality or predicting future events using machine learning are proved to be effective measures for controlling effluent water quality in wastewater treatment plants, making these approaches to be promising solutions for the reclaimed water treatment operation and management.

The superior performance of machine learning models comes from training high-quality datasets with a good amount of data that can fairly represent the system's dynamics. Most studies have only focused on evaluating the model performance by comparing the test loss values between models and the improvements over PID controllers without considering the collected dataset's quality. The noises in the data and the number of features (i.e., inputs or variables) are the two critical factors affecting machine learning models' accuracy and robustness. Many data pre-processing techniques are proposed and applied to enhance the dataset's quality by removing the data noise. For instance, some papers discussed pre-processed data for removing the noise in raw datasets using data smoothing filters (Cheng et al., 2020), or creating new features in addition to the original ones (Mamandipoor et al., 2020) to achieve data augmentation. Despite the efforts being made, the influences of the proposed data pre-processing techniques on the model performance have not yet been established.

Machine learning models for water quality control have two main types of algorithms, regression and classification. The former provides forecasting results of specific values, while the

latter offers a decision of yes or no (i.e., 1 or 0). The regression model is also called the forecasting model, which plays a vital role in water quality control in drinking water treatment plants (DWTPs) and wastewater treatment plants (WWTPs). The forecasted results can be effectively used to provide critical information for the water quality control strategy. The need to use forecasting models is to cope with the unpredictable nature of water quality and to plan control strategies ahead. Without future information, the treatment operations are less likely to guarantee the production of effluent quality satisfying the government regulation Chen et al. (2003) regardless of how the influent water quality may vary daily. In the reclaimed water system in Shek Wu Hui Effluent Polish Plant (SWHEPP), forecasting models are recommended for effluent treatment management and operation. From the available datasets, we noticed SHWEPP effluent contains NH_3-N concentrations and colour levels which exceed the reclaimed water standard. To generate non-potable resue water, it is critical to use on-line data to assist water quality control strategy. Currently, the available on-line sensors on-site are limited. Although the model can only train on limited data, it is still possible to train forecasting models with one feature, known as the univariate forecasting model. In this study, we will attempt to install one more on-line sensor and build machine learning models for forecasting NH_3-N concentrations and colour levels in the reclaimed water system. Meanwhile, data pre-processing and feature engineering techniques will be proposed and evaluated to address the research gaps of insufficient understanding of data pre-processing in building forecasting models in the reclaimed water system.

1.2 Objectives

The specific objectives of this thesis work are:
(1) To build baseline univariate NH_3-N and colour forecasting models using machine learning and deep learning models.
(2) To develop data pre-processing techniques for removing data noise to enhance model performance.
(3) To extract relevant information from the reclaimed water system using domain knowledge for applying feature engineering techniques.
(4) To create new features to augment the dataset's quality to improve forecasting model performance.

1.3 Organization of the thesis

In Chapter 1, "Introduction," the background information, objectives, and organization of the thesis were presented.

Chapter 2, "Literature Review", provides an overview of water quality control strategies in water treatment plants, wastewater treatment plants, and reclaimed water systems.

In Chapter 3, "Materials and Methods," the instruments for data collection of NH_3-N concentrations and colour levels, computer programming environment, and data preparation techniques were summarized. The processes of formulating extra features for training forecasting models were illustrated.

In Chapter 4, "Results and discussion," the performance of machine learning and deep learning models were compared. Forecasting models trained by different data pre-processing techniques and the influences of feature engineering on model performance were compared with the baseline model performance in test loss.

In Chapter 5, "Conclusions and Recommendations," the findings obtained from this thesis work were summarized, and possible future studies were recommended.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction to water quality control

2.1.1 Automated system for water quality control

A programmable logic controller (PLC) is an industrial computer system designed for any process requiring a series of devices and equipment to operate cohesively to achieve multiple purposes in manufacturing or treatment processes. The main components of PLC include a central process unit (CPU), input modules, and output modules (I/O). CPU is responsible for processing digital signals from input modules and sending commands through output modules based on the control logic programmed on the PLC. For chemical dosing control in water treatment plants (WTPs), the PLC system receives readings from turbidity and pH sensors and uses pumps to dose aluminum solution automatically (Andhare and Palkar, 2014). The PLC system with the capability of producing real-time output commands in response to the input signals also makes it widely used in wastewater treatment plants (WWTPs). For oxygen concentration control in the aeration tank, the PLC system receives signals from dissolved oxygen (DO) detectors and transmits signals to open or close the electric butterfly valves to alter the DO concentration (Zhu and Qiu, 2017). Although PLC systems are the most used systems across industries for their easy programming and reliable control, PLC system is merely a device that can be programmed to control operative devices with on-off logic (i.e., a logic control with two states). The straightforward implementation of the PLC system compromised its ability to perform complex tasks in a more dynamic water treatment environment. In reality, many WTPs or WWTPs require precise control of the treatment processes. Being aware of the limitations of the PLC systems, a more advanced controller called proportional–integral–derivative (PID) controller for receiving analog signals was developed to obtain more sophisticated controls over the operative devices.

To react to rapidly-changing environments in wastewater treatment plants, a PID controller generates an output value based on the continuous calculation of an error value e(t), which is the difference between the desired setpoint (SP) and a measured process variable. Then,

the controller applies a correction based on proportional, integral, and derivative terms in the control functions. The use of the ”P,” ”I,” and ”D” allows the system to quickly reach steady-states with feedback control systems (i.e., the system output is returned to the system input, which is included in the decision-making process of PID controller). Generally speaking, a PID controller is a technology (i.e., a specialist algorithm) for controlling a single device with more complex logic, while a PLC system is a physical system consisting of different modules capable of controlling dozens of devices only with two-state logic. In addition, A PID controller can be designed to operate on a PLC device and provide a more specific control strategy to a designated device. In WWTPs, a single-variable feedback analog control loop in PID can be used to control the temperature in the activated sludge treatment by stabilizing the system temperature in a shorter time (Bados and Morejon, 2020). The feedback control scheme is also applied in DWTPs to adjust the addition of chlorine dosage (i.e., also known as the disinfection process, chlorination, or post-chlorination) to reach the target concentration of free chlorine residual (FRC) (Wang and Xiang, 2019). The disinfection process is carried out in a chlorine contact tank, which provides sufficient time for the chlorine to disinfect pollutants. Since the chlorine added by the dosing device requires time to travel from the entry to the exit, the system output can only reflect the changes in water quality in a delayed time of 30 minutes. In the case of chlorination, the time lag makes feedback control difficult (Kobylinski et al., 2006) as the system is delayed in responding to any sudden surge of the pollutants when it can only receive output at the end of the disinfection process. PID controllers in WWTPs also encounter similar challenges as the increasing complexity of water quality and stricter regulations on the discharged water quality.

Many control strategies are proposed to address the challenges encountered in the process control system. For instance, feed forward-feedback control, linearized and optimal control, model-predictive control, fuzzy control (Demir and Woo, 2014), etc. Among the algorithms used in control strategies, Artificial Intelligence (AI) modeling has gained the most attention in recent years compared to modeling based on mathematical or empirical formulas. In DWTPs or WWTPs, fully understanding the treatment plants' physical, biological, and chemical interactions is very difficult. The unpredictable behaviors during the water treatment can be the significant changes in influent flow rate, water quality fluctuations, the complexity of the biological treatment process, and the large time delay between control variables and the process inputs, etc. Therefore, AI modeling shows great potential in dealing with the highly complex conditions in the treatment process (Li et al., 2021). The next sections will discuss the applica-

tions of different AI modeling methods.

2.1.2 Artificial Intelligence

Artificial intelligence (AI) can perform cognitive tasks with the development of computational solutions. The concepts of AI are usually confused, and in fact, AI is a broad term, and any kind of algorithms or models involved in decision-making with computation fall in the domain of AI. For example, AI can be fuzzy logic and optimization algorithms, which are formulated with human design and involved in the computer decision-making processes. Another subset of AI is called machine learning (ML), but generating an ML model is different from generating a fuzzy logic model. ML uses learning algorithms to generate a model via learning from the historical or large amount of data without being explicitly programmed. ML algorithms can be classified into three categories, which are Supervised, Unsupervised, and Reinforcement learning. In the training process of supervised learning, input variables (x) and output variables (Y) will be provided. The model will learn from the provided datasets to map the x to the Y. A supervised model can generate a prediction based on the new input data (i.e., also called the unseen data). Unsupervised learning is when the dataset is not labeled, the model can learn to infer patterns in the dataset without reference to the known outputs. This type of algorithm can find similarities and differences in the data. In reinforcement learning, models are designed to constantly interact with the environment in a try-and-error way and receive rewards and punishments based on the purpose of the tasks. Generating an optimal action to achieve the lowest penalties is the primary function of a reinforcement learning model. Supervised learning is commonly used for machine learning in water quality control strategies. Regression is a supervised machine learning technique used to predict continuous values. A regression model can estimate the relationship between the input variables in the system and the output target from given datasets and then use the nonlinear relationship to map the unseen input data to predicted output data. This type of applications best fits for water quality prediction (Librantz et al., 2018), and sensor fault detection (Cecconi and Rosso, 2021), etc.

2.1.3 Machine learning and deep learning

In machine learning, popular models which researchers frequently use for training predictive models are Supporting Vector Machine (SVM), Random Forest (RF), and Artificial Neural Networks (ANN). RF models are popular due to their superior performance compared to typical

machine learning algorithms. Xu et al. (2021) built an RF-based model to predict total nitrogen concentration in water bodies and proved RF models outperformed models such as K nearest neighbor (KNN), Ridge Regression, and Multilayer Perceptron (MLP). The other two widely used models, ANN and SVM, were compared carefully with the reliability and accuracy in predicting 1-day interval T-N concentration in a WWTP (Guo et al., 2015). The results showed that the SVM model has higher accuracy while the ANN model is more reliable for decision-making. Although most of the studies did not focus on the underlying causes of why SVM, RF, and ANN models have more excellent model performance, it would still seem that these models are reliable options for predicting water quality empirically.

As the computing power doubles every 18 months according to Moore's law, implementing Deep Learning (DL)—a subset of machine learning, requires less and less computing time and becomes universal for researchers to solve everyday tasks. One way to explain a DL model is with the definition of having neural networks with more than two hidden layers (i.e., the model complexity increased and required more computing power to calculate). In DL, various architectures are specifically structured based on the problems we attempt to solve. For image processing, Convolutional Neural Network (CNN) is designed to extract essential features from the image vectors. Another famous DL architecture is the Recurrent Neural Network (RNN), which is powerful in solving time series-related applications and Natural Language Processing (NLP) tasks (Li et al., 2018). In particular cases, different DL architectures can be stacked in series to solve specific tasks. A rainfall-runoff prediction model was built using CNN and RNN (Li et al., 2022). The raw data features were extracted by convolution and entered into the RNN models for processing time-series patterns. The results showed a low Kling–Gupta efficiency (KGE) of 0.75 in the high-water period. DL models can be compelling when multiple architectures are stacked into a single model to perform a specific task, which machine learning models cannot realize. That being said, DL models can achieve higher model performance in terms of prediction accuracy compared to ML models.

2.2 Water quality control with machine learning

2.2.1 Drinking water treatment plants

A drinking water treatment plant (DWTP) produces potable (i.e., drinking water) water for human consumption by removing contaminants from the source water, such as lakes or streams,

or from underground aquifers. The raw water enters DWTPs and goes through treatment units of coagulation, flocculation, sedimentation, filtration, and disinfection in sequence as the primary treatment scheme in the conventional DWTPs (Li et al., 2021). During the treatment process, colloids, suspended matter, pathogenic microorganisms, and organic matter are removed to meet the regulated standard. However, raw water quality is not always stable, and corresponding actions must be promptly adopted when events like the surge of pollutants or the large variability of the influent flow. In any event, the treated water from DWTPs should generate drinking water that complies with the World Health Organization's Guidelines (i.e., WHO guideline) for drinking water quality. Otherwise, the treated drinking water would either be discharged, resulting in the short-term outage of water supply to the downstream cities; or the users will receive contaminated drinking water, which can transmit diseases and cause illness.

Turbidity is one of the critical water quality indicators, which can be defined as the "optical quality" of water. The unit describing the turbidity is the Nephelometric Turbidity Unit (NTU). High turbidity levels in raw water can impede the effectiveness of filtration and chlorination processes and potentially cause short-term outages of the water supply. Heavy rainfall and fissures within the aquifer can also lead to turbidity events that are most likely to cause high turbidity (World Health Organization, 2017). The challenge in the event of high turbidity in raw water is that it occurs rapidly, and mitigating activities must be actionable immediately. To address the sudden event of such, Stevenson and Bravo (2019) trained forecasting models based on general linear model (GLM) and RF to predict the time when the turbidity reaches higher than 7 NTU. The results indicate that both models can successfully predict the events (i.e., with accuracy between 0.81 and 0.86), and the RF model is found to have higher precision due to its ability to capture the nonlinear relationship between rainfall (mm) and turbidity (NTU).

To maintain operational costs and water quality in the coagulation process, the amount of coagulant, mainly subject to the turbidity and alkalinity in the raw water, is traditionally determined through manual sampling and analysis. Jar tests are designed to find the optimal chemical dosage for coagulation to remove the turbidity in raw water. The entire process includes on-site sampling and more than 40 minutes of laboratory work (Gani et al., 2017). To replace the laborious jar tests, Wang et al. (2022) proposed using principal component regression (PCR), support vector regression (SVR), and Long Short-Term Memory (LSTM) neural network to build predictive models for estimating daily chemical dosage. Compared with the linear PCR model, nonlinear SVR and LSTM models capture more relationships between the chemical dose (e.g., ferric sulfate) and the raw water quality based on a higher R-squared value

of 0.70.

Disinfection is the last step of water treatment processes in drinking water treatment plants to generate safe potable water. In this step, chemical disinfectants such as chlorine, chloramine, or chlorine dioxide are added into the water to inactivate any remaining pathogenic microorganisms. However, the chlorination process requires precise dosing of disinfectant—too high will lead to the formation of disinfection byproducts (DBPs), and too low will result in insufficient levels of the residual disinfectant concentration. In both scenarios, the treated drinking water can pose health threats to the end-users. Although the PID controller can achieve automatic dosing of disinfectants according to the change in water quality, Wang et al. (2020) proposed a model predictive control based on machine learning models to improve the dosing process further. The study indicated that the predicted chlorine dosage from a Support Vector Regression (SVR) model could help the free chlorine residual in the water reach the setpoint concentration in a shorter time compared to the PID controller in both simulations and experimental conditions. Machine learning models can not only reduce the time required to reach setpoint concentration but also decrease the chemical usage required in DWTPs. An Artificial Neural Network-based model has proved to optimize the treatment operation by reducing the chemical usage in a chlorination dosing control system compared to using PID controller (Librantz et al., 2018).

The invariability of the raw water quality is always a big issue for disinfection. For instance, chlorine dose can be excessively dosed when the treated water contains fewer pollutants (e.g., non-organic matters and ammonia nitrogen). Exessive chlorine in water results in the waste of chemicals, which is reflected in the increased operational cost and the generations of undesired disinfection by-products (e.g., trihalomethanes (THMs), which are carcinogenic to humans). Xu et al. (2022) trained an ANN model for predicting the occurrence of THMs in tap water using simple and straightforward water quality parameters (e.g., pH, temperature, UVA_{254} and residual chlorine (Cl_2)). Despite the fact that the results showed a good model accuracy in predicting for THMs (i.e., T-THMs, TCM, and BDCM), the applications of the model are largely limited in reality due to the lack of dataset regarding quantity and quality. In fact, the lack of high-quality datasets for training ML models is a common issue, which explains, until recently, mathematical or empirical-based AI models like fuzzy logic (Gamiz et al., 2020; Godo-Pla et al., 2021) is still widely used for process control in WTPs.

2.2.2 Wastewater treatment plants

Human activities produce wastewater and discharge it from homes, businesses, factories, and commercial activities to the sewage systems which connect to wastewater treatment plants (WWTPs). The function of WWTPs is to remove contaminants from sewage and water so that the treated water can be returned to the natural water body without dangering any living beings residing in the ecosystem. Undertreated wastewater can lead to harmful algal blooms or cause oxygen deficit in the water (i.e., low oxygen content can kill the fish). The steps for treating municipal wastewater involve three major categories—primary treatment, secondary treatment, and tertiary treatment. Most of the particular matters will be removed in primary treatment via settling or floating; a secondary treatment is mainly responsible for removing BOD_5 in the biological processes; in the final tertiary treatment, membrane filtration, adsorption by activated carbon, and addition of disinfectant are applied optionally to further eliminate the undesired pollutants in the water.

Wastewater is categorized and defined according to its sources of origin. Domestic wastewater is water discharged from residential sources generated by kitchen wastewater, cleaning, and personal hygiene. Industrial/commercial wastewater is generated and discharged from manufacturing and commercial activities, such as the textile industry and food and beverage processing wastewater. Institutional wastewater is generated by large institutions such as hospitals and educational facilities. Regardless of the source of the wastewater, WWTPs have to achieve at least three sustainability targets: environmental protection (i.e., minimum pollutants discharge), social acceptance (i.e., human sanitary protection), and economic development (i.e., feasible operational and management costs) (Mannina et al., 2019). To effectively achieve these goals, process control is required to reduce energy consumption, improve effluent quality, and save costs in plant operation and management. The focus of this study is on discussing the development of using process control for treatment operation and management.

Under known operational conditions of a WWTP, machine learning models can be trained to assist the plant operators in optimizing treatment processes to improve effluent quality. Wang et al. (2021) proposed a machine learning framework, utilizing a model based on Random Forest to predict the effluent Total Suspended Solid (TSS) and phosphate (PO_4). This study uses data from six on-line sensors (i.e., flow rate, TSS, pH, PO_4, temperature, and total solids (TS) meters) across the treatment line to train the RF model. The results indicated that the influent temperature is the most influential variable for both TSS and PO_4 in the effluent, and PO_4

depends strongly on the TSS in aeration basins, etc. It has been suggested that the combined use of the RF model and analytical tools allows the author to pinpoint the critical factors influencing the effluent quality, which is regarded as an innovative approach. However, several significant drawbacks hinder such model developments using on-line sensors to collect training data. The term ”training data” is a dataset used to feed into the model for the model to learn and pick up hidden patterns in the data. Many of the existing WWTPs and DWTPs are not equipped with on-line sensors, and a lack of automation and instrumentation is universal. The difficulties in installing on-line sensors include the extra costs of purchasing hardware, extra labor works for maintenance, and most importantly, the optimal locations for sensor installation.

In secondary treatment, the relationships between the sludge and wastewater quality are complex due to the complex interactions between the microorganisms and the organic matters in the reactor (Wilén et al., 2018). To fully understand and describe the interactions in such systems requires a substantial amount of data. However, installing on-line sensors everywhere in the system is impossible. Zaghloul et al. (2021) attempted to find out the ideal locations and adequate number for on-line sensor installation. The author used the data collected from the on-line sensors installed in three lab-scaled secondary treatment reactors to train machine learning models to predict effluent quality. In addition, considering the intricacy of operational conditions in the secondary treatment, the author claimed that with the use of feature selection and ensemble model (i.e., average results from multiple model outputs), overfitting could be prevented. The issue of overfitting can be understood as the model memorizing the noises too much in a training dataset, resulting the poor performance when the model is used to predict outputs from unseen data.

Similar to the secondary treatment units, an electrocoagulation reactor is also a complex system in which the operation and management are based on pH value, current density, flow rate, and the initial concentration of heavy metal ions, etc. Interestingly, instead of using an ensemble model to prevent the overfitting issue claimed by Zaghloul et al. (2021), Zhu et al. (2021) used a deep learning Long and Short-term model (LSTM) and an error compensate Autoregressive Integrated Moving Average model (ARIMA) to predict the removal rate of heavy metal ion concentration in wastewater. An LSTM-ARIMA model has strengthed the model performance compared to the solely used LSTM or ARIMA model in predicting removal rate shown by the Results. A possible rationalization of using an LSTM model without worrying about model overfitting is that deep learning is sophisticated enough for learning the nonlinear patterns in complex systems, while machine learning models like RF might fail to capture the

intricate relationships, resulting in overfitting.

Technological advancement allows easy access to real-time water quality data via on-line sensors. The collected real-time data can be used to train predictive models and assist the plant operation and management. Despite the advantages of what on-line sensors are capable of, sensor calibration and maintenance are critical. The malfunctioned sensor can induce wrong decisions for plant operation, ultimately deteriorating treatment efficiency in WWTPs. Haimi et al. (2015) suggested that reliable and moderately-priced on-line sensors are not always available; in addition, sensor malfunctions (i.e., fouling or erroneous measurement) can cause the down-time of the sensors. For the unavailable sensors (i.e., "hard-to-measure" or expensive sensors), many research works have proposed building "soft sensors." Instead of using hardware sensors to measure the water parameters, the soft sensor generates real-time values through a machine learning model, which is trained by other easy-to-measure water quality data. In the works of Wang et al. (2019), easy-to-measure variables such as pH, flow rate, TSS, and ammonium nitrate (NH_4-N) are input to machine learning models to predict hard-to-measure water quality parameters of COD and total phosphate (TP). Pattnaik et al. (2021) also used DO, pH, conductivity, turbidity, and temperature to train a model to predict BOD. It is believed that both research works can solve the issues of the unavailability of specific water quality sensors.

The automated treatment operation and management heavily rely on the reliability of the on-line sensors; thus, preventing and the early detection of sensors malfunctioned is the utmost concern to the plant operators. Sensor fault detections can be categorized into three groups which are (1) individual faults— outlier data that can be distinguished concerning other data points; (2) contextual faults—an anomalous instance in a specific context and normal in another; (3) collective faults—a cluster of rare instances with respect to other data trends (Chandola, 2009). Many research papers have proposed using machine learning models to help identify sensor fouling.

Two main algorithms, regression and classification, can be used to find fouling signals. A regression algorithm can identify fouling signals by calculating the difference between model-predicted outputs (e.g., ammonium or COD concentration) to the actual signals. A classification algorithm can distinguish fouling signals through the direct outputs of the model (i.e., the model outputs 2 class labels, one represents normal, and the other is abnormal). Cecconi and Rosso (2021) proposed an ammonium fault detection mechanism, utilizing a regression ANN model, along with principal component analysis (PCA) and Shewhart monitoring charts (i.e., statistical

control chart). The remarkable idea of this study is to analyze the residual between the predicted ammonium and the actual ammonium sensor signal and identify the individual and contextual faults with the help of statistical tools. Despite the fact that the accuracy of the fault detection mechanism can reach R^2 value of 0.87, the method comes with significant limitations. The author points out that to maintain the high accuracy of the predictive model, the quality of the input data needs to be carefully attended to by performing manual cleaning procedures on a weekly basis.

Research has focused on solving collective faults in sensor fault detection rather than collective faults. The primary reason is that collectives faults are hidden in regular signals, and the expert can only discover the irregularity by comparing sets of signals in sereis. Thus classification technique using deep learning is proposed to address collective faults in the works of Mamandipoor et al. (2020). It is believed that this is the first research paper using an LSTM network to achieve a fully automatic fault detection method in WWTPs. In contrast to other works, input variables for model training heavily rely on the experts' manual selection before inputting into models like PCA and fuzzy neural networks. The significance of using a deep learning network is its capability to capture long-term temporal dependencies from a large dataset compared to machine learning models (i.e., PCA-SVM model). The results showed that the accuracy (i.e., F1-score) from the LSTM model is 92%, outperforming the PCA-SVM model of 87%. This finding suggests that using DL models in classification problems is promising for solving collective faults.

2.2.3 Water reclamation system

The increasing demand for water in cities is mainly attributed to the rapid urbanization and the population moving from rural to urban centers. In many major cities, the evergrowing water usage and wastewater discharge drive the development of water reclamation (Lyu et al., 2016). In WWTPs, the technologies applied in water reuse include disinfecting with chlorine addition, ultra-violet (UV) irradiation, biological treatment, membrane filtration, etc (Norton-Brandão et al., 2013). However, even with the most advanced water treatment technology, the treated reclaimed water quality is still subject to the variability and variations of pollutant contents in wastewater effluent (Chen et al., 2003), and can potentially fail to meet the reclaimed water standard. The research studies propose to apply machine learning techniques to assist the disinfection process in water reclamation can be categorized into three groups (1) optimize

the treatment management in WWTPs to alleviate the loadings of water reclamation process (Al-Ghazawi and Alawneh, 2021; Viet et al., 2021); (2) actively branch out the desired, and undesired wastewater effluent for subsequent disinfection process of water reuse or direct disposal into water body (Chen et al., 2003); (3) adapt process control methods to stabilize the disinfection performance in the reclaimed water system (Demir and Woo, 2014).

Technology advancement and research studies on water reuse have been discussed for more than two decades. However, there are not too many research publications that aim to improve the reclaimed water system as a whole in recent years. The economic reasons behind constructing water reuse facilities could be a major obstacle for the government sectors. The economic burden of either building new reclaimed water institutions in new locations or retrofitting existing WWTPs is overwhelming (Adewumi et al., 2010). To discover more values and reusable resources from water reuse, Chojnacka et al. (2020) takes the circular economy perspective into accelerating the process of adopting water reuse systems for agriculture production. The author introduces the potential of gradually replacing chemical fertilizers with partially treated wastewater for sustainable crop production despite there are many limitations to be overcome. In Italy, the circular concept is also studied by Colella et al. (2021). Four different resource recovery senarios were brought up, and two of the senarios include the nutrient recovery turned into nitrogen and phosphorus fertilizers. Several researchers in recent years have provided the overall potential and challenges of treated wastewater reuse in the world; it is believed the day of using reused water universally will soon come with collaboration across different disciplines.

Reclaimed water for non-potable reuses can serve for irrigation for agriculture, toilet flushing, and irrigation for landscaping, etc. Water Supply Department (WSD) will soon implement a reclaimed water supply system in SWHEPP by disinfecting the tertiary-treated sewage (i.e., MBR permeate). The produced reclaimed water will be served for non-potable reuse and is required to satisfy the water quality standards shown in Table. 2.1.

2.3 Tools and techniques for enchancing the performance of machine learning modeling

2.3.1 Programming languages

Matrix Laboratory (Matlab) is a proprietary programming and numeric computing platform used across industries and academia for data analysis, algorithm developments, and model

Table 2.1: Endorsed Reclaimed Water Quality Standards from Water Supply Department.

Parameter	Unit	Requirement [a]
E. coli	cfu/100 mL	Not detectable
Colour	Hazen Unit	≤ 20
Ammoniacal Nitrogen (NH_3-N)	mg/L as N	≤ 1
Total Residual Chlorine	mg/L	≥ 0.2
Dissolved Oxygen	mg/L	≥ 0.2
Turbidity	NTU	≤ 5
5-day Biochemical Oxygen Demand	mg/L	≤ 1
pH	-	6-9
Threshold Odour Number	-	≤ 100
Synthetic detergents	mg/L	≤ 5

[a] The water quality standards for all parameters are applicable at the point-of-use of the system.

buildings. In the wastewater treatment industry, Matlab is known for using an add-on software called Simulink for modeling, simulating, and analyzing the dynamic system (i.e., chemically enhanced primary clarifier (Bachis et al., 2015). The use of Matlab-Simulink in the wastewater treatment industry is known for the development of control strategies for WWTP automation. In 1987, International Water Association (IWA) developed the first mathematical model for simulation-based evaluation, which is Activated Sludge Model 1 (ASM 1), and the modified activated sludge models and Benchmark Simulation Models (BSM) was further developed in the following years (bin Talib, 2011). The difference between the two is that ASM is designed for developing control strategies exclusively in the activated sludge treatment process, and BSM 1 is to develop the automation in the entire WWTP (Ballhysa et al., 2020).

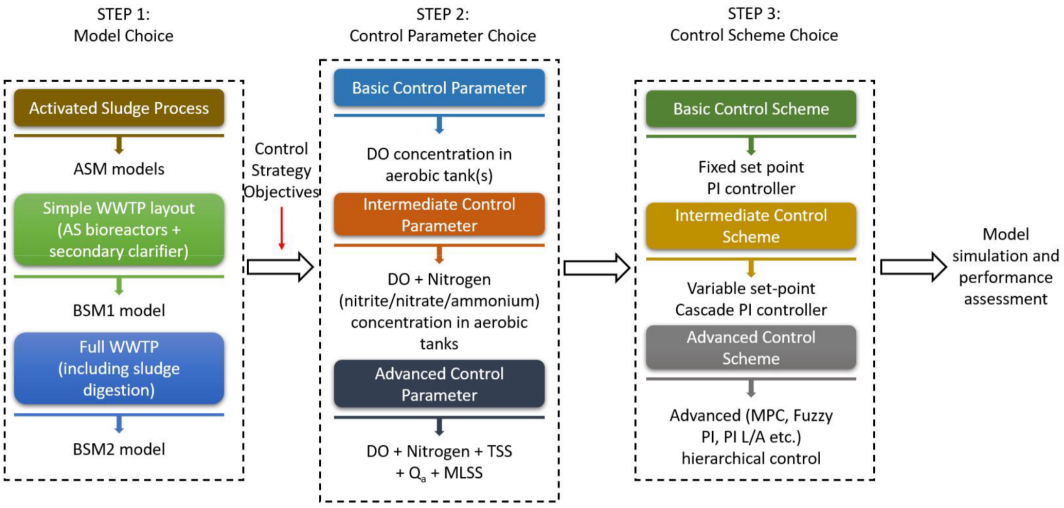


Figure 2.1: Proposed framework for control strategy designed by Ballhysa et al. (2020).

In recent years, many publications presented an exciting way to demonstrate how machine learning-based model predictive control (MPC) can outperform the conventional PID controller in WWTPs using BSM. Fig. 2.1 shows how the control scheme choice iterates from fixed set point PI controller to advanced MPC control, etc. The researchers use Matlab-Simulink to simulate the treatment processes in WWTPs. At the same time, the block of PID controllers is replaced with machine learning models, and the effluent quality or treatment system performance can be differentiated via BSM simulated results. Wang et al. (2020) compared the stability of chlorinated water quality in the effluent of a DWTP with two control strategies, which are PID feedback controls and a predictive model-based support vector machine (SVM). The BSM simulated results showed that the SVM model required 21 minutes less to reach the residual chlorine setpoint than PID feedback controls. A proposed neuro-fuzzy PID controller (i.e., a hybrid machine learning model consisting of neural networks and fuzzy logic) also showed superior performance in optimizing the chlorine dosing rate to minimize the chance of errors (Hong et al., 2012). The significance of using BSM in Matlab-Simulink enables the performance of traditional and machine learning-based control strategies can be compared in objective and fair senarios, also providing the practicability of machine learning to the experts in the field. Matlab is a powerful and resourceful platform providing various machine learning functions, including point-and-click apps for training and evaluation, available classification and regression algorithms, Automatic machine learning (AutoML), etc (MathWorks, 2022c). The direct access to the abundant features and the integration of Simulink make Matlab an appealing option for many researchers in the wastewater treatment industry, especially in the research domain of machine learning and control strategy simulation. Despite the countless benefits of using Matlab, the Python programming language stands out differently.

Python is a high-level, interpreted, and object-oriented programming language and features simple and easy-to-learn syntax providing good readability (Fundation, 2022). The large developer community (e.g., GitHub and Stackoverflow) and open-source access (i.e., free of charge) have made Python an ideal tool for machine learning starters. The most cutting-edge research in the field of Artificial Intelligence is often led by Tech Giants like Google and Amazon, which conduct research via Python (e.g., machine learning frameworks of TensorFlow (Google)in Python), as well as the big research community using Python. All the latest updates and developments relating to machine learning architectures and techniques are usually accessible in the open-source Python community, including the example codes. Contrary to Python, users of commercial software Matlab need to wait for the software engineers working in Matlab

to update the latest machine learning applications onto the Matlab platform, which is a time-consuming process and creates a delay of time and accessibilities to many resources (Castro, 2018). Machine learning developers in the wastewater treatment industry can freely choose between the programming methods based on the research need. Those looking for mature machine learning algorithms can simply use Matlab and be satisfied with the functionalities; on the other hand, those who intend to incorporate more new techniques and architectures in machine learning models can consider using Python as the programming language. Interestingly, MathWorks recently announced using Python functions in Simulink Model (MathWorks, 2022a); despite the update from Matlab, to the best of my knowledge, there are no research papers developing machine learning algorithms on Python and running on Matlab-Simulink.

2.3.2 Data pre-processing techniques

The ubiquitous sensors installed in WWTPs for treatment automation generate a massive amount of data on a daily basis. Before being used for any purposes, the data must be understandable for explanation and relevant enough for water experts to extract valuable information (Kehrein et al., 2020). Without the help of Artificial Intelligence, data manipulation before training machine learning models can be time-consuming and challenging. The specifically designed algorithms can perform data evaluation and augmentation to improve data quality. Any statistical or machine learning algorithms which can complete these tasks are known as the data pre-processing techniques. The causes of sensors rendering undesired data with low quality are the limitations of the hardware sensors and the dynamics of the sampling locations. In general, the false data generated by sensors can be described in eight distinct states (Rosen et al., 2008; Newhart et al., 2019):

1) Operational: Sensor is working properly with normal measurement noise.
2) Excessive drift: When a sensor outputs a value progressively further from the truevalue.
3) Shift: When the output of the sensor is a constant amount away from its true value.
4) Fixed value: When the sensor is stuck and keeps repeating the same value.
5) Complete failure: Similar to a fixed value fault, but the sensors either give offthemaximum or minimum, value, zero or no value at all.
6) Wrong gain: When signals away from the calibration point are under- orover-amplified bythe sensor.
7) Calibration: The sharp change in sensor output directly following a calibration.

8) Isolated fault: When a single point in a series shows an incorrect value.

The researchers and experts have been proposing solutions for filling the data gaps created by sensor faults and maintenance operations. However, the number and length of missing values are mainly subject to the dynamics of the system being monitored and other factors. In their open-source wastewater data treatment toolkit,De Mulder et al. (2018) has recommended five data imputation strategies aimed at data generated from water resource recovery facilities:

1) Interpolate.
2) Use a correlation with other available measurement signals.
3) Replace with a corresponding value in an average daily profile.
4) Repeat the values obtained on the preceding day.
5) Replace with the output of a model.

The efficient monitoring of sensors and proper use of the data for developing control strategies in the wastewater treatment industry rely on careful data quality control. In recent years, automated data evaluation has drawn the attention of experts and researchers in this field as manual detection of sensor fouling is unrealistic because the tasks are labor-intensive and laborious. Alferes et al. (2013) presented three practical approaches for data quality validation, which are capable of automated calculating single abnormal values and collective faults over a long period. The author claimed that the significance of the research work is performing a data quality validation scheme on the multivariate dataset. The pitfall of the study is that despite the promising approaches proposed, the validity still depends on the thresholds or acceptability limits in the actual WWTPs. Similar to the data imputation strategies, the real situation differs tremendously across different WWTPs. That being said, instead of providing general guidance on how to manipulate data, the focus should be emphasized on how to use algorithms to help users understand, analyze, and process the fouling data.

2.3.3 Feature engineering techniques

Feature engineering aims to enrich the raw dataset by selecting, manipulating, and transforming data, which forms a better dataset relating to the underlying targets to be learned by the machine learning model. Feature engineering and data pre-processing are easily confused with each other. The fundamental difference between the two is that the former creates essential

features not included in the raw data; the latter is a data noise removal and cleaning process. In the study of Mamandipoor et al. (2020), feature engineering was performed to generate five extra features, which are the statistical metrics of mean, maximum, minimum, variance, and standard deviation of a specific input feature. However, in comparing the final results, the author only emphasized evaluating model accuracies across varied machine learning models (i.e, PCA-SVM and LSTM models). Another interesting technique used by Zaghloul et al. (2021) is to create the gradient values of an input variable to assist the model in better learning the trend of the historical removal rate of water parameters in aerobic granular sludge reactors. Similar to the results shown in the work of Mamandipoor et al. (2020), the influence of how engineered features affect the ultimate model accuracy is excluded in the results and discussion part. Thus, creating a lack of knowledge in how significant the feature engineered inputs are to the model accuracy and which techniques can be used in which senarios.

There is considerable ambiguity concerning the necessity of using feature-engineered inputs in training predictive models in WWTPs. In predicting total nitrogen (TN) in the effluent, Guo et al. (2015) input nine features and performed feature sensitivity analysis, which can capture the change of the output values attributed to the change input. The result showed that the top three most significant inputs, temperature, TN flow, and pH, are critical in predicting TN. The author claimed physical related cause-and-effect relationships between the effluent TN and those top three effective features could be elucidated by the machine learning model. In another work on predicting influent BOD concentration, the study clearly stated that using five inputs instead of three will cause model overfitting. Three inputs for model training were considered sufficient (Alsulaili and Refaie, 2021). Variables created from feature engineering have no physical properties, leading to extra unexplainable essence in addition to the black-box nature of machine learning models. Besides, extra model inputs from feature engineering can also cause overfitting if the data quality is not carefully evaluated. Said by Andrew Ng, ”Coming up with features is difficult, time-consuming, requires expert knowledge. Applied machine learning is basically feature engineering”. From the quote and the recent studies, we are uncertain how feature engineering techniques can practically help the development of machine learning models in the wastewater treatment industry. More research is required to elucidate further the effectiveness of performing feature engineering.

CHAPTER 3

METHODS AND MATERIALS

3.1 Wastewater treatment plant description

3.1.1 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^3/day to 93,000 m^3/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^3/day by the end of 2025. As shown in Fig. 3.1, 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. 3.1.

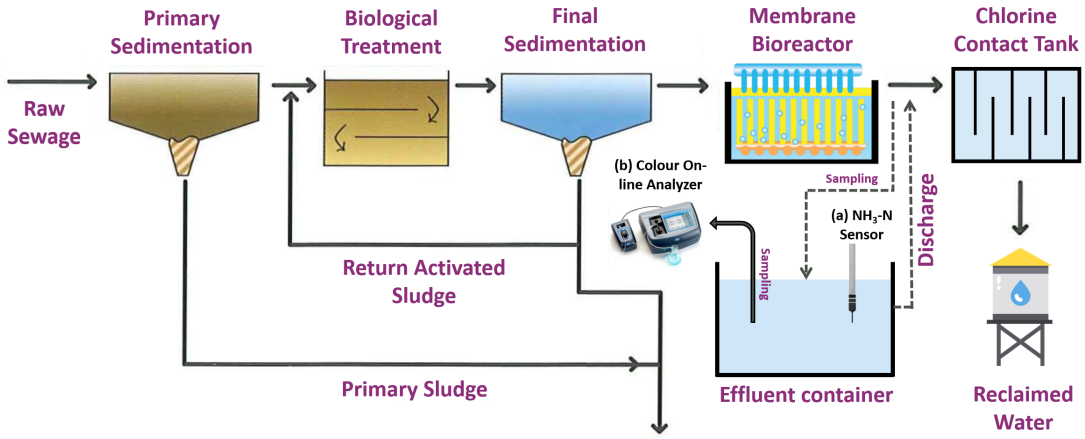


Figure 3.1: Sewage treatment process flowchart at SWHEPP (adapted from Drainage Services Department 2020)

3.2 Data collection and preparation

Figure 3.2: 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.

3.2.1 On-line data monitoring and collection

To enable us to perform on-line monitoring of ammonium concentrations (NH_3-N) in the MBR effluent, an Ammonium and Potassium Probe, AmmoLyt®Plus 700 IQ (Xylem Company) was installed as Fig. 3.3a in the effluent container, as shown in Fig. 3.2. 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 NH_4-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_4-N concentrations the dominant species. Both ammonia and ammonium contain one nitrogen atom; 1 mg/L NH_3-N is the same as 1 mg/L NH_4-N. Thus, to prevent confusion, in the following paragraph, the unit of NH_4-N will be expressed by NH_3-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. 3.3b.

(a) AmmoLyt®Plus 700 IQ, Xylem.

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

Figure 3.3: 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. 3.4a and a peristaltic pump as Fig. 3.4b 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. 3.4c.

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. 3.6. 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

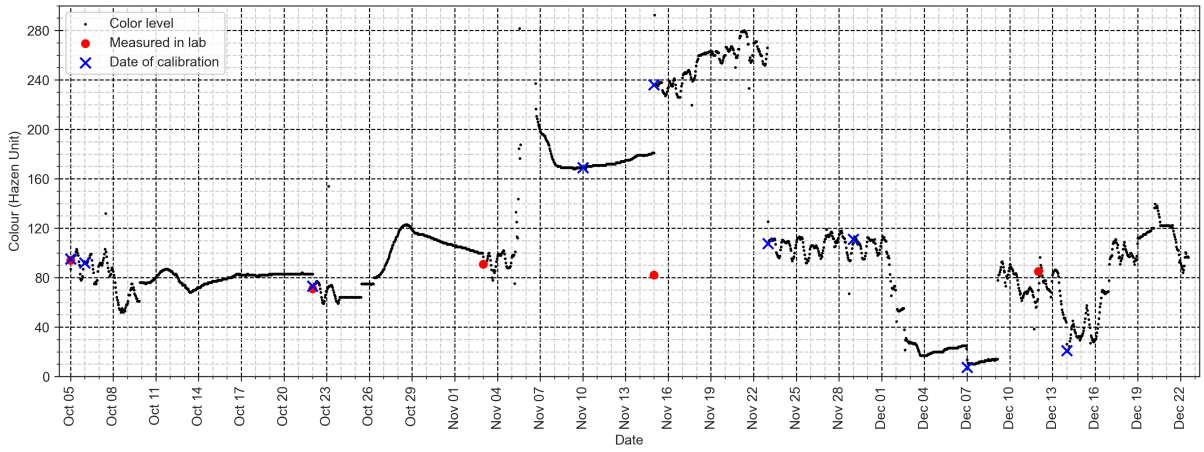
(b) DR3900 spectrophotometer, Hach Company

(c) Customized clicker/actuator with programmable timer

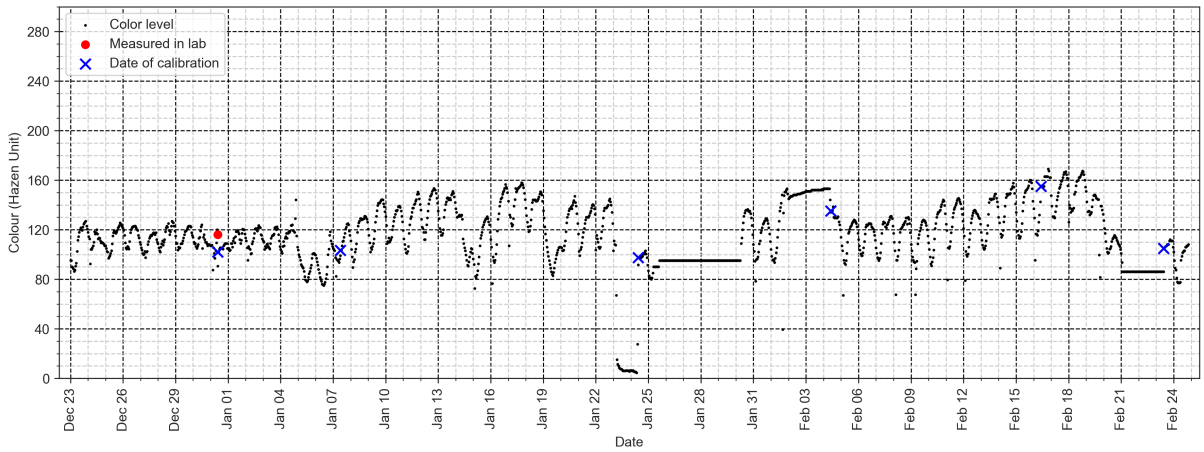
Figure 3.4: Instruments of on-line colour analysis system.

As shown in Fig. 3.5, 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. 3.7.



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



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

Figure 3.5: The dates of manually calibration and colour level measured in the laboratory were plotted as blue crosses and red dots.

3.2.2 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 $\boldsymbol{y}$ and the model outputs $\hat{\boldsymbol{y}}$, 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:

$$MSE = \frac{\sum(y_i - \hat{y}_i)^2}{n} \tag{3.2.1}$$

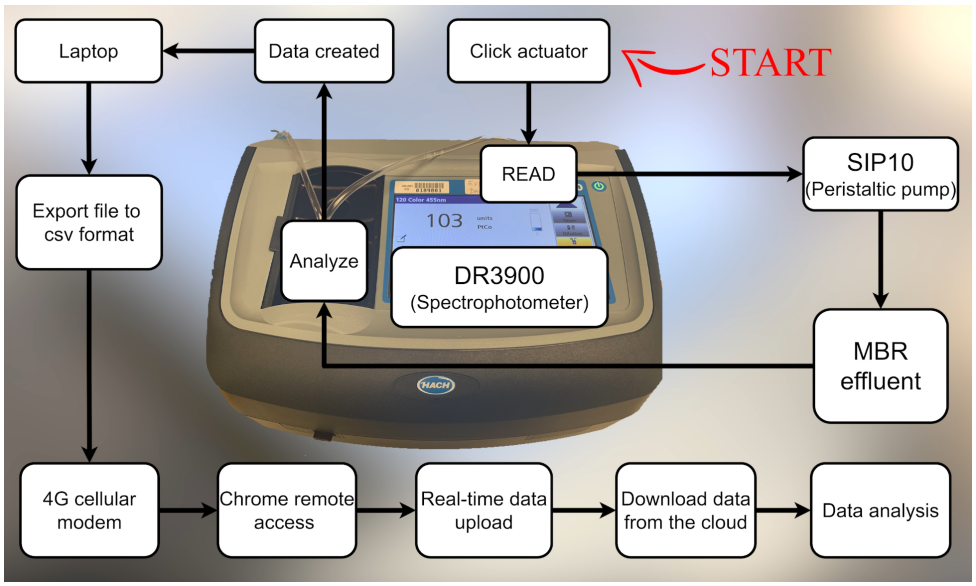


Figure 3.6: Schematic diagram of the custom-made on-line colour analysis system.

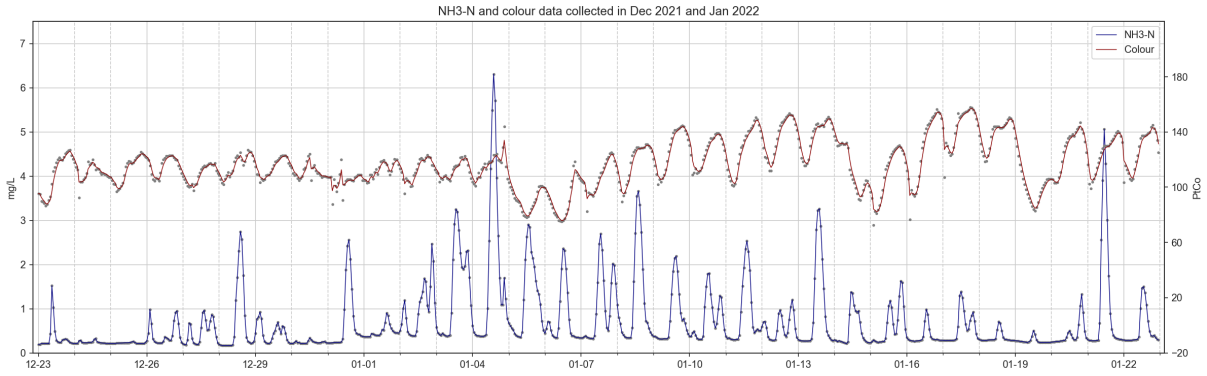


Figure 3.7: Ammonia and colour data collected from 23 December 2021 to 22 January 2022.

3.2.3 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. 3.8. 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

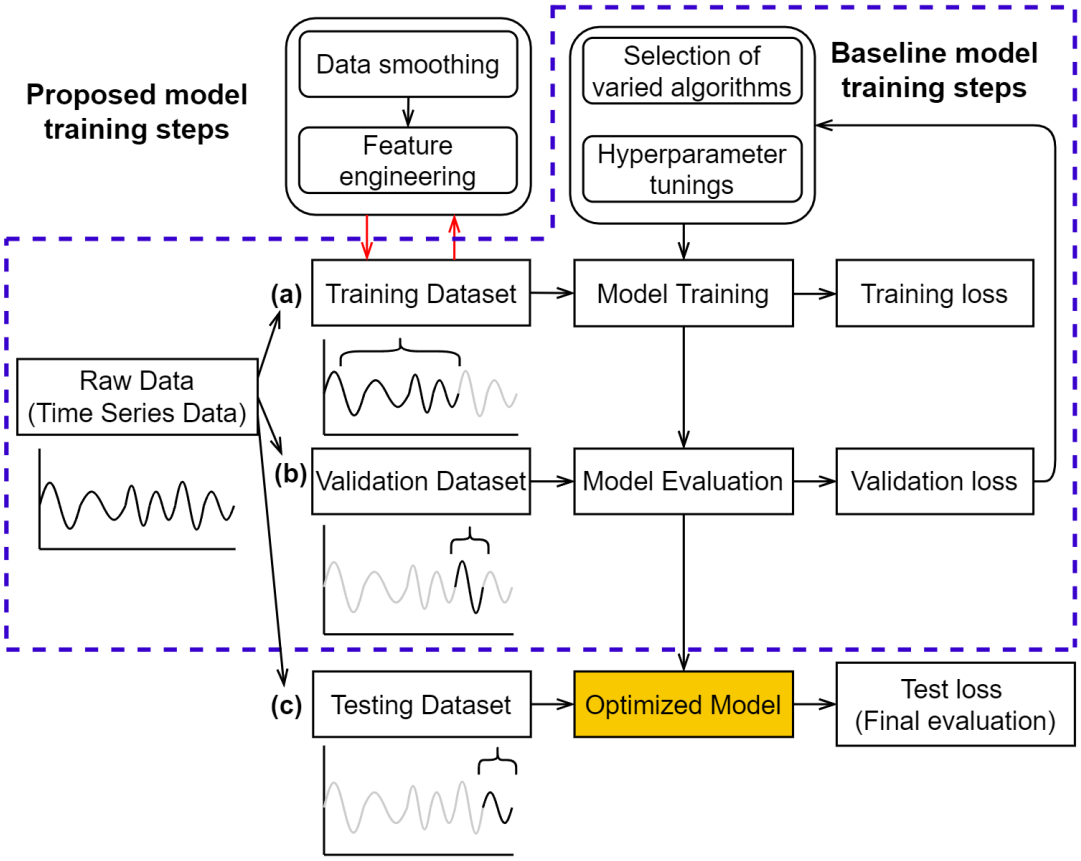


Figure 3.8: Training steps of the machine learning models.

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.

3.2.3.1 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. 3.9a, 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.3.9a)
2) Determine polynomial degree (e.g., different polynomial degree is compared in Fig. 3.9a).
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 $\boldsymbol{Y_j}$ can be expressed in Eq. 3.2.2:

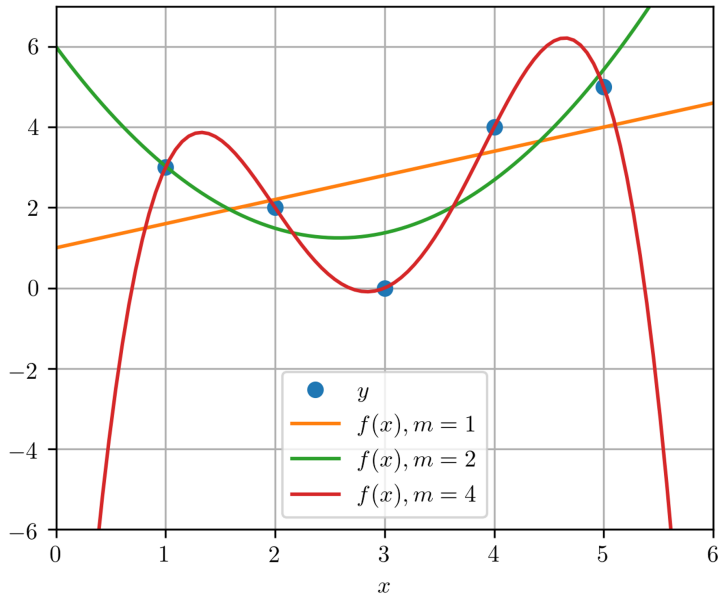
$$Y_j = (C \otimes y)_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i y_{j+i}, \; \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \tag{3.2.2}$$

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)).

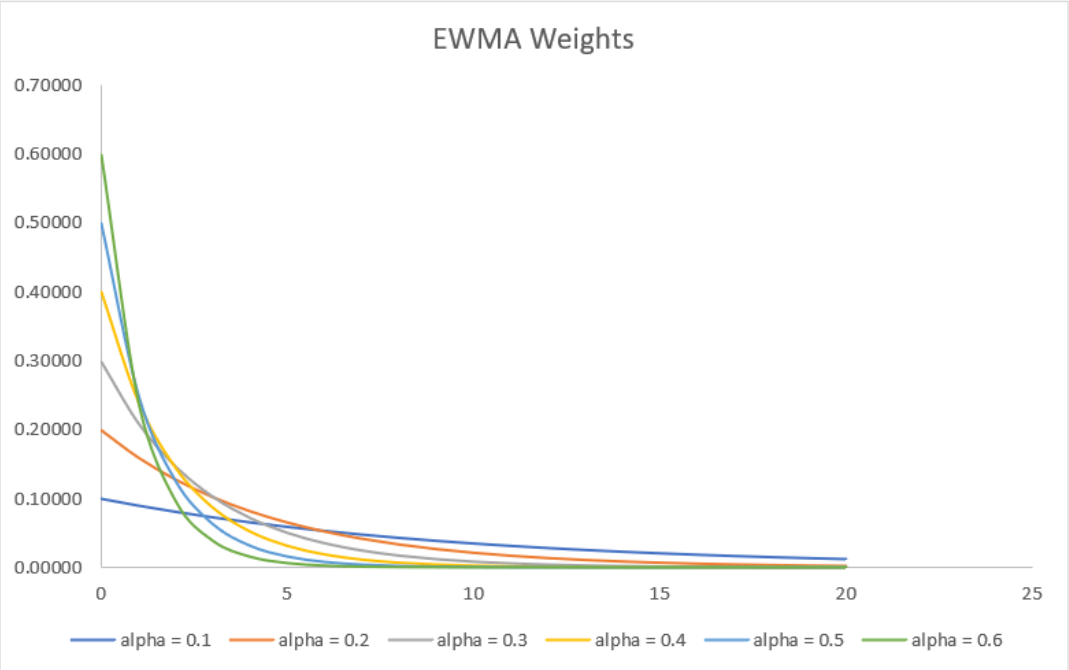
Exponentially weighted moving average (EWMA), also known as autoregressive (AR) filtering, is a technique that filters measurements. An EWMA filter smoothes a measured data point by exponentially averaging that particular point with all previous measurements. The EWMA equation can be expressed in Eq. 3.2.3:

$$\begin{aligned} \alpha &= \frac{2}{span + 1} \\ y_0 &= x_0 \\ y_t &= (1 - \alpha) y_{t-1} + \alpha x_t \end{aligned} \tag{3.2.3}$$

where α corresponds to the decay paratmeters, 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 3.9: 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. 3.1.

Table 3.1: 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. 3.10 and Fig. 3.11 show the influences of different windows sizes of SG and EWMA filters on ammonia concentrations and colour levels datasets.

3.2.3.2 Outlier Removal

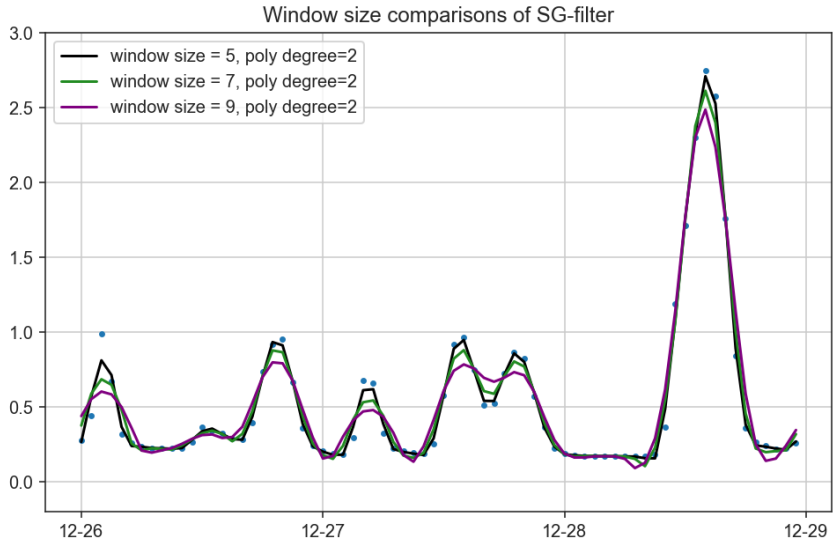
Although the extreme values in the raw ammonia dataset were removed based on basic rules (i.e., concentrations higher than 7.0 mg/L), the ammonia sensor can still collectively capture unideal data points. In the outlier removal process, we intended to identify the collective faults of ammonia data in 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_3-N fluctuation ≤ 0.1 (i.e., lower than the sensor resolution).
2) No diurnal fluctuation (i.e., Fluctuation = peak value – bottom line value).

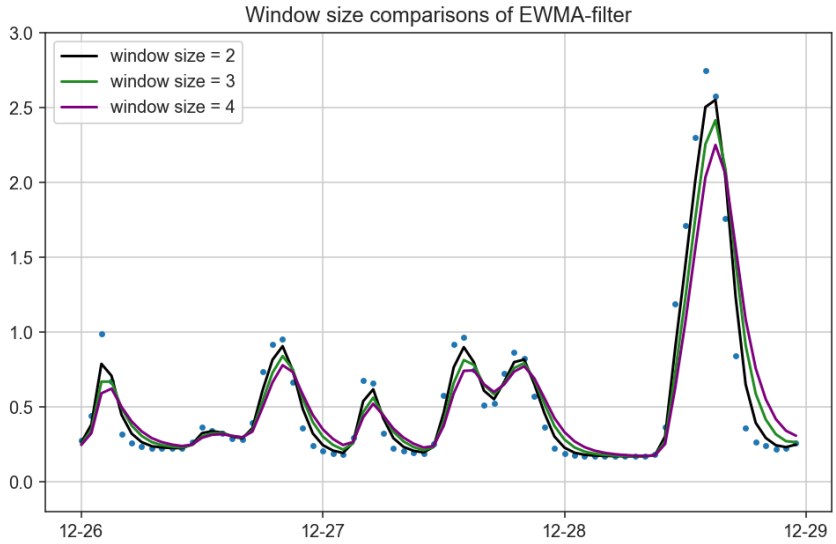
Peak analysis was performed on the daily ammonia data to automatically identify normal or abnormal signals. The analysis takes a one-dimension array (i.e., the data form of ammonia in a day) and finds all local maximum values by comparing neighbouring values. This function will also provide information such as width and prominence, as in Fig. 3.12 to help us identify whether the diurnal fluctuation exists.

3.2.3.3 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. 3.13. 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,



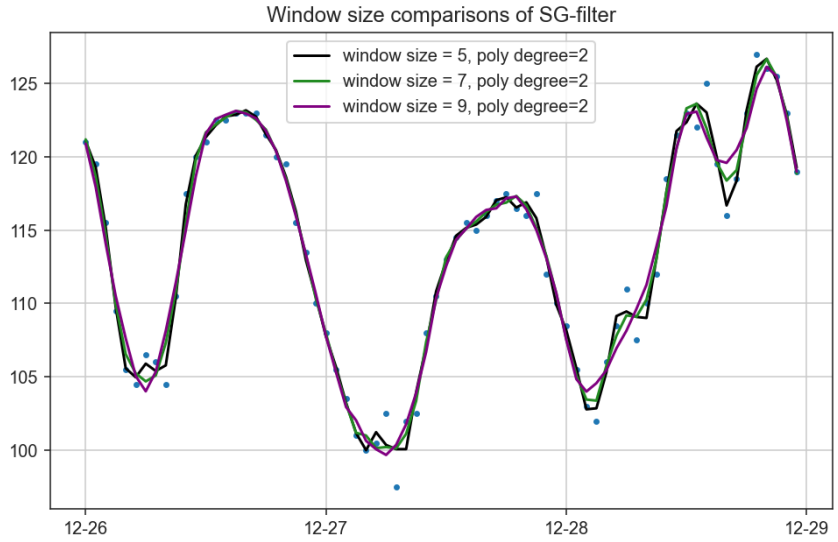
(a) Ammonia data filtered by SG filters with different window sizes.



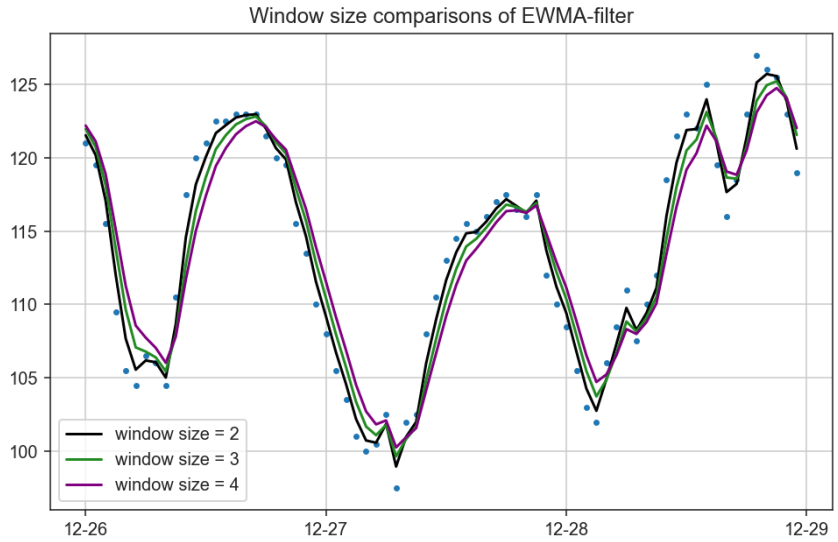
(b) Ammonia data filtered by EWMA filters with different window sizes.

Figure 3.10: Comparisons of the applying different window sizes on ammonia concentration datasets.

the colours level were also higher, as shown in Fig 3.15a. 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. 3.14a, 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. 3.15b, 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 3.11: 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. 3.19.

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. 3.16, 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. 3.17, the peak analysis tool helped us to identify the ammonia concentrations' peak hour,

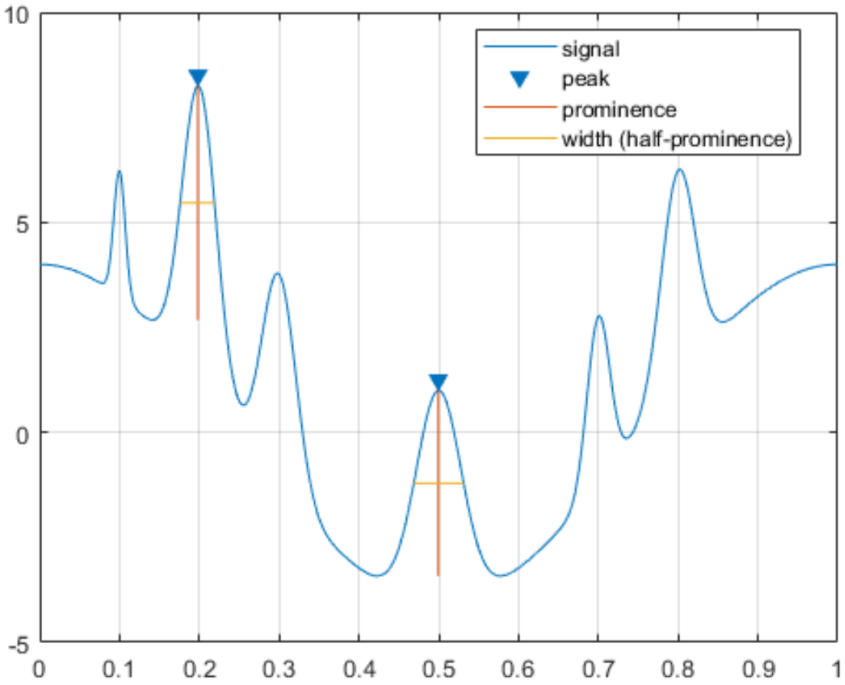


Figure 3.12: Illustration of peak analysis. Four important elements were automatic calculated by the function (MathWorks, 2022b).

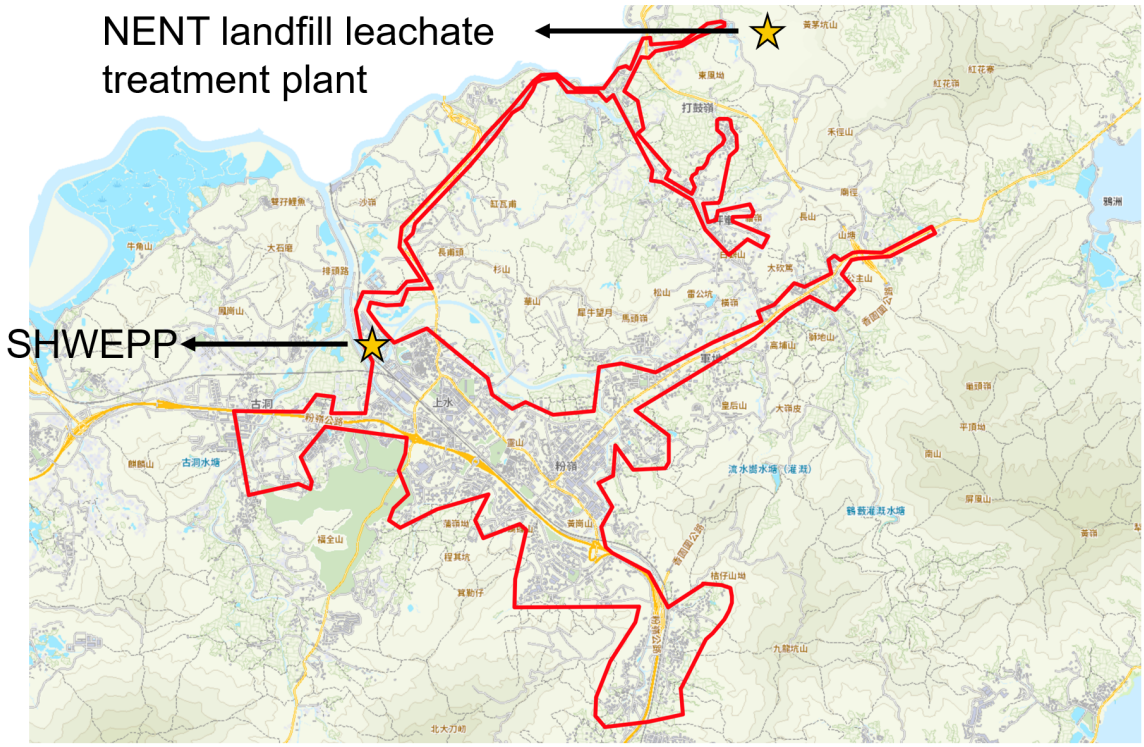
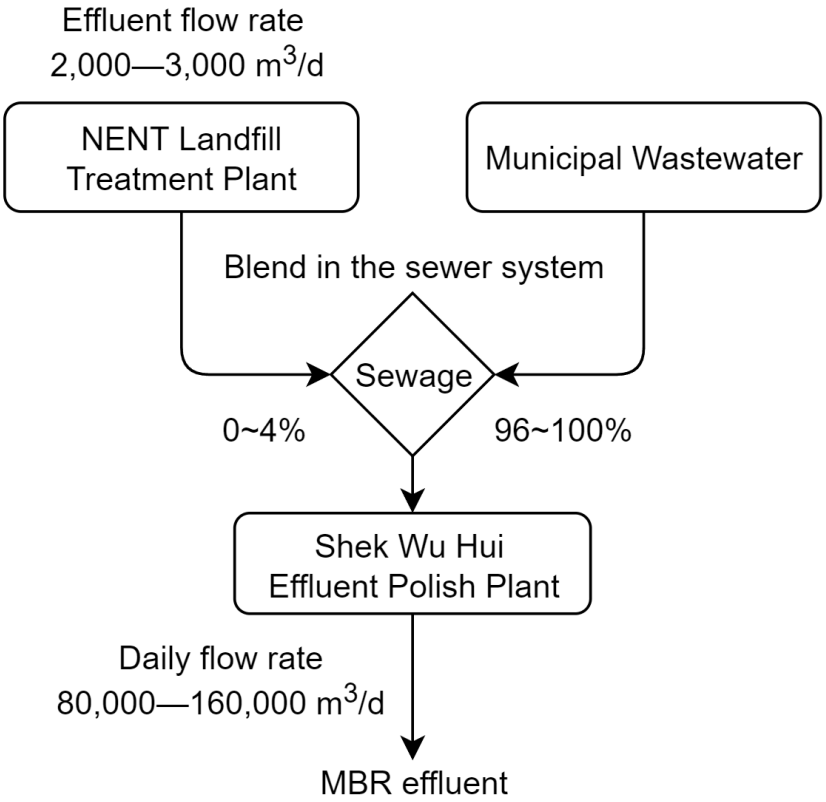
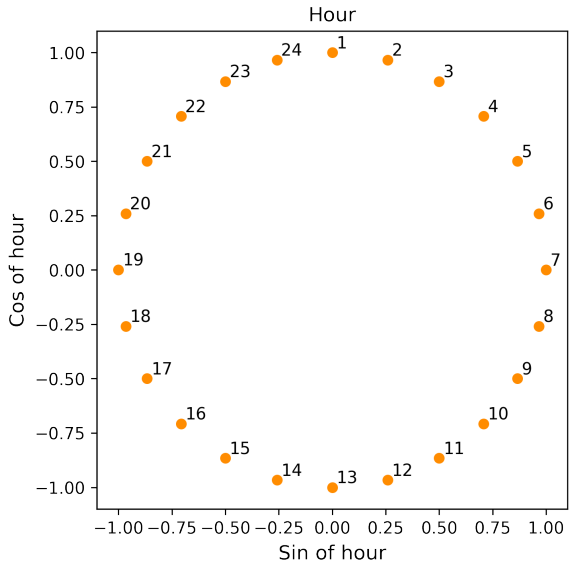


Figure 3.13: Sewer system coverage of SHWEPP. The covered areas (i.e., area circled in red boundary) include Fanling/Sheung-Shui new towns and NENT landfill leachate treatment plant.

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-



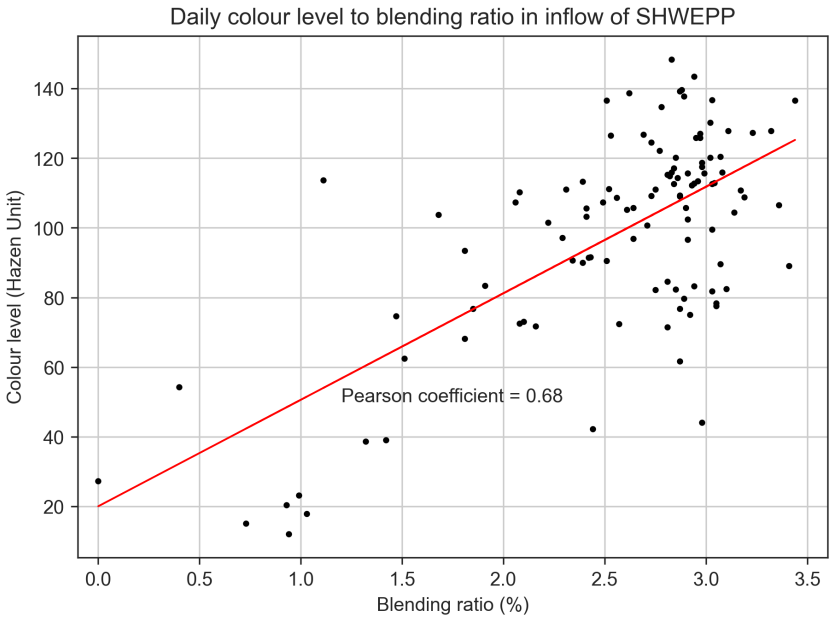
(a) Flowchart showing the blending of treated leachate effluent with municipal wastewater.



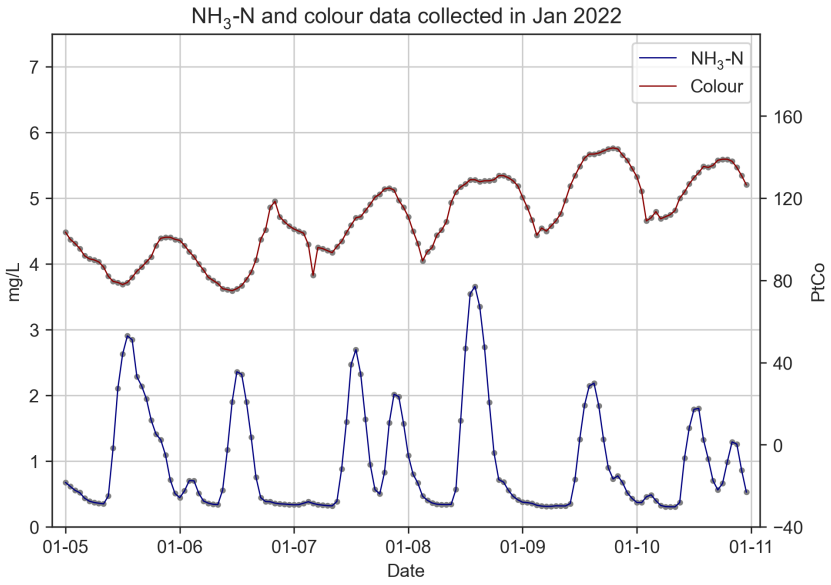
(b) Positional encoding of hour components.

Figure 3.14: 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 3.15: 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 bed 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. 3.14b.

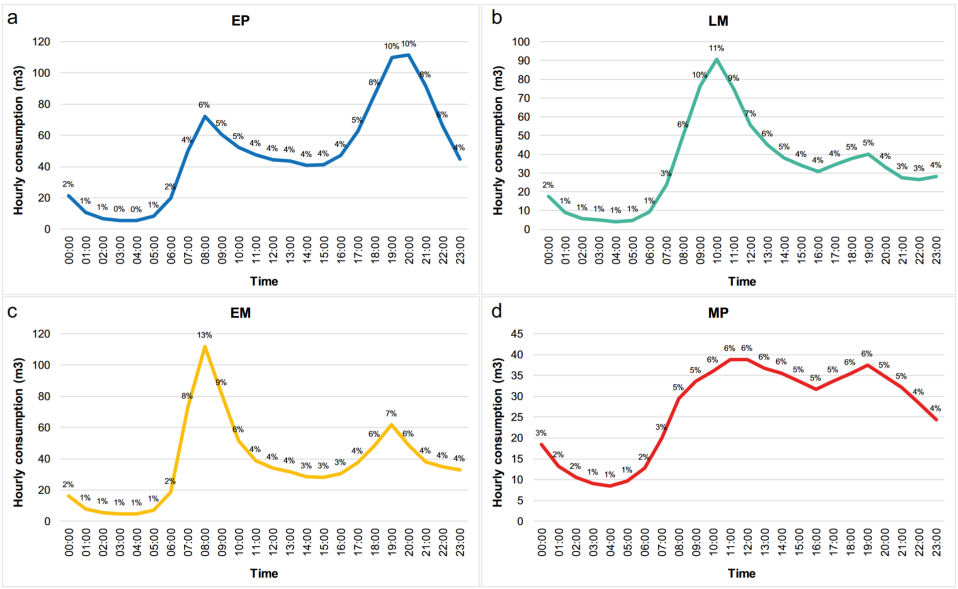


Figure 3.16: Hourly water consumption patterns in households (Abu-Bakar et al., 2021). (a) Cumulative pattern and percentage of hourly consumption for households in the "Evening Peak (EP)" cluster (b) Cumulative pattern and percentage of hourly consumption for households in the "Late Morning Peak Peak (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).

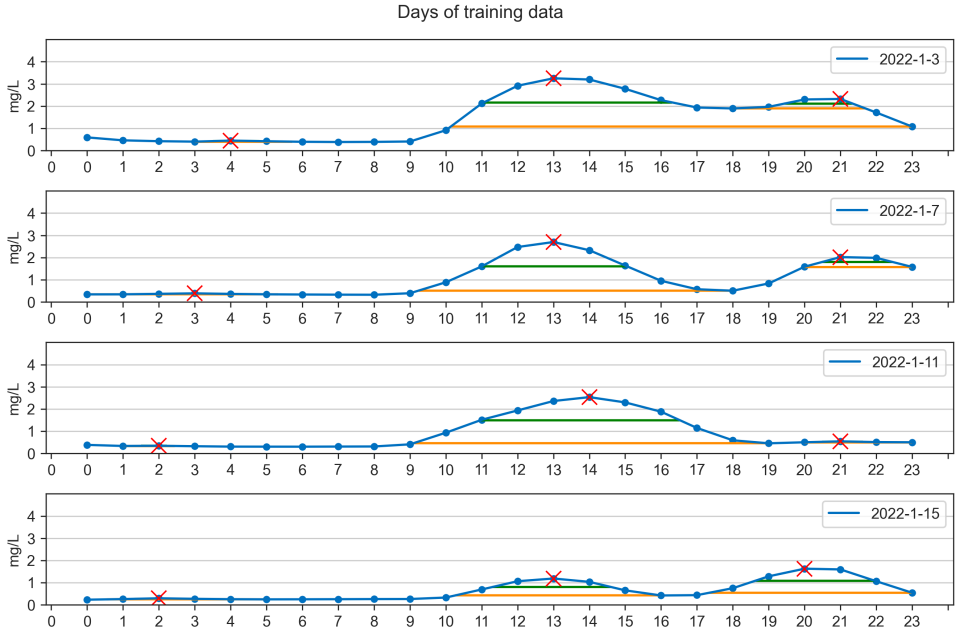


Figure 3.17: The daily patterns of ammonia concentrations on 3, 7, 11, and 15 January 2022.

3.2.4 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 $\boldsymbol{X}$ and output variables $\boldsymbol{Y}$ (i.e., training X/training Y, testing X/testing Y). During the training process, machine learning algorithms will learn a target function $\boldsymbol{f}$ to best map $\boldsymbol{X}$ to $\boldsymbol{Y}$. 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. 3.8. 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. 3.18 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^{th} 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^{th} 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.

For forecasting 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^{th} 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^{th} hour), as shown in Forecast Set 2 in Fig. 3.18. 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^{th} 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. 3.8, is used to assess the model performance until we obtain the optimized hyperparameter settings, including the number of neurons

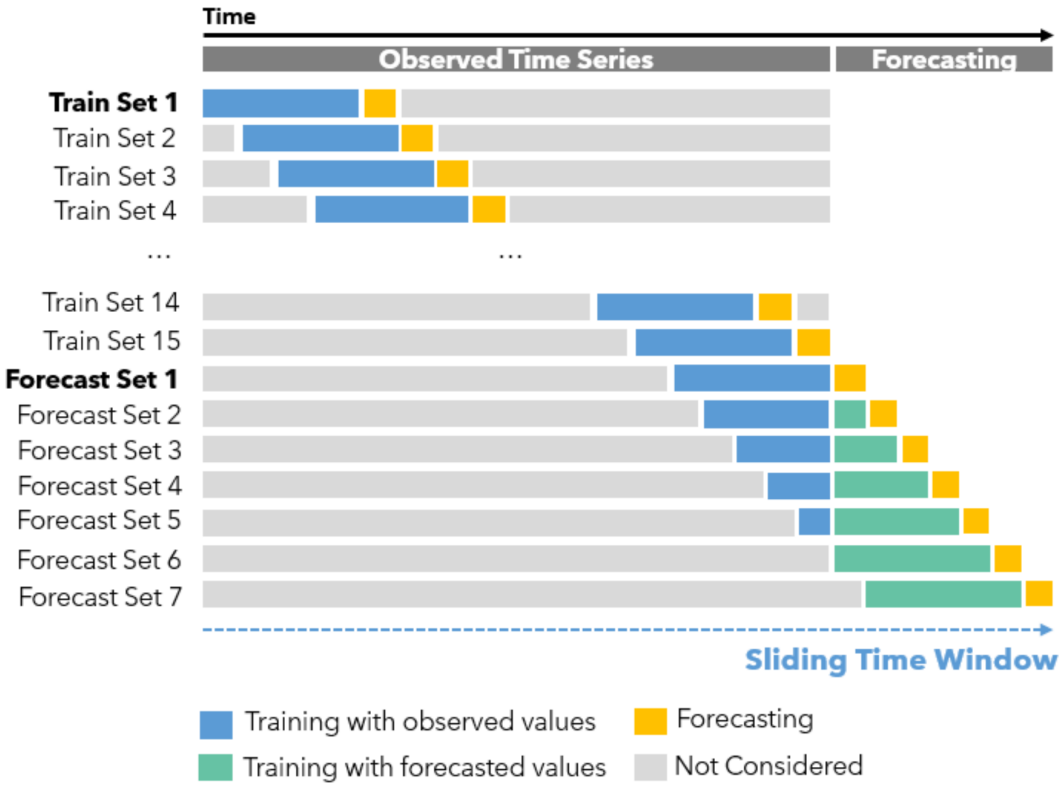


Figure 3.18: Concept of forecasting models (Liu, 2020).

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. 3.8. 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. 3.8, 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.

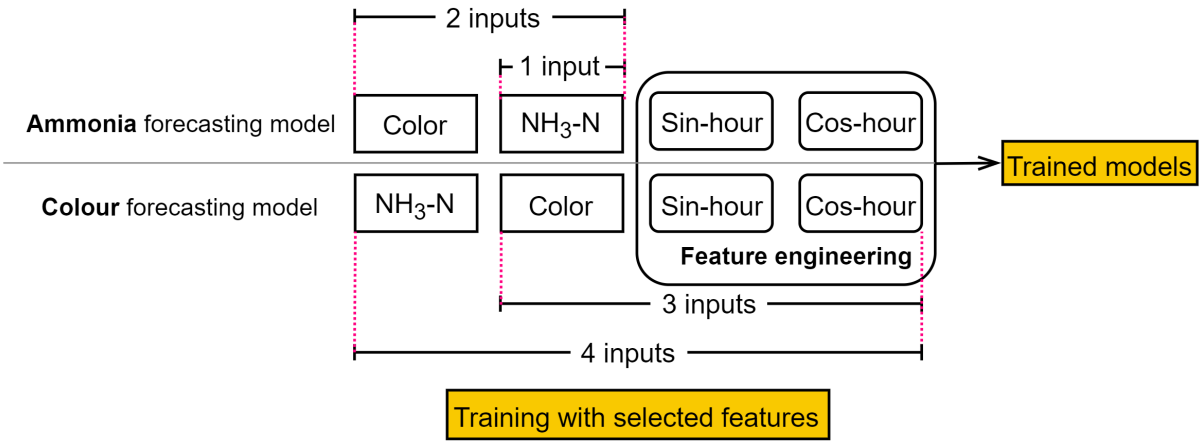


Figure 3.19: Illustration of feature selections for model training.

3.2.5 Feature selection

Fig. 3.19 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.

3.3 Machine learning models

3.3.1 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. 3.20a. The training algorithm applies the general technique of bootstrap aggregating, also known as bagging, to tree learners. Given a training set $X = x_1, ..., x_n$ with targets $Y = y_1, ..., y_n$, 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 $b = 1, ..., B$:

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. 3.3.1:

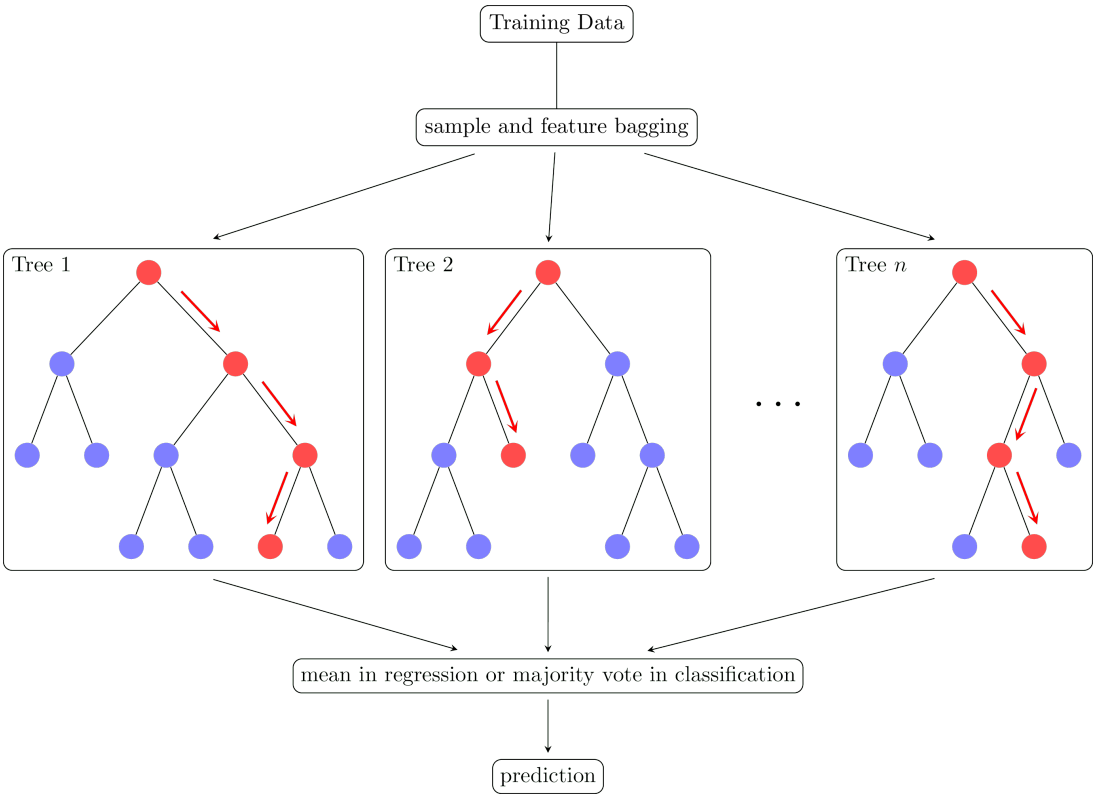
$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x^{'}) \tag{3.3.1}$$

3.3.2 Deep Neural Network

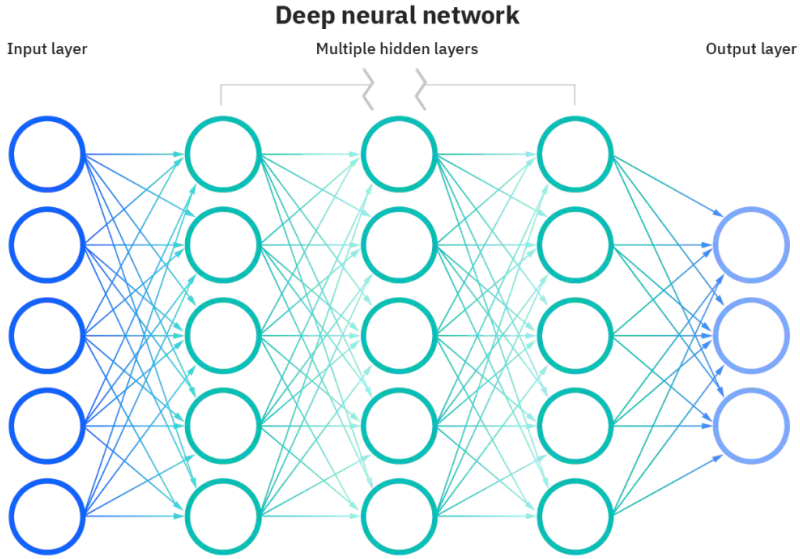
Artificial Neural Network (ANN) is a broad term that encompasses any form of Deep Learning model. A typical ANN consists of input, hidden, and output layers, and each layer comprises multiple neurons (i.e., nodes). The connected neurons simulate the human brain by processing and transmitting input signals 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. 3.20b. The DNN models are nonlinear, which finds the correct mathematical manipulation to turn the input into the output (Bangaloreai, 2018).

3.3.3 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,



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



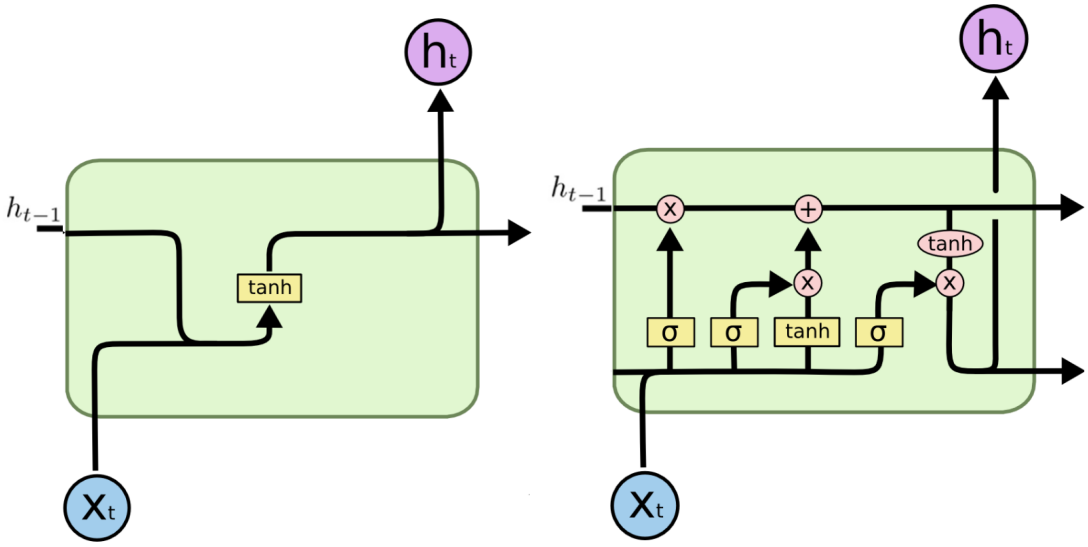
(b) Deep Neural Network (DNN) (IBM, 2022).

Figure 3.20: Illustration of RF and DNN model structure.

thus capturing short-term dependencies. The RNN architecture is presented in Fig. 3.21a. The general formulation of a RNN is expressed in Eq. 3.3.2 (Mamandipoor et al., 2020):

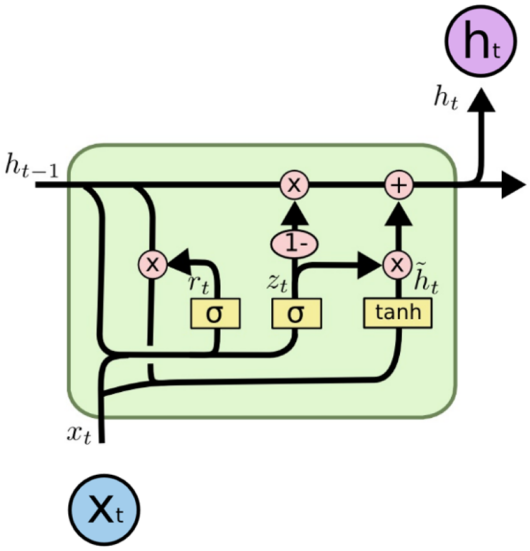
$$h_t = \sigma(W^h h_{t-1} + W^x x_t + b) \tag{3.3.2}$$

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, σ is the sigmoid activation function.



(a) Recurrent Neural Network (RNN).

(b) Long Short-Term Memory (LSTM).



(c) Gate Recurrent Unit (GRU).

Figure 3.21: 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, σ is the sigmoid activation function,$\times$ is the vector pointwise multiplication, $+$ is the vector pointwise addition.

3.3.4 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 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. 3.3.3a, 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. 3.3.3b 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. 3.3.3c. 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. 3.3.3e and Eq. 3.3.3f (Le et al., 2019). The equations for LSTM structure are shown in Eq. 3.3.3:

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f) \tag{3.3.3a}$$

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \tag{3.3.3b}$$

$$\tilde{C}_t = tanh(W_n[h_{t-1}, X_t] + b_n) \tag{3.3.3c}$$

$$C_t = C_{t-1}f_t + \tilde{C}_t i_t \tag{3.3.3d}$$

$$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o) \tag{3.3.3e}$$

$$h_t = O_t tanh(C_t) \tag{3.3.3f}$$

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, σ 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.

3.3.5 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. 3.21c, 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. 3.3.4a, 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. 3.3.4b, which is used to decide how much of the past information to forget. Although Eq. 3.3.4a and Eq. 3.3.4b 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 ($]tildeh$) as in Eq. 3.3.4c, 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. 3.3.4d. The equations of GRU structures are presented in Eq. 3.3.4 (Cheng et al., 2020):

$$z_t = \sigma(X_t W_{xz} + h_{t-1} W_{hz} + b_z) \tag{3.3.4a}$$

$$r_t = \sigma(X_t W_{xr} + h_{t-1} W_{hr} + b_r) \tag{3.3.4b}$$

$$\tilde{h_t} = tanh(X_t W_{xh} + (r_t \circ h_{t-1}) W_{hh} + b_h) \tag{3.3.4c}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tilde{h_t} \tag{3.3.4d}$$

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, σ 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.

3.3.6 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. 3.20a, each estimator, known as the tree in the forest, makes a decision. Therefore, we need to set the num-

ber 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

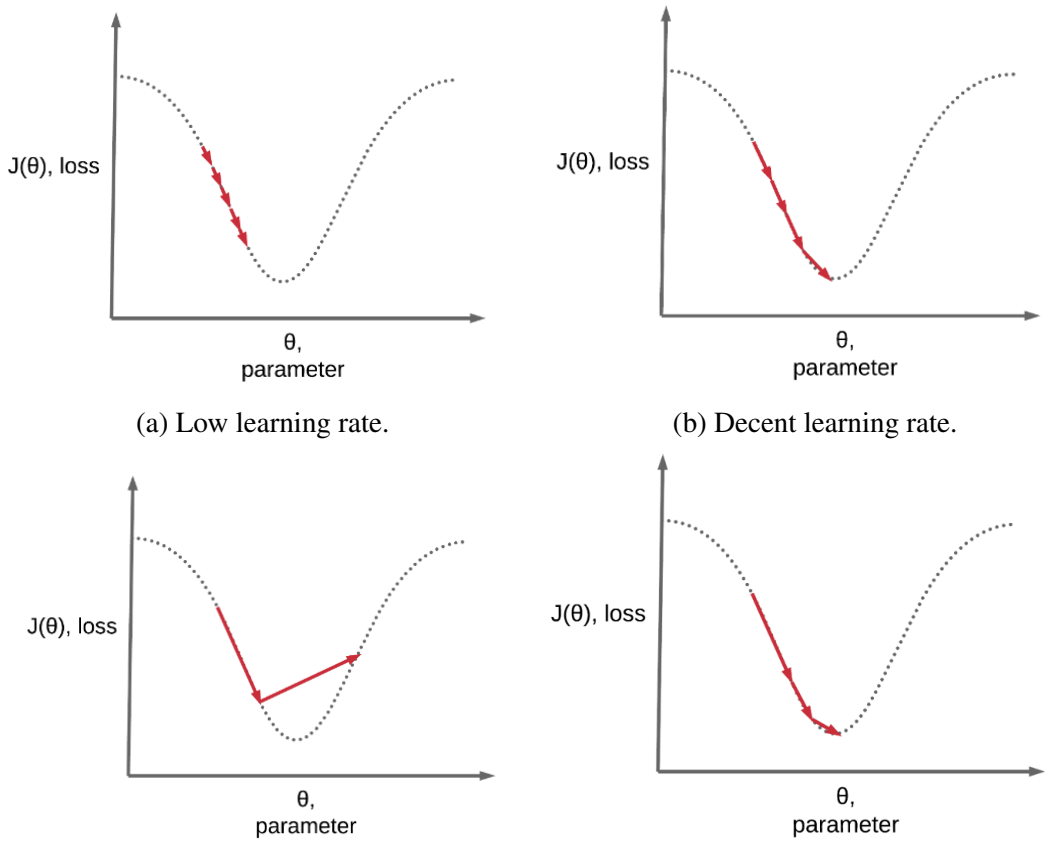
Model-specific hyperparameters

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. 3.22c, yet a too low learning rate will either be too slow to converge or get stuck in a local minimum loss as in Fig 3.22a. A good size of learning rate should reach the minimum loss at a reasonable time, as in Fig. 3.22b. 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. 3.22d.

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



(a) Low learning rate.

(b) Decent learning rate.

(c) High learning rate.

(d) Decaying learning rate.

Figure 3.22: Illustration of how different step sizes of learning rate reach the minimum loss (Ritchie Ng, 2019).

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 3.2: Final model configurations.

Model	Input	h.d[a]	Output	Num. of Exp[b]	Comments
RF	24[c]	-	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

[a] Hidden layer.

[b] The times the experiments were repeated.

[c] 24 hourly data points were input into the models for training.

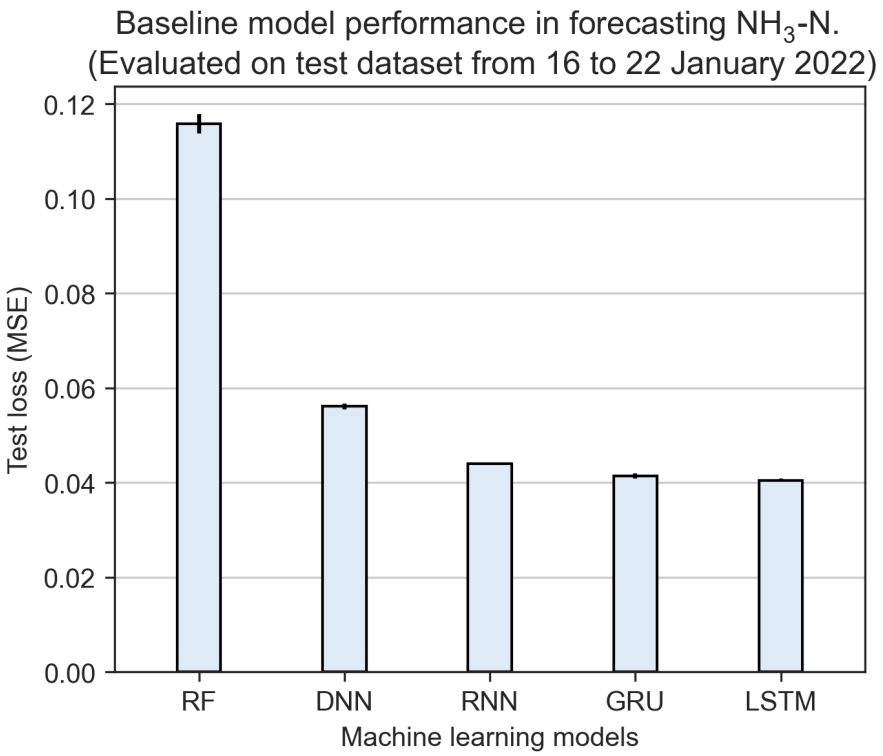
CHAPTER 4

RESULTS AND DISCUSSION

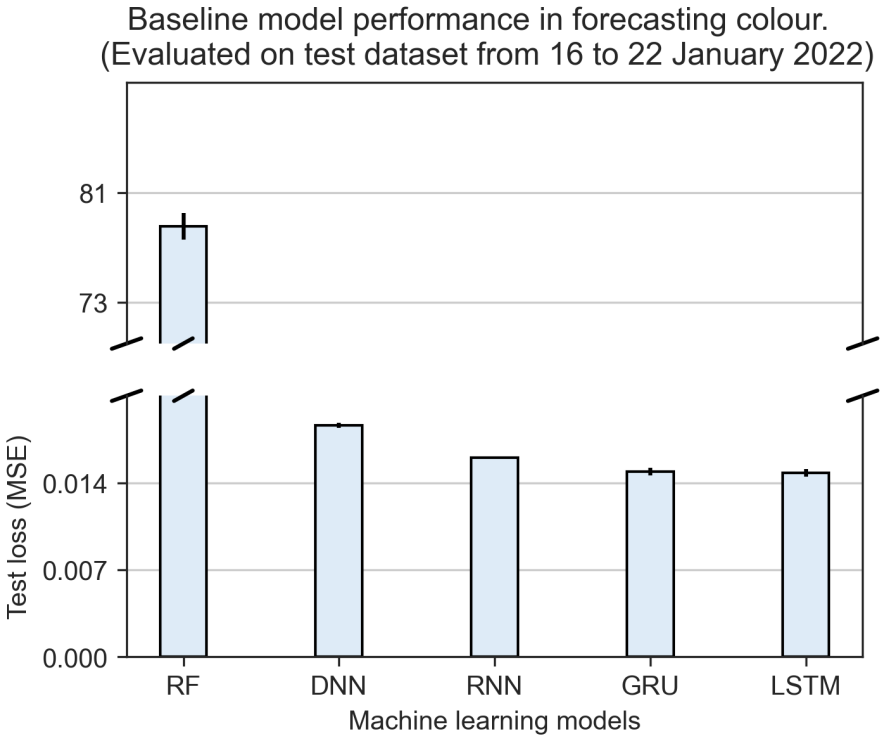
4.1 Baseline performance of the forecasting models

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. 4.1. As shown in Fig. 4.1a, 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



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



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

Figure 4.1: Baseline performance of the ammonia and colour forecasting models.

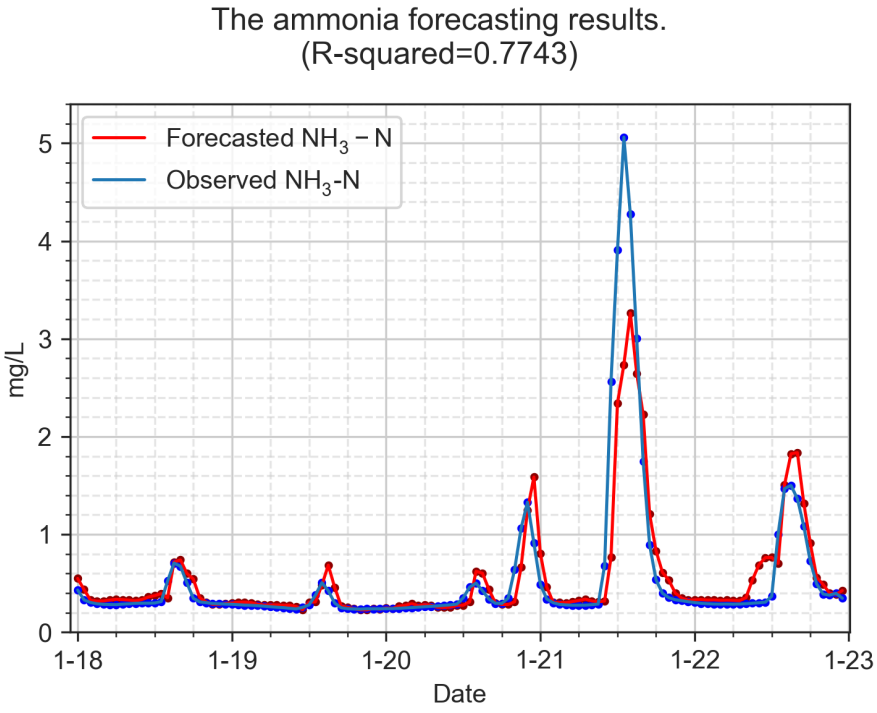
of recurrent neural network. Compared to RNN models, both models contain more ”gates” in the architectures to help control the flow of information, enabling the models to capture more details. The number of gates in RNN, GRU, and LSTM is one, three, and four; theoretically, GRU and LSTM can learn more information from the data based on a greater number of gates. The results in Fig. 4.1a 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. 4.1b, 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. 4.3a, 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.

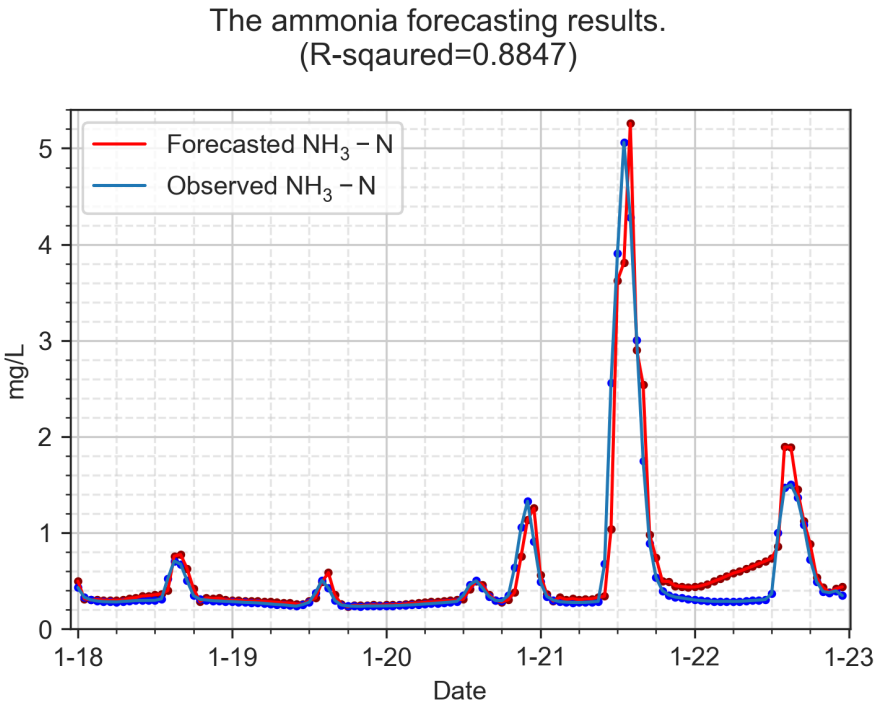
The performance of DNN, RNN, GRU, and LSTM models, from the best to the least, are identical to what we observed in the results of ammonia forecasting models. LSTM model has the lowest test loss of 0.0148, followed by the GRU, RNN, and DNN models. In colour forecasting models, the model performance of 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. 4.2

and Fig. 4.3, one-step-ahead forecast horizon of ammonia concentrations and colour levels are plotted by RF as in Fig. 4.2a and Fig. 4.3a and LSTM models as in Fig. 4.2b and Fig. 4.3b. It is easier to observe that the RF models are less capable of predicting the water quality parameters.

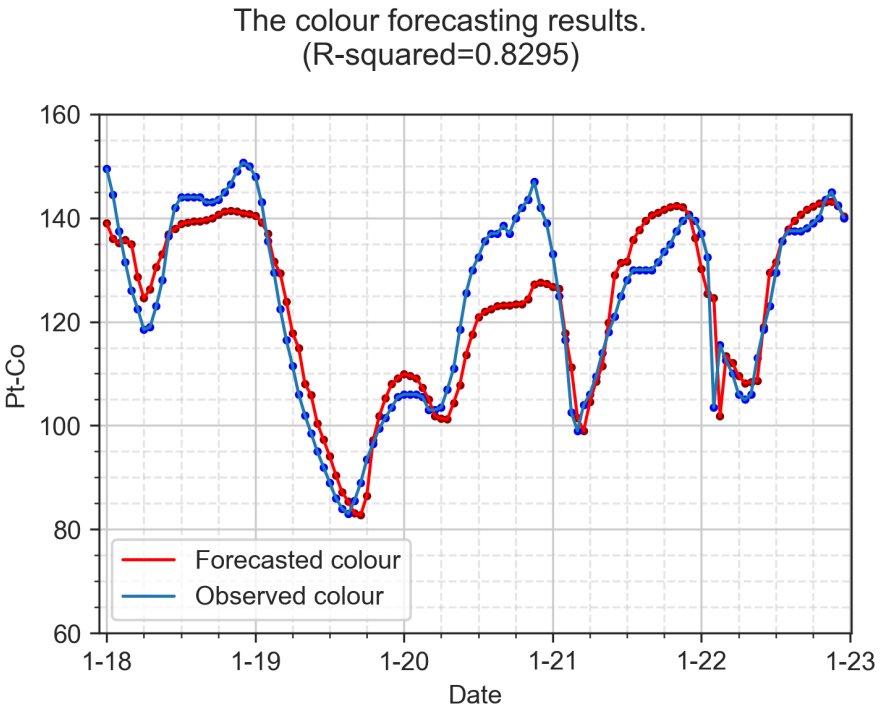


(a) Baseline RF model forecasting ammonia concentration.

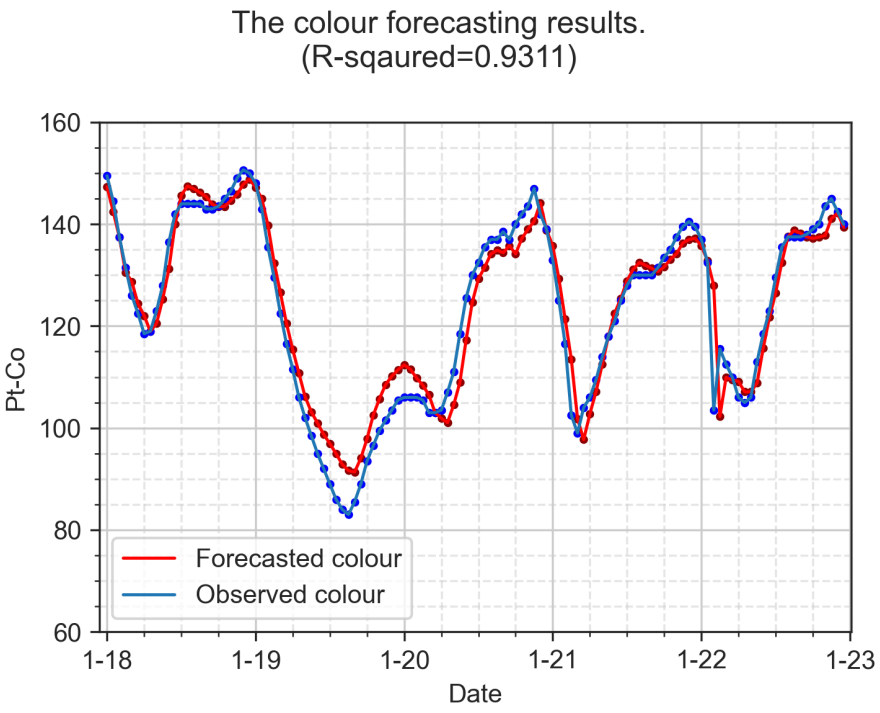


(b) Baseline LSTM model forecasting ammonia concentration.

Figure 4.2: Visualization of the baseline ammonia forecasting results.



(a) Baseline RF model forecasting colour levels.



(b) Baseline LSTM model forecasting colour levels.

Figure 4.3: Visualization of the baseline colour forecasting results.

4.2 Improved performance on forecasting models using data pre-processing techniques

4.2.1 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. 4.1 and Table. 4.3, 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 4.1: Baseline performance of the ammonia forecasting model, evaluated on test dataset from **16 to 22 Janurary 2022**. Loss values were calculated by MSE.

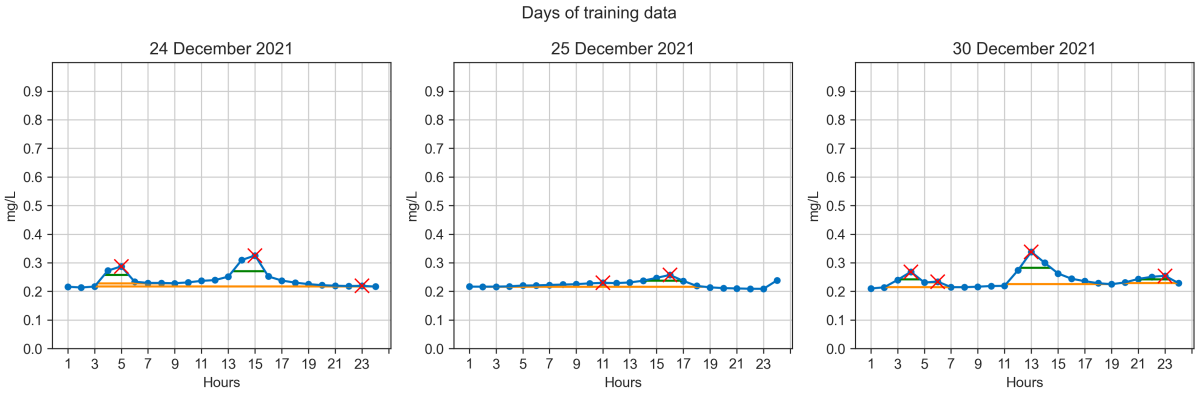
Model-Dataset	Test loss	Valid loss	Model-Dataset	Test loss	Valid loss
GRU-sg7	0.0383	1.2508	RNN-or	0.0432	1.6345
GRU-sg5	0.0385	1.2644	RNN-ew3	0.0434	1.6041
LSTM-ew3	0.0388	1.0796	RNN-obs	0.0440	1.6734
LSTM-sg5	0.0388	1.2346	RNN-sg9	0.0442	1.7046
LSTM-sg7	0.0388	1.1804	DNN-obs	0.0561	3.2383
GRU-ew2	0.0389	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. 4.1, 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. 4.4, 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.

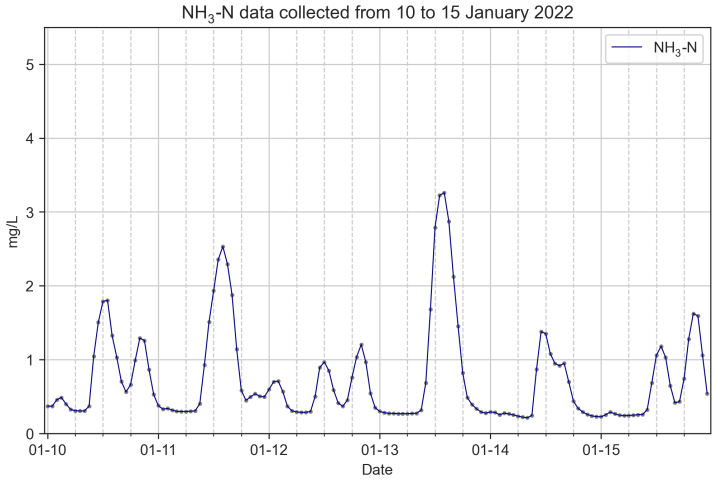
Figure 4.4: Results of the removed outliers from the training dataset.

in terms of lowing 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.

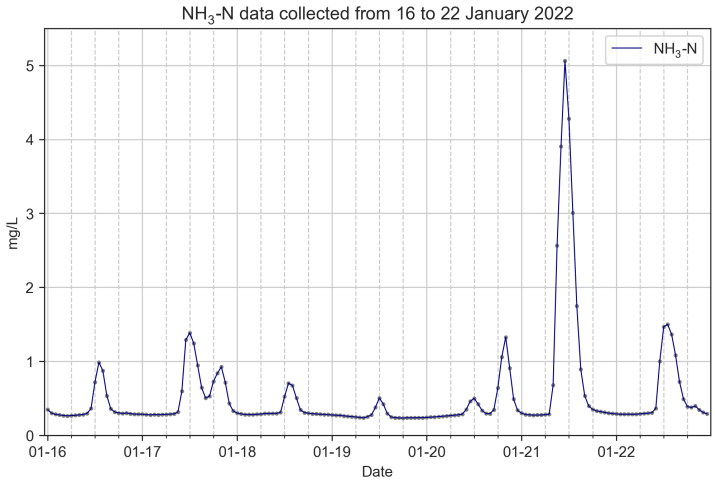
Empirically, the best-performed Model-Dataset combination should match the lowest test with the lowest validation loss values when using the same testing dataset to evaluate a group of models. For instance, the GRU-sg7 model in forecasting ammonia has the lowest test loss of 0.0383, yet the validation 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. 4.5, the fluctuation patterns of NH_3-N in validation dataset as in Fig. 4.5a is much resemble to the testing dataset from Fig. 4.5c compared to testing dataset from Fig. 4.5b. Further tests were carried out using a testing dataset from October to re-evaluate the model performance from Table. 4.1. 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.

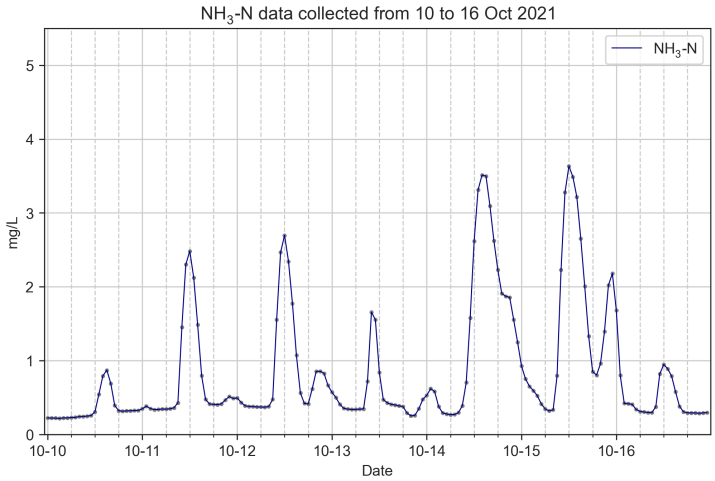
As shown in Table. 4.2, the models with the top lowest test loss values are 0.0158, 0.0161,



(a) Validation dataset from January 2022.



(b) Testing dataset from January 2022.



(c) Testing dataset from October 2021.

Figure 4.5: 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 4.2: 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. 4.3. 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-performanced 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. 4.2 and Table. 4.3. 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^{th}, 20^{th}, and 1^{st} 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 4.3: Baseline performance of the colour forecasting models, evaluated on test dataset from **16 to 22 Janurary 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	-	-	-

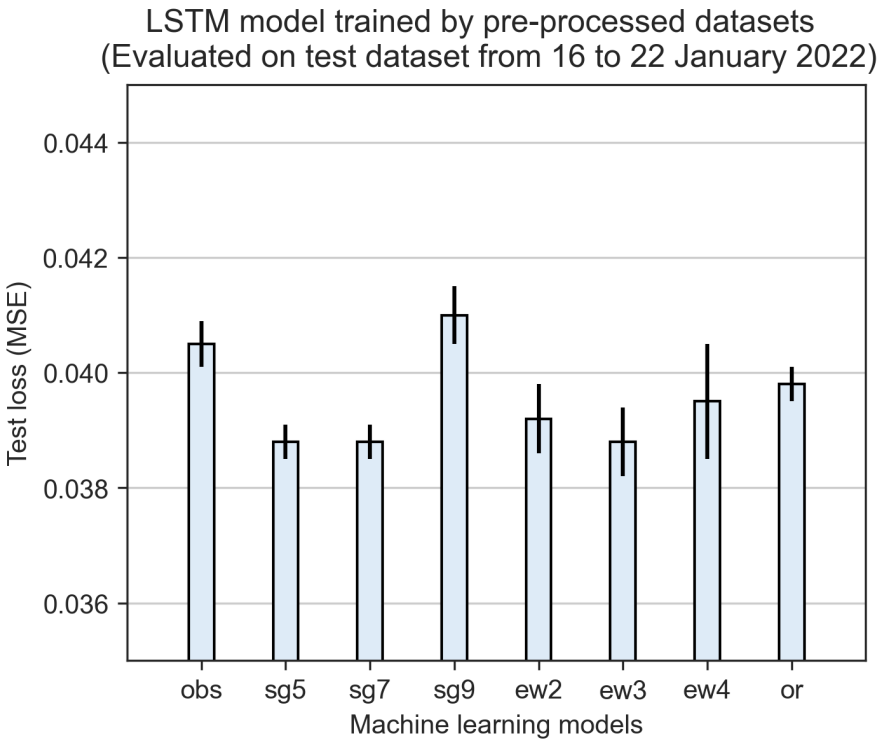
By comparing the baseline performance and the influences of data pre-processing techniques on machine learning models, our findings appear to be well substantiated by using LSTM models for training ammonia and colour forecasting models due to their outstanding

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.

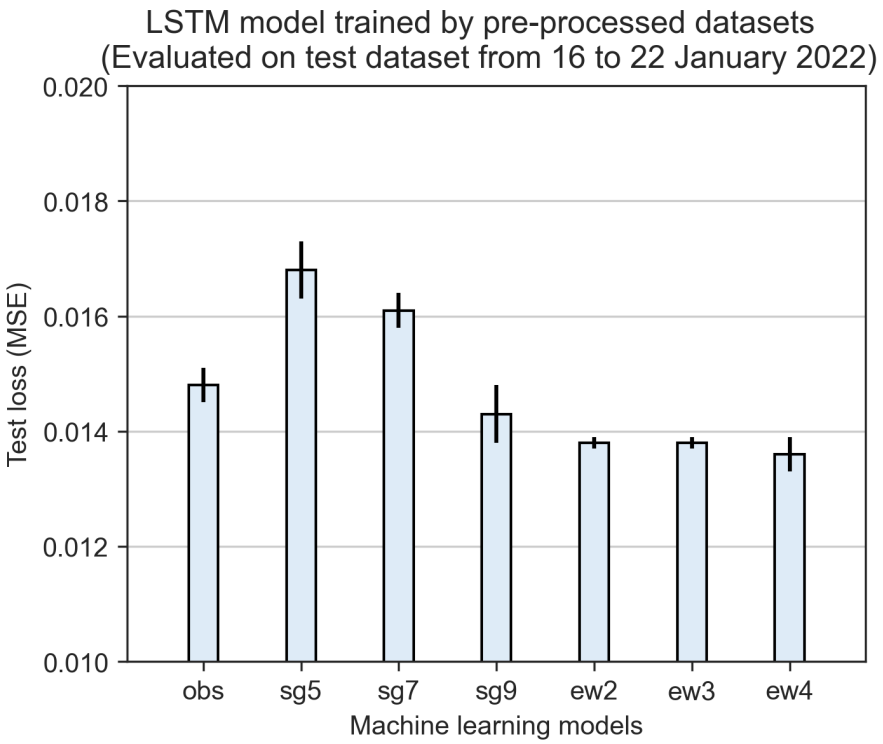
4.2.2 The effects of window sizes of the data smoothing filters

The influences of window sizes in the data smoothing process are investigated using LSTM models and illustrated in Fig. 4.6. Larger and smaller SG window sizes have different impacts on ammonia and colour forecasting models. In ammonia forecasting models, as shown in Fig. 4.6a, 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. 4.6b, 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 4.6: Baseline performance of the ammonia and colour forecasting models.

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

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

4.3.1 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. 3.19. 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. 4.7, 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. 4.7, 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 LSTM-4 $<$ LSTM-3 $<$ LSTM-2 $<$ 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. 4.7, 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.

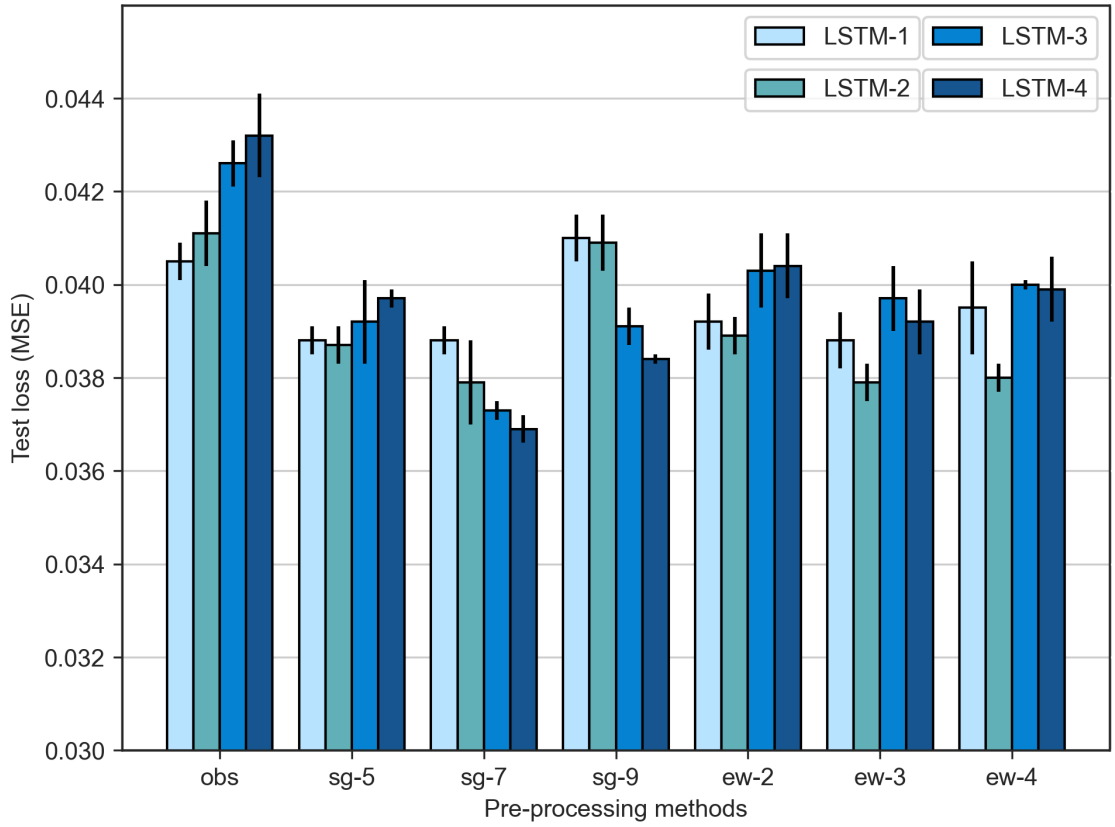


Figure 4.7: Comparisons of the model performance in forecasitng ammonia concentrations.

4.3.2 Colour forecasting models

As shown in Fig. 4.8, 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.

4.3.3 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. 4.9, the proposed model training methods helped the model to forecast better on 21 January as in Fig. 4.9b 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. 4.10, 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. 4.9. 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

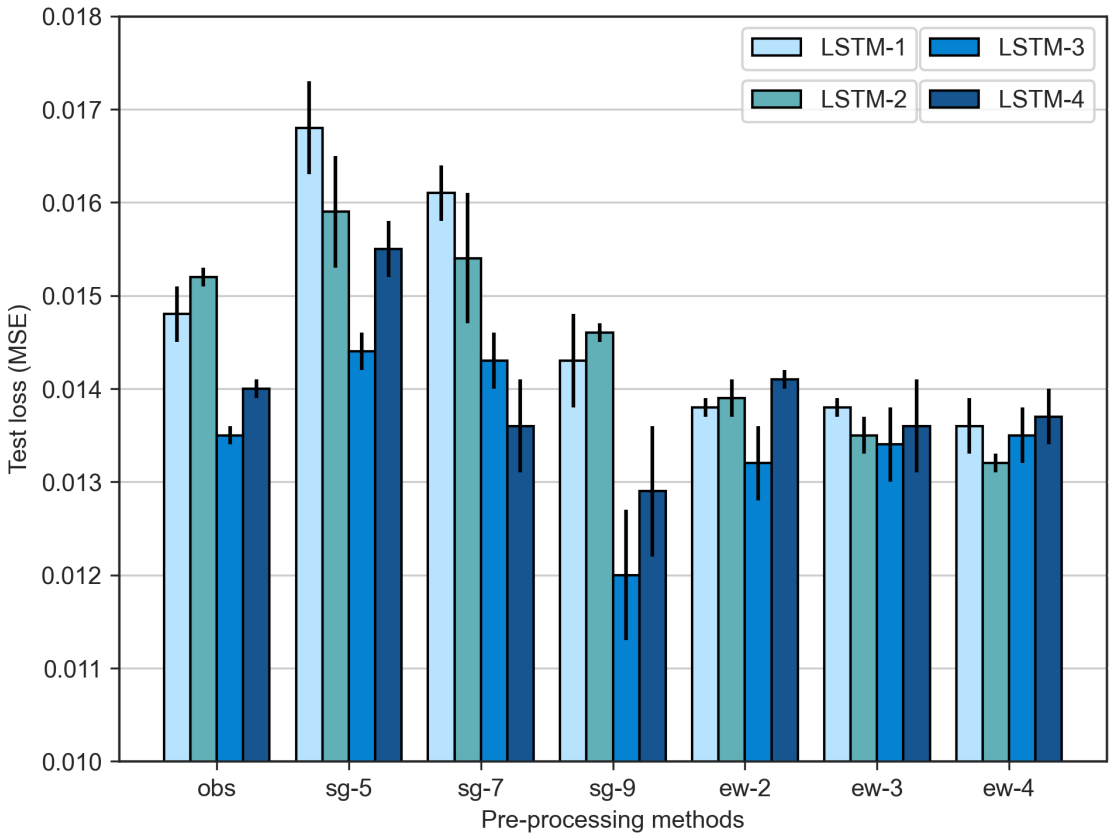
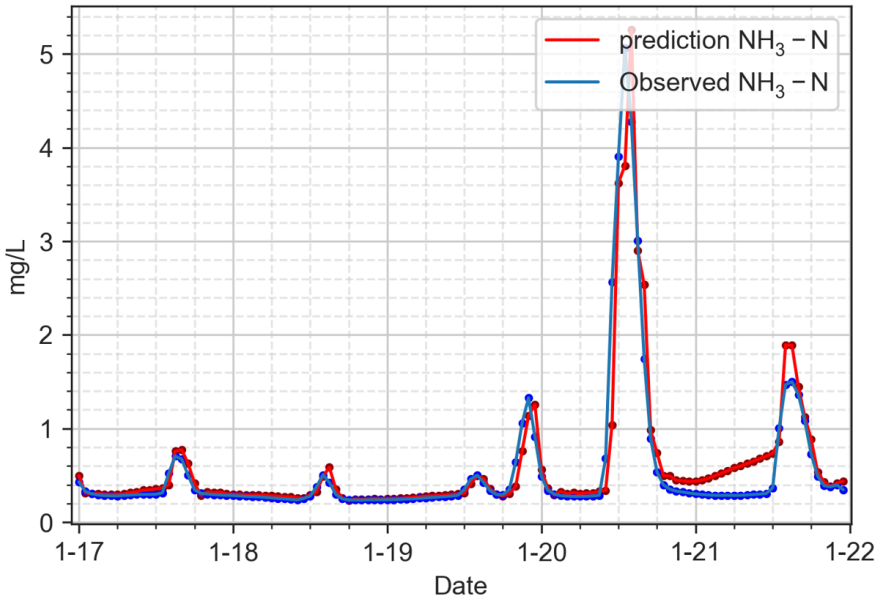


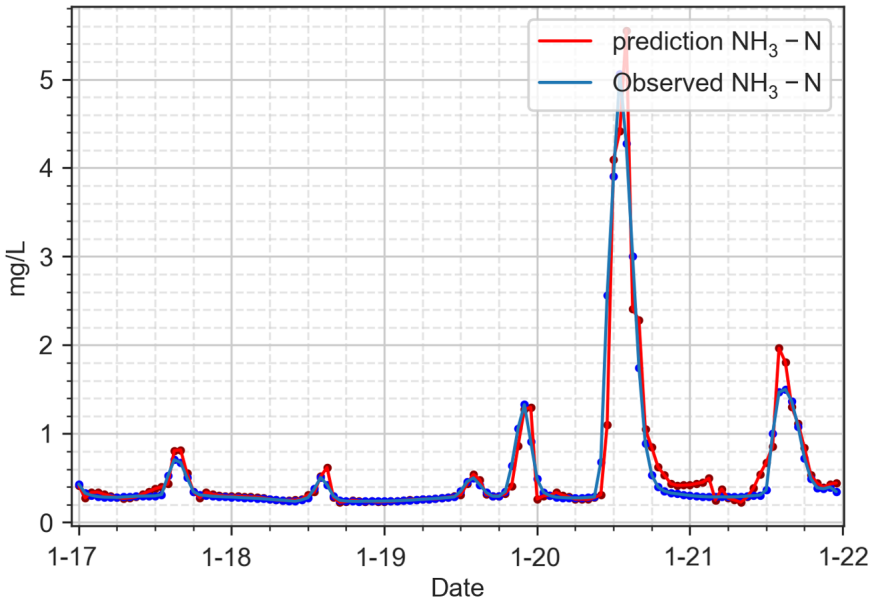
Figure 4.8: Comparisons of model performance in forecasting colour levels.

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. 4.10b 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

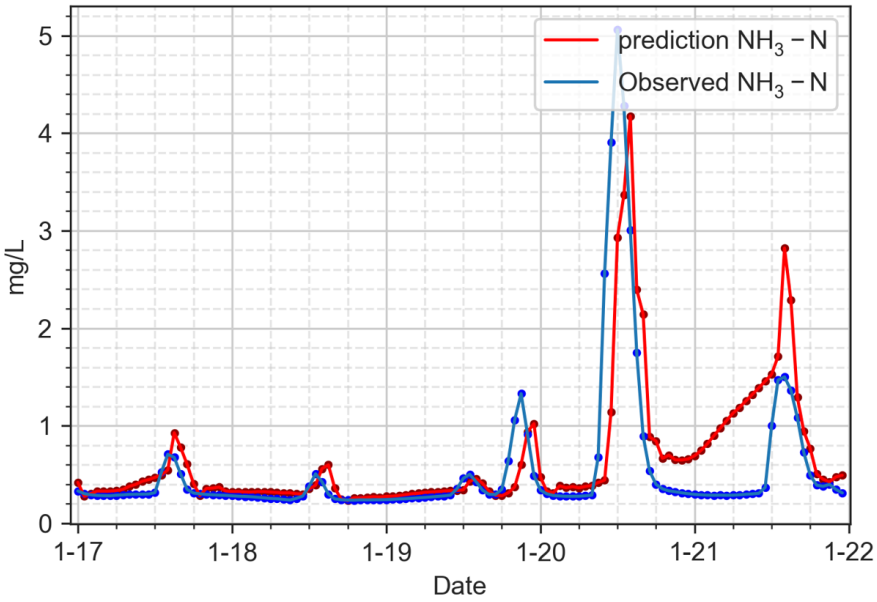


(b) LSTM-4-sg7, MSE = 0.0529

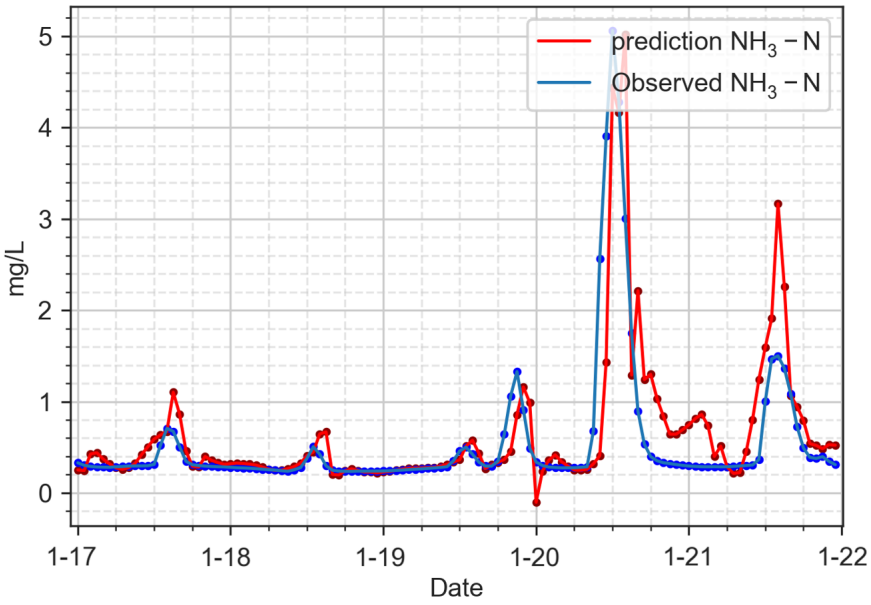
Figure 4.9: Visualization of the ammonia forecasting models at forecast horizon of one.

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 errours values of up to 5 Hazen Units and



(a) LSTM-1-obs, MSE = 0.2916

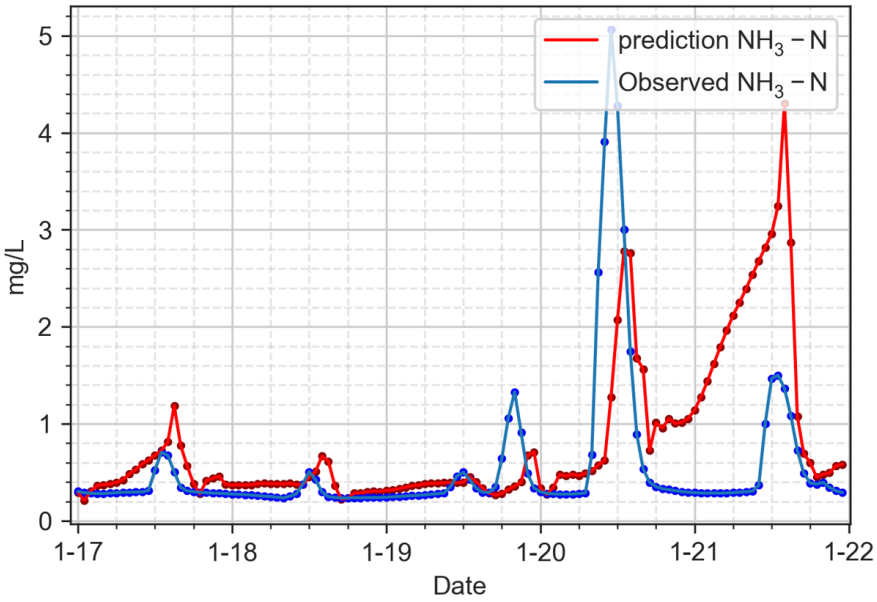


(b) LSTM-4-sg7, MSE = 0.2351

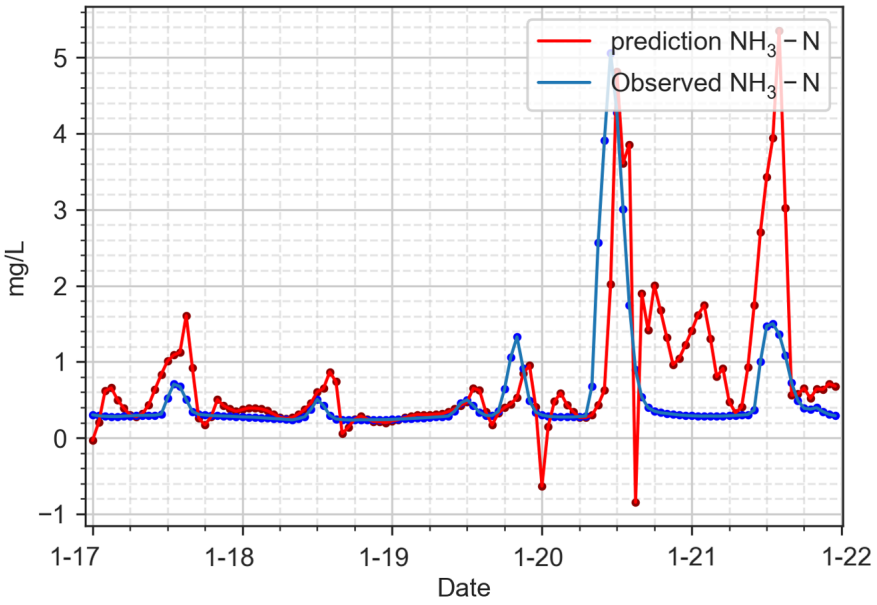
Figure 4.10: Visualization of the ammonia forecasting models at forecast horizon of two.

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. 4.13,



(a) LSTM-1-obs, MSE = 0.8025

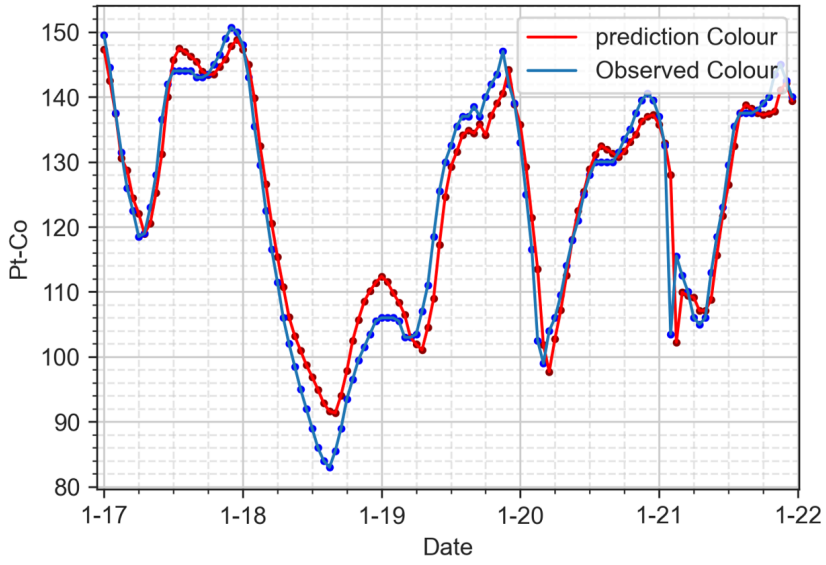


(b) LSTM-4-sg7, MSE = 0.7637

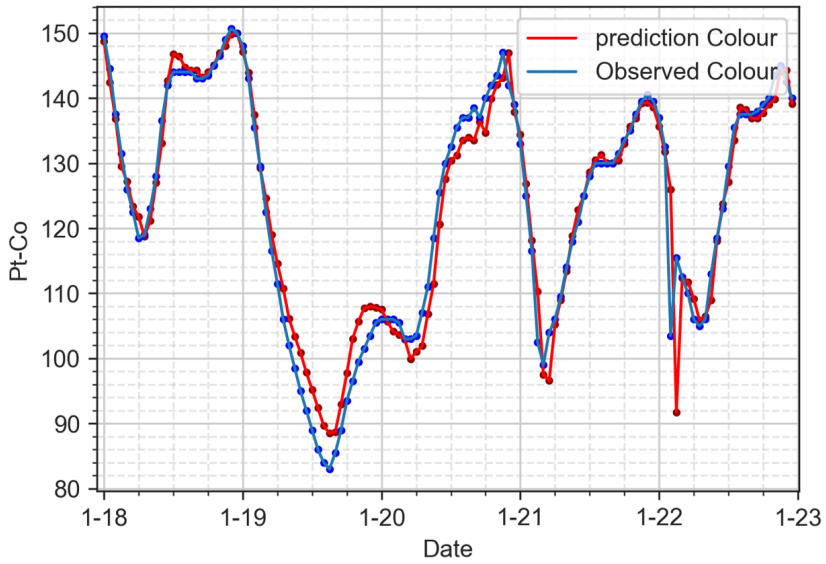
Figure 4.11: Visualization of the ammonia forecasting models at forecast horizon of three.

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



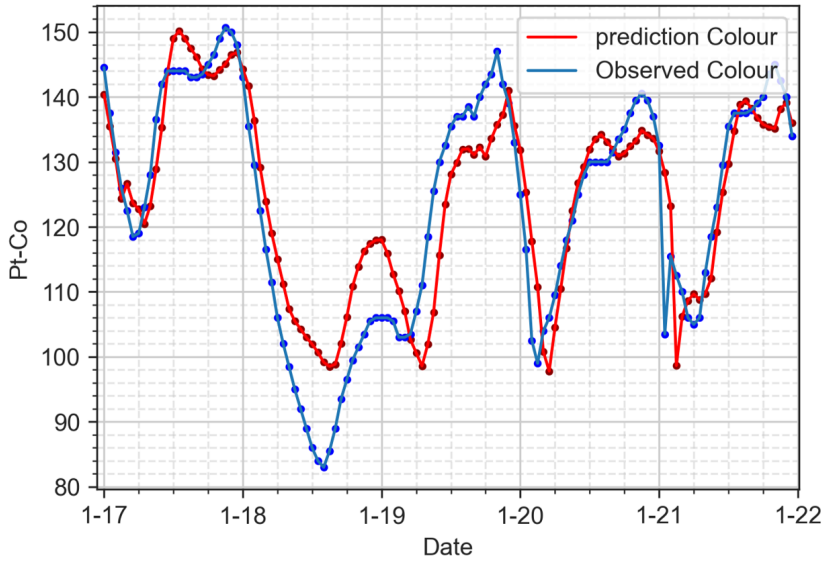
(a) LSTM-1-obs, MSE = 22.4922



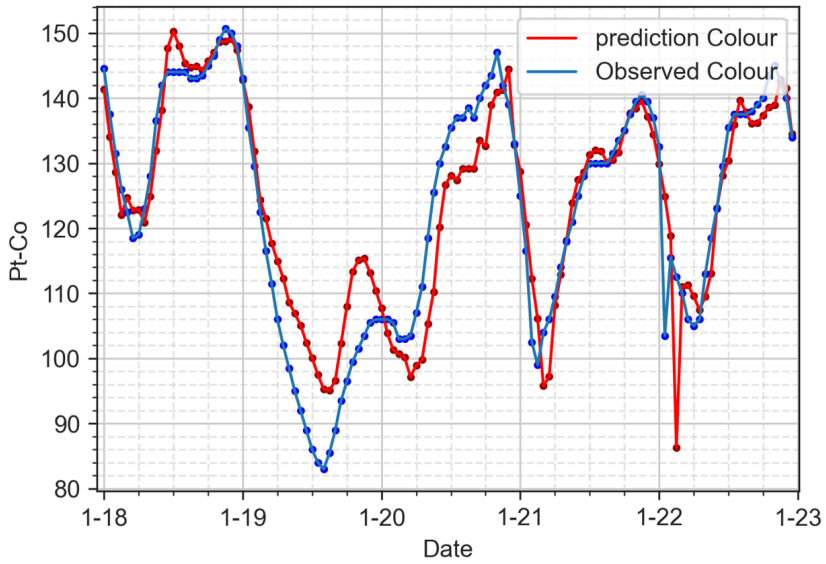
(b) LSTM-3-sg9, MSE = 17.5955

Figure 4.12: 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



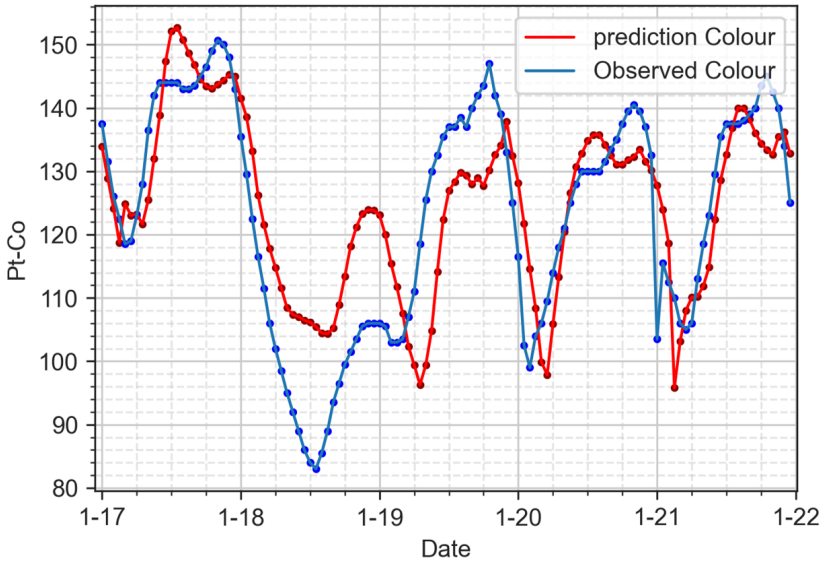
(a) LSTM-1-obs, MSE = 62.6678



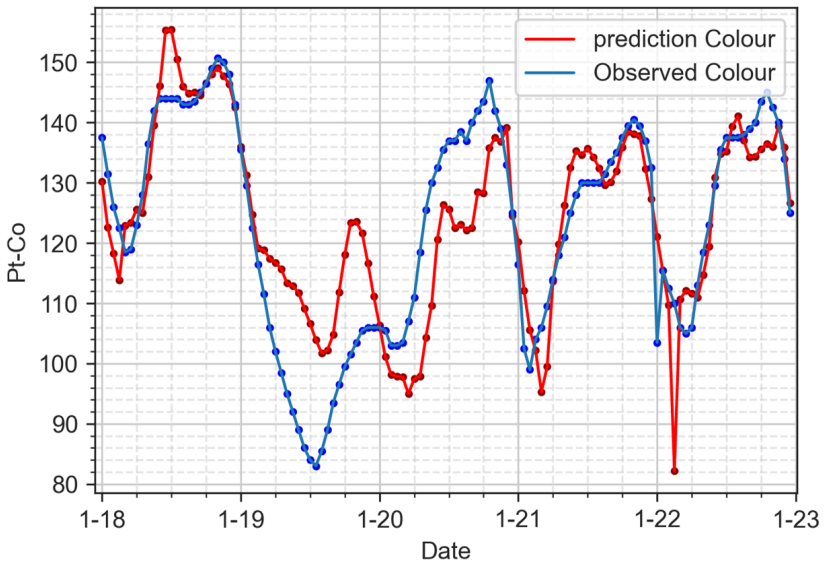
(b) LSTM-3-sg9, MSE = 47.4252

Figure 4.13: Visualization of the colour forecasting models at forecast horizon of two.

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



(a) LSTM-1-obs, MSE = 116.8928



(b) LSTM-3-sg9, MSE = 103.4329

Figure 4.14: Visualization of the colour forecasting models at forecast horizon of three.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

5.1.1 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.

5.1.2 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. 4.6. 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.

5.1.3 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.

5.2 Recommendations for future research

Due to the insufficient amount of ammonia and colour data, we cannot differentiate whether the undesired model performance was caused by the heterogeneity of the validation and testing datasets or caused by the pre-processing and feature engineering techniques we applied to the datasets. It is recommended a larger dataset (e.g., a larger dataset in length and better data quality with more input features for training) should be used in the future study when evaluating the proposed methods in this study. The insufficient data could also lead to the unstable performance of different models trained by the same data smoothing techniques. 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 in different ways 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.

All the forecasting models in this study only focus on predicting ammonia concentration and colour levels in the reclaimed water system. In future research, more water quality parameters should be included. In reclaimed water systems, the concentration of water quality parameters such as turbidity and E. coli are also regulated by Water Supply Department. Violating any water quality parameter will directly lead to the disqualification of being used as reclaimed water. Using more water quality parameters as features has extra benefits for building forecasting models. The hidden correlations between each water quality parameter will most likely help build more accurate water quality forecasting models.

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.

J.R. Adewumi, A.A. Ilemobade, and J.E. Van Zyl. Treated wastewater reuse in South Africa: Overview, potential and challenges. *Resources, Conservation and Recycling*, 55(2):221–231, December 2010. ISSN 09213449. doi: 10.1016/j.resconrec.2010.09.012.

Samarth Agrawal. Hyperparameters in Deep Learning, February 2019.

Ziad Al-Ghazawi and Rami Alawneh. Use of artificial neural network for predicting effluent quality parameters and enabling wastewater reuse for climate change resilience – A case from Jordan. *Journal of Water Process Engineering*, 44:102423, December 2021. ISSN 22147144. doi: 10.1016/j.jwpe.2021.102423.

Janelcy Alferes, Anders Lynggaard-Jensen, Thomas Munk-Nielsen, Sovanna Tik, Luca Vezzaro, Anitha Kumari Sharma, Peter Steen Mikkelsen, and Peter A. Vanrolleghem. Validating data quality during wet weather monitoring of wastewater treatment plant influents. *Proceedings of the Water Environment Federation*, 2013(12):4507–4520, January 2013. ISSN 19386478. doi: 10.2175/193864713813686060.

Abdalrahman Alsulaili and Abdelrahman Refaie. Artificial neural network modeling approach for the prediction of five-day biological oxygen demand and wastewater treatment plant performance. *Water Supply*, 21(5):1861–1877, August 2021. ISSN 1606-9749, 1607-0798. doi: 10.2166/ws.2020.199.

Sunil L Andhare and Prasad J Palkar. SCADA a tool to increase efficiency of water treatment plant. *Asian Journal of Engineering and Technology Innovation*, page 8, 2014.

Giulia Bachis, Thibaud Maruéjouls, Sovanna Tik, Youri Amerlinck, Henryk Melcer, Ingmar Nopens, Paul Lessard, and Peter A. Vanrolleghem. Modelling and characterization of primary settlers in view of whole plant and resource recovery modelling. *Water Science and Technology*, 72(12):2251–2261, December 2015. ISSN 0273-1223, 1996-9732. doi: 10.2166/wst.2015.455.

Jhon Stalin Figueroa Bados and Iralmy Yipsy Platero Morejon. Design of a PID Control System for a Wastewater Treatment Plant. In *2020 3rd International Conference on Robotics, Control and Automation Engineering (RCAE)*, pages 31–35, Chongqing, China, November 2020. IEEE. ISBN 978-1-72818-638-2. doi: 10.1109/RCAE51546.2020.9294199.

Nobel Ballhysa, Soyeon Kim, and Seongjoon Byeon. Wastewater Treatment Plant Control Strategies. *International journal of advanced smart convergence*, 9(4):16–25, December 2020. doi: 10.7236/IJASC.2020.9.4.16.

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

Adi Hasif bin Talib. *Modeling and Control of Wastewater Treatment Process*. PhD thesis, Universiti Teknologi Petronas, May 2011.

Sebastian Castro. Why should I choose matlab deep learning toolbox over other opensource frameworks like caffe, onnx, pytorch, torch etc?, October 2018.

Francesca Cecconi and Diego Rosso. Soft Sensing for On-Line Fault Detection of Ammonium Sensors in Water Resource Recovery Facilities. *Environmental Science: Water Research and Technology*, 2021. doi: 10.1021/acs.est.0c06111.

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

Varun Chandola. Anomaly Detection : A Survey. page 72, September 2009.

J.C. Chen, N.B. Chang, and W.K. Shieh. Assessing wastewater reclamation potential by neural network model. *Engineering Applications of Artificial Intelligence*, 16(2):149–157, March 2003. ISSN 09521976. doi: 10.1016/S0952-1976(03)00056-3.

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.

K. Chojnacka, A. Witek-Krowiak, K. Moustakas, D. Skrzypczak, K. Mikula, and M. Loizidou. A transition from conventional irrigation to fertigation with reclaimed wastewater: Prospects and challenges. *Renewable and Sustainable Energy Reviews*, 130:109959, September 2020. ISSN 13640321. doi: 10.1016/j.rser.2020.109959.

M. Colella, M. Ripa, A. Cocozza, C. Panfilo, and S. Ulgiati. Challenges and opportunities for more efficient water use and circular wastewater management. The case of Campania Region, Italy. *Journal of Environmental Management*, 297:113171, November 2021. ISSN 03014797. doi: 10.1016/j.jenvman.2021.113171.

C. De Mulder, T. Flameling, S. Weijers, Y. Amerlinck, and I. Nopens. An open software package for data reconciliation and gap filling in preparation of Water and Resource Recovery Facility Modeling. *Environmental Modelling & Software*, 107:186–198, September 2018. ISSN 13648152. doi: 10.1016/j.envsoft.2018.05.015.

DeepAI. Loss Function, June 2022.

Feridun Demir and Wilbur W. Woo. Feedback control over the chlorine disinfection process at a wastewater treatment plant using a Smith predictor, a method of characteristics and odometric transformation. *Journal of Environmental Chemical Engineering*, 2(2):1088–1097, June 2014. ISSN 22133437. doi: 10.1016/j.jece.2014.04.006.

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

Python Software Fundation. What is Python? Executive Summary, July 2022.

Javier Gamiz, Ramon Vilanova, Herminio Martinez-Garcia, Yolanda Bolea, and Antoni Grau. Fuzzy Gain Scheduling and Feed-Forward Control for Drinking Water Treatment Plants (DWTP) Chlorination Process. *IEEE Access*, 8:110018–110032, 2020. ISSN 2169-3536. doi: 10.1109/ACCESS.2020.3002156.

Paran Gani, Norshuhaila Mohamed Sunar, Hazel Matias-Peralta, and Ab Aziz Abdul Latiff. Effect of pH and alum dosage on the efficiency of microalgae harvesting via flocculation technique. *International Journal of Green Energy*, 14(4):395–399, March 2017. ISSN 1543-5075, 1543-5083. doi: 10.1080/15435075.2016.1261707.

Lluís Godo-Pla, Jose Javier Rodríguez, Jordi Suquet, Pere Emiliano, Fernando Valero, Manel Poch, and Hèctor Monclús. Control of primary disinfection in a drinking water treatment plant based on a fuzzy inference system. *Process Safety and Environmental Protection*, 145: 63–70, January 2021. ISSN 09575820. doi: 10.1016/j.psep.2020.07.037.

Hong Guo, Kwanho Jeong, Jiyeon Lim, Jeongwon Jo, Young Mo Kim, Jong pyo Park, Joon Ha Kim, and Kyung Hwa Cho. Prediction of effluent concentration in a wastewater treatment plant using machine learning models. *Journal of Environmental Sciences (China)*, 32:90–101, 2015. doi: 10.1016/j.jes.2015.01.007.

Henri Haimi, Francesco Corona, Michela Mulas, Laura Sundell, Mari Heinonen, and Riku Vahala. Shall we use hardware sensor measurements or soft-sensor estimates? Case study in a full-scale WWTP. *Environmental Modelling and Software*, 72:215–229, 2015. doi: 10.1016/j.envsoft.2015.07.013.

Sung-Taek Hong, An-Kyu Lee, Ho-Hyun Lee, No-Suk Park, and Seung-Hwan Lee. Application of neuro-fuzzy PID controller for effective post-chlorination in water treatment plant. *Desalination and Water Treatment*, 47(1-3):211–220, September 2012. ISSN 1944-3994, 1944-3986. doi: 10.1080/19443994.2012.696810.

IBM. Neural Networks, June 2022.

Philipp Kehrein, Mark van Loosdrecht, Patricia Osseweijer, Marianna Garfí, Jo Dewulf, and John Posada. A critical review of resource recovery from municipal wastewater treatment plants – market supply potentials, technologies and bottlenecks. *Environmental Science: Water Research & Technology*, 6(4):877–910, 2020. ISSN 2053-1400, 2053-1419. doi: 10.1039/C9EW00905A.

Edmund A. Kobylinski, Gary L. Hunter, and Andrew R. Shaw. On Line Control Strategies for Disinfection Systems: Success and Failure. *Proceedings of the Water Environment Federation*, 2006(5):6371–6394, January 2006. ISSN 1938-6478. doi: 10.2175/193864706783761716.

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.

Lei Li, Shuming Rong, Rui Wang, and Shuili Yu. Recent advances in artificial intelligence and machine learning for nonlinear relationship analysis and process control in drinking water treatment: A review. *Chemical Engineering Journal*, 405:126673, February 2021. ISSN 13858947. doi: 10.1016/j.cej.2020.126673.

Peifeng Li, Jin Zhang, and Peter Krebs. Prediction of Flow Based on a CNN-LSTM Combined Deep Learning Approach. *Water*, 14(6):993, March 2022. ISSN 2073-4441. doi: 10.3390/w14060993.

Zhe Li, Caiwen Ding, Siyue Wang, Wujie Wen, Youwei Zhuo, Chang Liu, Qinru Qiu, Wenyao Xu, Xue Lin, Xuehai Qian, and Yanzhi Wang. E-RNN: Design Optimization for Efficient Recurrent Neural Networks in FPGAs, December 2018.

André Felipe Librantz, Fábio Cosme Rodrigues dos Santos, and Cleber Gustavo Dias. Artificial neural networks to control chlorine dosing in a water treatment plant. *Acta Scientiarum. Technology*, 40(1):37275, September 2018. ISSN 1807-8664, 1806-2563. doi: 10.4025/actascitechnol.v40i1.37275.

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.

Sidan Lyu, Weiping Chen, Weiling Zhang, Yupeng Fan, and Wentao Jiao. Wastewater reclamation and reuse in China: Opportunities and challenges. *Journal of Environmental Sciences*, 39:86–96, January 2016. ISSN 10010742. doi: 10.1016/j.jes.2015.11.012.

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.

Giorgio Mannina, Taise Ferreira Rebouças, Alida Cosenza, Miquel Sànchez-Marrè, and Karina Gibert. Decision support systems (DSS) for wastewater treatment plants – A review of the state of the art. *Bioresource Technology*, 290:121814, October 2019. ISSN 09608524. doi: 10.1016/j.biortech.2019.121814.

Bareera Maryam and Hanife Büyükgüngör. Wastewater reclamation and reuse trends in Turkey: Opportunities and challenges. *Journal of Water Process Engineering*, 30:100501, August 2019. ISSN 22147144. doi: 10.1016/j.jwpe.2017.10.001.

MathWorks. Call Python Function Using MATLAB Function and MATLAB System Block, April 2022a.

MathWorks. Documentation-Findpeaks, June 2022b.

MathWorks. MATLAB for Machine Learning, June 2022c.

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.

Kathryn B. Newhart, Ryan W. Holloway, Amanda S. Hering, and Tzahi Y. Cath. Data-driven performance analyses of wastewater treatment plants: A review. *Water Research*, 157:498–513, June 2019. ISSN 00431354. doi: 10.1016/j.watres.2019.03.030.

Diana Norton-Brandão, Sigrid M. Scherrenberg, and Jules B. van Lier. Reclamation of used urban waters for irrigation purposes – A review of treatment technologies. *Journal of Environmental Management*, 122:85–98, June 2013. ISSN 03014797. doi: 10.1016/j.jenvman.2013.03.012.

Christopher Olah. Understanding LSTM Networks, August 2015.

Bhawani Shankar Pattnaik, Arunima Sambhuta Pattanayak, Siba Kumar Udgata, and Ajit Kumar Panda. Machine learning based soft sensor model for BOD estimation using intelligence at edge. *Complex & Intelligent Systems*, 7(2):961–976, 2021. doi: 10.1007/s40747-020-00259-9.

Ivan Pisa, Ignacio Santin, Antoni Morell, Jose Lopez Vicario, and Ramon Vilanova. LSTM-Based Wastewater Treatment Plants Operation Strategies for Effluent Quality Improvement. *IEEE Access*, 7:159773–159786, 2019. doi: 10.1109/ACCESS.2019.2950852.

Janosh Riebesell. Random Forest, June 2022.

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

C. Rosen, L. Rieger, U. Jeppsson, and P. A. Vanrolleghem. Adding realism to simulated sensors and actuators. *Water Science and Technology*, 57(3):337–344, February 2008. ISSN 0273-1223, 1996-9732. doi: 10.2166/wst.2008.130.

I. Santín, C. Pedret, and R. Vilanova. Fuzzy Control and Model Predictive Control Configurations for Effluent Violations Removal in Wastewater Treatment Plants. *Industrial & Engineering Chemistry Research*, 54(10):2763–2775, March 2015. ISSN 0888-5885, 1520-5045. doi: 10.1021/ie504079q.

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.

Matthew Stevenson and Cristián Bravo. Advanced turbidity prediction for operational water supply planning. *Decision Support Systems*, 119:72–84, April 2019. ISSN 01679236. doi: 10.1016/j.dss.2019.02.009.

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

TheWorldBank. Circular Economy: An Opportunity to Transform Urban Water Services, September 2021.

UNICEF. URBAN WATER SCARCITY GUIDANCE NOTE: PREVENTING DAY ZERO. Technical report, March 2021.

Nguyen Duc Viet, Duksoo Jang, Yeomin Yoon, and Am Jang. Enhancement of membrane system performance using artificial intelligence technologies for sustainable water and wastewater treatment: A critical review. *Critical Reviews in Environmental Science and Technology*, pages 1–31, June 2021. ISSN 1064-3389, 1547-6537. doi: 10.1080/10643389.2021.1940031.

Dong Wang, Sven Thunéll, Ulrika Lindberg, Lili Jiang, Johan Trygg, Mats Tysklind, and Nabil Souihi. 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.

Dongsheng Wang and Hao Xiang. Composite Control of Post-Chlorine Dosage During Drinking Water Treatment. *IEEE Access*, 7:27893–27898, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2901059.

Dongsheng Wang, Jingjin Shen, Songhao Zhu, and Guoping Jiang. Model predictive control for chlorine dosing of drinking water treatment based on support vector machine model. *DESALINATION AND WATER TREATMENT*, 173:133–141, 2020. doi: 10.5004/dwt.2020.24144.

Hui Wang, Tirusew Asefa, and Jack Thornburgh. Integrating water quality and streamflow into prediction of chemical dosage in a drinking water treatment plant using machine learning algorithms. *Water Supply*, 22(3):2803–2815, March 2022. ISSN 1606-9749, 1607-0798. doi: 10.2166/ws.2021.435.

Xiaodong Wang, Knut Kvaal, and Harsha Ratnaweera. Explicit and interpretable nonlinear soft sensor models for influent surveillance at a full-scale wastewater treatment plant. *Journal of Process Control*, 77:1–6, 2019. doi: 10.1016/j.jprocont.2019.03.005.

Wikipedia. Random forest, June 2022a.

Wikipedia. Savitzky–Golay filter, June 2022b.

Britt-Marie Wilén, Raquel Liébana, Frank Persson, Oskar Modin, and Malte Hermansson. The mechanisms of granulation of activated sludge in wastewater treatment, its optimization, and impact on effluent quality. *Applied Microbiology and Biotechnology*, 102(12):5005–5020, June 2018. ISSN 0175-7598, 1432-0614. doi: 10.1007/s00253-018-8990-9.

World Health Organization. Water quality and health - review of turbidity: Information for regulators and water suppliers. Technical report, World Health Organization, Geneva, 2017.

Jianlong Xu, Zhuo Xu, Jianjun Kuang, Che Lin, Lianghong Xiao, Xingshan Huang, and Yufeng Zhang. An Alternative to Laboratory Testing: Random Forest-Based Water Quality Prediction Framework for Inland and Nearshore Water Bodies. *Water*, 13(22):3262, November 2021. ISSN 2073-4441. doi: 10.3390/w13223262.

Zeqiong Xu, Jiao Shen, Yuqing Qu, Huangfei Chen, Xiaoling Zhou, Huachang Hong, Hongjie Sun, Hongjun Lin, Wenjing Deng, and Fuyong Wu. Using simple and easy water quality parameters to predict trihalomethane occurrence in tap water. *Chemosphere*, 286:131586, January 2022. ISSN 00456535. doi: 10.1016/j.chemosphere.2021.131586.

Mohamed Sherif Zaghloul, Oliver Terna Iorhemen, Rania Ahmed Hamza, Joo Hwa Tay, and Gopal Achari. Development of an ensemble of machine learning algorithms to model aerobic

granular sludge reactors. *Water Research*, 189:116657–116657, 2021. doi: 10.1016/j.watres.2020.116657.

Hongqiu Zhu, Qiling Wang, Fengxue Zhang, Chunhua Yang, and Yonggang Li. A prediction method of electrocoagulation reactor removal rate based on Long Term and Short Term Memory - Autoregressive Integrated Moving Average Model. *Process Safety and Environmental Protection*, 152:462–470, 2021. doi: 10.1016/j.psep.2021.06.020.

Huijun Zhu and Xinglei Qiu. The Application of PLC in Sewage Treatment. *Journal of Water Resource and Protection*, 09(07):841–850, 2017. ISSN 1945-3094, 1945-3108. doi: 10.4236/jwarp.2017.97056.

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
```