ACIT5900

MASTER THESIS

in

Applied Computer and Information Technology (ACIT)

May 2023

Robotics and Control

Virtual measurements in Wastewater treatment plant: Machine learning models for predicting the PO4 concentration in the effluent

Mohamed Abdishakur Mohamed

Department of ACIT

Faculty of Technology, Art and Design

OSLOMET

Preface

In the Name of Allah, the Most Gracious, the Most merciful. Praise be to Allah, the lord for all worlds, and may peace and blessings be upon His Messenger, Muhammad (peace be upon him).

It is with immense gratitude and humble commitment that I present this work, "*Virtual measurement in wastewater treatment plant: Machine learning models for predicting the PO4 concentration in the effluent",* as part of my master 's thesis. This has been a journey of learning, discovery, and growth, and I acknowledge the blessings and guidance from Allah the Almighty throughout this project.

The aim of this master's thesis is to develop a comprehensive machine learning algorithms for effluent quality control for HIAS wastewater treatment plant. The study includes various machine learning algorithms. The aim is to assess effectiveness of these models in accurately predicting effluent parameter values and classifying effluent quality based on time-series historical dataset, from HIAS wastewater treatment plant. The work involves data preprocessing, model training, evaluation and validation through evaluation metrics.

I would like to express my sincere appreciation to my supervisors, Arvind Keprate, Abhilash Nair and Tiina Komulainen, for their scholarly expertise and constant support. Their insightful feedback, and continuous support, and patient mentoring have been instrumental in shaping this work. I would also like to offer my sincere thanks to the members of my thesis group for their expertise, time, and valuable assistance in evaluating and reviewing this work. Their suggestions and constructive comments have greatly developed the quality of the research.

Furthermore, I would like to express my deep gratitude to my family and friends for their support. Especially my parents, where their unwavering encouragement , love and understanding throughout this academic journey. Their moral support and prayers were a source of motivation for me. Thanks for believing in me.

Finally, I acknowledge that this work would not have been possible without the guidance and blessing from Allah the Almighty. I am grateful for the knowledge granted to me. May this work serve as a means of seeking knowledge, benefitting humanity, and earning the pleasure of Allah, the most High.

Table of Contents

List of Figures

List of Tables

Abstract

Due to the underlying complexity of wastewater treatment plant (WWTP) processes, it might be challenging to respond appropriately and promptly to the dynamic process conditions in order to ensure the quality of the effluent, particularly when operational cost are a major consideration. In order to avoid various limitations of conventional mechanistic models, machine learning (ML) methods have been utilized to model WWTP processes. Additionally, the time lags between process steps have been neglected, making it difficult to explain the relationships between operational factors and effluent quality. Therefore, in this study multiple machine learning methods were developed to improve effluent quality control in WWTPs by clarifying the relationships between operational parameters and effluent parameters. To be more specific, the objective in this study is to predict the concentration of phosphate (PO4) in the effluent of HIAS wastewater treatment plant.

In this study, machine learning algorithms for effluent quality control in WTTPs is proposed. The various ML algorithms consist of Regression models (Linear Regression, Lasso Regression, and Ridge Regression), data-driven multi-class classification models (Decision Trees, Gradient Boosting Decision Tree, and XGBoost), and Long Short-Term Memory (LSTM) model specifically designed for time-series data analysis.

The dataset utilized in this study, contains time-series data, historical operational variables and effluent parameters from HIAS wastewater treatment plant in Hamar, involving decent size of samples (8662). One effluent parameter, Phosphate in effluent (PO4), and 19 operational parameters are studied. The data preprocessing method used to prepare it for ML models, includes handling missing values and outliers to ensure reliable and consistent analysis. The ML models are trained, validated, and evaluated using appropriate evaluation metrics, such as R-squared Mean Error (RSME), Mean Absolute Error, and Mean Squared Error (MSE) to assess the performance and effectiveness of the machine learning models.

The results demonstrated the effectiveness of the ML models in improving effluent quality control. Among the regression model, Linear and Ridge regression performed best, achieving a moderate fit with an R2 score 0.527. Lasso Regression demonstrated very poor and weak performance . In terms of data-driven muti-class classification, Gradient Boosting Decision Tree model outperformed the other classification models with an R2 score of 0.869, indicating a good fit. The LSTM model displayed significant promise in accurately predict the PO4 concentration in the effluent among the ML models utilized in this study , achieving an substantial fit with an R2 score of 0.926. These results could support the development of more advanced control strategies to increase the impact on PO4 removal.

Keywords **-** ***Wastewater treatment, effluent control, Prediction model, Machine learning, LSTM, Regression, data-driven multi-class classification, Data preprocessing, Time-series dataset.***

1 INTRODUCTION

1.1 General

Due to its severe effects on the environment and human health, the issue of wastewater disposal has grown to be a significant matter on a global scale. As treatment plant play a significant part in wastewater management, they should be maintained effectively. Since the entire system must be taken into account, years of data must be examined and processed in order to develop a sufficient foundation for performance evaluation. The performance of the WWTP can be predicted utilizing important pollution variables and machine learning models. Certain key parameters in a WWTP can be used to evaluate plant performance. Chemical oxygen demand (COD), suspended solids (SS) and phosphate (PO4) are examples of these parameters. However the method for monitoring the effluent of the plant requires an understanding of the plant's performance as well as the variables impacting the water parameters, such as season, time and people's lifestyle. (Raed Jafar, 2022)

The Moving Bed Biofilm reactor process of a wastewater treatment plant is complex due to the complex nature of the treatment process, different flow rates and the changes in the composition of raw wastewater. Also, the effectiveness of controlling the quality of wastewater discharge is decreased by the lack of continuous monitoring of the pollutants variables. Traditional modelling have its limitations, since microbial reactions in conjunction with environmental interactions are time-variable, nonlinear, and complicated. Utilizing virtual measurement as a tool for discovering complex dependencies between process variables and identifying the system behavior of the wastewater treatment plant can be an efficient method to handle this task, where data is analyzed and the inter-relationship of process variables in real Enhanced biological phosphate removal wastewater treatment plant is diagnosed. (Raed Jafar, 2022)

Since virtual measurement can provide solution for monitoring of a wastewater treatment plant, one should know what exactly virtual measurement is. Rather than using physical sensors and instruments, virtual measurement refers to a type of measurement that is performed using software simulations or models. In virtual measurement, the behavior of a system or process is simulated using a software programs, and the result of the simulation are analyzed to obtain measurements of various parameters of interest. Virtual measurement is commonly used in scientific

applications and engineering, where it can provide efficient and cost-effective ways to analyze complex processes and systems. Virtual measurement can also be used in research, where it can be used to simulate test hypotheses and experiments in a controlled manner. Since conducting experiments can be time-consuming, expensive and dangerous, virtual measurement can be particularly useful in fields such as chemistry and physics. (Maddi Etxegarai, 2022)

The study of virtual measurements and predicting the PO4 concentration in the effluent of HIAS wastewater treatment plant is the subject of this thesis. As well as the development and implementation of virtual measurement methods, the analysis of the data gathered, and the identification of key parameters that affects the performance of the system. The study will explore the potential of machine learning methods as the virtual measurement to optimize the quality control of phosphate in effluent of the process based on the data collected through sensor measurements.

The findings of this thesis will contribute to enhancing the efficacy of the wastewater treatment process through machine learning and provide a better understanding of how to improve the monitoring and control in the effluent. The thesis will be a valuable resource for academics and professionals working in the wastewater treatment plant.

1.2 Process description

The HIAS process is a single continuous reactor, multistage, enhanced biological phosphorus removal (EBPR), moving bed biofilm reactor (MBBR), wastewater treatment plant (WWTP). The movement of biofilms between anaerobic and aerobic zones for phosphorus removal and accumulation presents a problem for EBPR. In aerobic circumstances, phosphorus-accumulating organisms (PAO) in EBPR systems take up and store phosphorus as polyphosphate (polyP). During anaerobic conditions, the organisms aggressively incorporates reduced substances like short chain fatty acids (SCFAs), amino acids, and transforms them into polymers like polyhydroxyalkanoate (PHA) at the expense of energy stored as polyP. When the PAOs are cycled between anaerobic and aerobic conditions, these processes are energetically advantageous in phosphate and SCFA rich environments. (Didrik Villard, 2022)

The HIAS process uses active transport of biofilm carriers and wastewater across ten basins to meet the anaerobic and aerobic requirements. The first three being anaerobic and the follow seven being aerobic. Wastewater and biofilm carriers are combined in the

mechanically agitated anaerobic phase of the reactor before moving by gravity through three anaerobic and seven aerobic zones. The reactor's treated wastewater and sloughed-off biofilms exit the tenth zone, while the carriers of the biofilms are transported dry back to the reactor's beginning. (Didrik Villard, 2022)

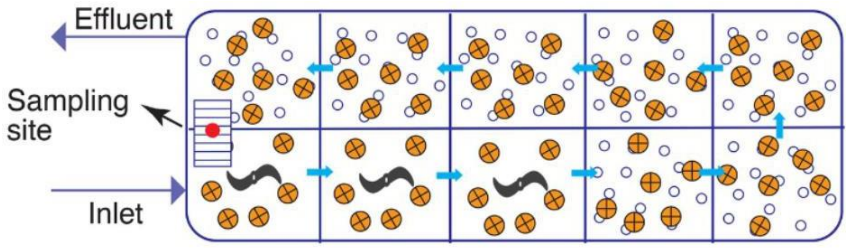

Figure 1, Schematic of the Hias EBPR wastewater treatment process. (Didrik Villard, 2022)

2 State of the Art

Machine learning based methods have been applied in multiple fields, particularly environmental issues. In order to improve the prediction of the treatment process machine learning were employed to the wastewater treatment plants. Where some key variables was used to evaluate the performance of the treatment plant, such as chemical oxygen demand (COD), biological oxygen demand (BOD), nitrates (NO_3) and phosphate (PO_4). The paper (Raed Jafar, 2022) showed that the use of machine learning models can provide an effective tool for modeling the complex processes of a treatment plant. (Raed Jafar, 2022), also refer to several studies that have employed machine learning. One of these studies available mentioned in the research paper evaluated the performance using machine leaning in one the wastewater treatment plant in Kuwait city, also known as Al-Ardiya. The results demonstrated that machine learning provide a flexible tool for modeling the wastewater treatment plants. (Raed Jafar, 2022) mentions that there were two model constructed using artificial neural network (ANN) to predict the biological oxygen concentration at the influent and the effluent of the Govindpura sewage treatment plant in Bhopal. This resulted in 80% removal of biological oxygen demand (BOD). In an another procedure machine learning was employed for prediction of wastewater treatment plant, where the objective was to predict the treatment efficiency and the effectiveness of input parameters on predicting the wastewater treatment plant.

They came to the conclusion that combining the input variables for the inflow rate, the effluent of total suspended solids and mixed liquor suspended solids produced the optimal model. the results were R2=0.898, and MSE = 0.443. (Raed Jafar, 2022)

In order to overcome the limitations of traditional mechanistic models, (Dong Wang S. T., 2021) used machine learning methods to model the wastewater treatment plant processes. To optimize effluent quality control in WWTPs, they put forward an innovative Machine learning context based on Random Forest, Deep Neural Network, partial dependence plot (PDP) and variable importance measure (VIM). The suggested machine learning framework seems to have the possible improvement of effluent quality management approaches at Umeå WWTP in Sweden. (Dong Wang S. T., 2021)

In one of Dong et al research their aim was to develop a machine learning-based method for controlling the quality of treated wastewater in treatment plants . The study's findings recognizes the increasing importance of ensuring that discharge from treatment plants meets the stringent quality standards set by the agencies. Traditional control methods have shown to be insufficient to meeting these standards, leading to the need for newer approaches. (Dong Wang S. T., 2021)

The authors explain a machine learning framework that makes use of sensors and processed data for treatment plant to create prediction models that can accurately estimate the effluent quality. The models can be used to optimize the treatment process in real-time and are trained using historical data. The authors also highlight the importance of feature selection, which is the process of identifying the most important variables that contribute to the quality of the effluent, as a critical component of the machine learning framework. (Dong Wang S. T., 2021)

The study provides data and evidence that effluent quality control in wastewater treatment facilities can be greatly enhanced by machine learning approaches. The findings demonstrate that the framework is capable of making accurate predictions about the effluent quality, and that these predictions are significantly more accurate than those using traditional control approaches. The authors further show that the framework is capable of handling non-linear and complex relationships between the different variables in treatment processes. (Dong Wang S. T., 2021)

Quang et al published a paper where the study aimed to explore the potential of machine learning algorithms for predicting the quality of wastewater from different full-scale wastewater treatment plant based on big data analysis.

The Authors proceeded by discussing the importance of predicting and monitoring wastewater quality in order to ensure the efficient functioning of WWTPs and environmental protection. They also pointed out that traditional methods for analyzing wastewater quality require a considerable amount of time and effort, which makes them inefficient for real-time prediction and monitoring. There, authors suggested using machine learning algorithms as a potential solution to this problem. (Quang Viet Ly, 2022)

The authors collected a large dataset consisting of 25 032 samples of influent and effluent wastewater quality parameters from six different wastewater treatment plant in Vietnam. The authors used the dataset containing various parameters such as oxygen demand (COD), ammonia nitrogen (NH3-N), total suspended solids (TSS), biochemical oxygen demand (BOD) to train and test several machine learning algorithms, including support vector regression, random forest, and artificial neural networks.

The result showed that all machine learning algorithms performed well in predicting the wastewater quality parameters, with artificial neural networks outperforming the other two algorithms. The study also found that that the accuracy of machine learning algorithms improved when more data were used for training. The authors concluded that machine learning algorithms can be used effectively for predicting wastewater quality from full-scale wastewater treatment plant based on big data analysis. (Quang Viet Ly, 2022)

In the literature, it widely recognized that the management and operation most wastewater treatment plant rely mostly on online monitoring instruments, combined with the professional experience and knowledge to evaluate the primary indicators of water quality in wastewater treatment plant and sewage treatment, which is risky given the increasingly demanding sewage discharge standards. There numerous studies on assisted treatment plant operation utilizing machine learning methods have been

conducted in recent years in order to decrease the risk of over-discharge of wastewater treatment plant and improve the efficacy of sewage treatment. (Rui Wang, 2021)

Previous research has shown that the influent quality was primary used to predict effluent quality. Concurrently, the main operating parameters of the WWTPs, such dissolved oxygen (DO), MLSS and sludge retention time (SRR), are all in the process of dynamic change and adjustment during daily administration and operation. It is well recognized that process parameters and water quality indicators have internal logical relationships.

The complexity of wastewater treatment has previously been summarized by (XIN LIU, 2021) using mechanism models, such as Activated Sludge Model No.1 (ASM1) and Activated Sludge Model No.2 (ASM2). Then, scholars and related organizations developed the activated sludge treatment benchmark simulation model (BSM1), which can monitor important features of wastewater, in order to objectively evaluate the performance of the wastewater treatment control strategy. While it may provide better experimental results, user need to know the expertise of the various systems in advance (XIN LIU, 2021). Additionally, these models are developed implementing specific circumstances into consideration. The generality of the models is limited by the numerous adjustments and tests needed to apply them in different situations. Data driven models can be developed through data and algorithms, in contrast to mechanistic modelling, which indicates that they do not require a thorough understanding of the process's mechanism. Furthermore, WWTPs collect, store and monitor a significant amount of data during daily operations, which makes data-driven models more practical in applications.

A neural network is a data driven model that imitates the structure of biological neurons (XIN LIU, 2021). Neural networks have been gradually incorporated into the wastewater treatment industry for data-driven modelling due to its robust fitting and adaptability. To determine the correlation between the two parameters, (XIN LIU, 2021) has summarized in his article that Matheri et al. developed an ANN model to forecast the concentration of COD and trace metals in WWTPs. The results in the article demonstrates that a simulation model of WWTP can be developed with neural networks.

In order to simulate the performance of an auto-aerated immobilized biomass (AIB)

reactor filled with sponge media, Bakr et al. developed an ANN model that was optimized based on the Levenberg-Marquardt algorithm. An experiment displayed that the model's R2 (coefficient of determination) value was satisfactory fit in training, testing, and verification and that the model can reflect reality. (XIN LIU, 2021), also summarized the effluent COD of the Touggourt WWTP was predicted using ANN model by Bekkari et al. and the findings showed that this modelling approach can be a useful tool for controlling, predicting and simulating the performance of WTTP.

The mentioned research above fails to properly resolve the sequence dependence between input variables and ignores the time series characteristics of wastewater data, which restricts the model's ability to handle time series forecasting tasks. Common issues, such as gradient vanishment and explosion conditions may develop during training a neural network, especially as the number of neural network layers rises. However these issues has been addressed by LSTM neural network by introducing gating. It is an enhanced neural network built on a recurrent neural network (RNN) that is capable of balancing the temporal and nonlinear relationship of wastewater data. The state of the art of LSTM neural network is at present being used successfully in applications for natural language processing, speech recognition and other tasks. Many scholars have attempted to apply the LSTM neural network to the field of wastewater treatment in recognition of its success in these other fields. To simulate the wastewater treatment process, (XIN LIU, 2021) mentioned that Zhiwei et al. created the LSTM model to predict the nutrient removal efficiency of WWTP.

The LSTM neural network is capable of extracting reliable features from data, however it is unable to learn locally important features. Recent studies suggest that time-series tasks prediction can be achieved using LSTM neural networks based on attentional mechanisms. (XIN LIU, 2021) demonstrated that LSTM neural network can gather local information and effectively handle long-term dependencies by incorporating a self-attention mechanism. In (XIN LIU, 2021) summary of LSTM neural network used in recent studies, he also mentions that Zang et al. demonstrates the efficiency of LSTM model in time-series prediction tasks by comparing the LST model based on attention mechanism with variety of neural network models. The experimental findings demonstrates that the accuracy and practicability of the LSTM model can be increased

by including an attention mechanism. Therefore, implementing an attention mechanism to an LSTM neural network can enhance the neural networks ability for obtaining locally important features from wastewater data, thus improving the model's stability and predictability (XIN LIU, 2021).

3 Theoretical background

The Utilization of machine learning teaches and train machines how to handle data more effectively. Sometimes, even after viewing the data, we are unable to evaluate or interpret the information. In that approach, we implement machine learning. The accessibility of a significant number of datasets has increased demand for machine learning. Machine learning is used in multiple industries to retrieve relevant data. Understanding from the data is the objective of machine learning. How to make machines learn on by themselves without being explicitly programmed has been topic of many research studies. Several programmers use various methods to resolve this problem, which involves large data sets. (Mahesh, 2018)

Machine learning uses a variety of methods to address data issues. Data scientist want to emphasize that there is no algorithm that works well for every situation. The kind of algorithm used depend on the type of problem one attempting to resolve, how many variables there are, what kind of model will perform effectively, and other factors (Mahesh, 2018). The following is an overview of some machine learning algorithms implemented and used in this project.

3.1 Supervised learning

A function that maps an input to an output is learned through supervised learning using sample input-output pairs. It utilize labelled data training data composed up of a collection of training examples to infer a function. Algorithms that require external supervision are those that fall under the category of supervised machine learning. train and test datasets are created from the input dataset are constructed from the input dataset. The output variable in the train dataset has to be classified or predicted. All algorithms identify some sort of

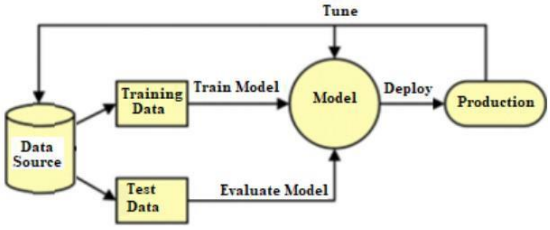

Figure 2, Supervised learning workflow (Mahesh, 2018).

patterns from the training dataset and use them to classify or predict the test dataset. The flowchart below shows the process used by supervised machine learning algorithms (Mahesh, 2018).

3.1.1 Linear Regression

Regression analysis is a statistical method know as linear regression that is frequently used to identify the interdependent quantitative relationship between two or more variables. There is linear relationship between the input eigenvector x and the output value y. $f\ (x)\ =\ h_\theta\ (x)\ =\ \theta_0 +\ \theta_{1x1}\ +\ \theta_{2x2} + \ldots + \theta_{nxn}$. (Rui Wang, 2021)

$f(x)$: $predicted\ values$; $\theta_0, \ldots, \theta_n$: Linear model parameter

The objective of linear regression is to identify a line that minimize the following loss function and fits data points as closely as feasible:

$$J = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 \qquad (1)$$

Where y_i is the actual value , f(xi) is predicted value; the closer this function fits, the smaller its value. Lasso Regression and Ridge Regression both modify the standard linear regression's loss function while maintaining the other components intact. The loss function in Lasso Regression, which adds L1 regularization to the linear regression's loss function, is as follows:

$$J\ = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 +\ \lambda\omega_1 \qquad (2)$$

λ : weight coefficient, 1 norm.

Ridge regression adds L2 regularization to the loss function of linear regression, and its loss function is as follows:

$$J\ = \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2 +\ \lambda\omega_2^2 \qquad (3)$$

2 Norm:

Ridge and Lasso and be used to resolve the overfitting issues with linear regression, because adding L1 and L2 regularization allows the weights to be as minimal as possible for all parameters, then a reduced parameter can be constructed. As a result, the

model's parameter values are smaller than those that are usually sampled, allowing to adapt to various data sets and in certain aspects, avoiding the over-fitting phenomena. In particular, Ridge and Lasso makes it easier to obtain a weight close to 0. (Rui Wang, 2021)

3.1.2 Ridge

The most commonly utilized form of regularized regression is called Ridge Regression, which limits the sum of squares of the weights of the coefficients though a constraint on the p coefficients (Mayooran Thevaraja, 2019). Ridge regression can be formulated as follow:

$$\beta_{ridge} = arg \min_{\beta \in R_p} \sum_{i=1}^{n} (Yi - XiT\beta)^2 \quad (4)$$

$$with\ sibject\ to \sum \beta_j^2 \leq t\ for\ t \geq 0$$

Therefore, the feasible set for this minimization problem is limited to $S(t) = \{\beta \in Rp : \|\beta\| \leq t\}$ where β does not include the intercept β_0. Due to the L2-penalty, the ridge estimator is constant when the xj are scaled (Mayooran Thevaraja, 2019). Centering the predictors solves this problem and can be formulated using Lagrange multiplier as:

$$\beta_{ridge} = arg \min_{\beta \in R_p} \{\sum_{i=1}^{n} (Yi - X_i^T\beta)^2 + \sum_{j=1}^{p} \beta_j^2\} \quad (5)$$

Convex Minimization can be to solve the optimization problem in equation (5). Because X is assumed to be full-rank, it follows that the Residual Sum of squares, or RSS is convex in if XTX is positive definite. The sum squares for j can also be demonstrated to be convexity quite simply (Mayooran Thevaraja, 2019). One way to formulate the RSS for ridge regression is as follows:

$$\mathrm{RSS}(\beta;\ \lambda) = (\mathrm{y} - \mathrm{X}\beta)^{\mathrm{T}}(\mathrm{y} - \mathrm{X}\beta) + \lambda\beta^{\mathrm{T}}\beta \quad (6)$$

As conducted for the conventional OLS criterion for multiple linear regression, one can minimize this criterion by applying straightforward matrix calculus approaches (Mayooran Thevaraja, 2019). To put it another way, by setting the first derivative to zero, we can obtain:

$$\partial\partial\beta\ RSS(\beta;\ \lambda) = 2(X^TX)\beta - 2X^Ty + 2\lambda\beta = 0 \quad (6)$$

The following sentence would further simplify this expression:

$$(X^T X + \lambda I)\beta = X^T y \qquad (8)$$

The ridge estimator are, therefore:

$$\hat{\beta}_{ridge} = (X^T X + \lambda I)^{-1} X^T y \qquad (9)$$

3.1.3 Lasso

By setting a number of the slope parameters to zero, the lasso regression aims to provide a sparse solution. Additionally, the lasso is constructed in relation to the central matrix, X. the intercept, 0 is also not included in the penalty term because the L1-penalty is only applied to the slope coefficients (Mayooran Thevaraja, 2019). As a constrained minimization problem, the Lasso can be represented as:

$$\beta_{lasso} = arg \min_{\beta \in R^p} \sum_{i=1}^{n} (Yi - X_i^T \beta)^2 \qquad (10)$$

Subject to: $\sum_{i=1}^{p} |\beta_j| \leq t$

For $t \geq 0$ using the Lagrangian for the penalty, which is formulated as follows:

$$\beta_{ridge} = arg \min_{\beta \in R_p} \{ \sum_{i=1}^{n} (Yi - X_i^T \beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \} \qquad (11)$$

Where $\lambda \geq 0$ and, as before, here exist a one to one correspondence between t and λ. The lasso does not allow a closed-form solution, in contrast to ridge regression. The L1-penalty renders the y_i solution non-linear. An effective approximation can be made to the solution of the quadratic programming problem described by the restricted minimization mentioned above (Mayooran Thevaraja, 2019).

3.1.4 Decision Tree

A decision Tree is a method for decision analysis that be used to determine the probability that the expected value of net present value, also referred as NPV is greater than or equal to zero, determine whether a project is feasible by considering the probability that various scenarios will occur and evaluate the project risk. It is graphical method for using probability analysis intuitively. A decision tree is prediction model used in machine learning that illustrates the mapping between object values and object attributes. (Rui Wang, 2021)

A decision tree is a tree structure in which each leaf node represent the Y values to be predicted, each inner node represent a judgment on an attribute, and in each branch reflects the judgment's result. The following two processes , which are carried out be learning the training data, constitute the majority of the development of a decision tree. Nodes are generally split into two child notes (or N child nodes, if the structure being represented is not a binary tree) depending on how difficult it is to determine the attribute represented by each node. (Nashia Deepnarain, 2019)

Determining the threshold value: select the appropriate one to reduce the prediction error rate. An index called information entropy is used to evaluate information uncertainty. The following is information entropy formula:

$$H(x) = -\sum_{x \in X} P(x) \log_2 P(x) \qquad (12)$$

$P(x)$: the probability of x

This formula is to divide information entropy change value before and after the dataset, and the feature with the largest information entropy change range is selected as the base for the data set partition. To select the best data row molecular dataset, the feature with the largest information gain is selected as the splitting node, the sub-dataset is processed recursively after partitioning, and after following processes are repeated for the features that have been picked. Two things must happen in order for recursion to stop: first, all of the features must be utilized, and second, the information entropy gain after division must be sufficiently small. (Rui Wang, 2021)

3.1.5 Gradient Boosting Decision Tree

A common used algorithm for Regression and Classification is Gradient Boosting Decision Tree (GBDT). The Regression and Classification Trees provide as the decision tree utilized by GBDT. Identifying the ideal partition point is the crucial element of the tree algorithm. Furthermore, all of the desirables values for each feature are contained at the partition point in the regression tree. The entropy or Gini coefficient serves as the optimum partition point criterion in the classification tree, whereas the sample label in the regression tree continuous number. As a result, it no longer appropriate to utilize

metrics like entropy and should instead use the square error, a reliable measure fitness. (Rui Wang, 2021).

The steps of GBDT are presented as follows:

1. The initial constant values of the model β is given :

$$F_0(x) = argmin_{\beta} \sum_{i=1}^{n} L(y_i, \beta) \qquad (13)$$

2. The gradient direction of residuals is determined for m = 1 : M, where M is the number of iterations.

$$y_i^* = - \left[\frac{\partial L(y_i, F(x_i))}{(\partial F(x_i))}\right]_{F(x_i)-F\_m(x)} , I = \{1, 2, \ldots\ldots, N\} \qquad (14)$$

3. To fit simple data and create the initial model, the fundamental classifiers are utilized. The model's parameter a_m is calculated using the last square approach and the model $h(x_i, a_m)$ is fitted.

$$a_m = argmin_{\alpha,\beta} \sum_{i=1}^{n} [y_i^* - \beta h(x_i, a)]^2 \qquad (15)$$

4. The loss function is reduced. A new step for the model, or the weight of the existing model, is determined by Eq. (8)

$$a_m = argmin_{\alpha,\beta} \sum_{i=1}^{n} L(y_i, F_{m-1}(x) + \beta h(x_i, a)) \qquad (16)$$

5. Following is an update to the model:

$$F_m(x) = F_{m-1}(x) + \beta_m h(x_i, a) \qquad (17)$$

However, when raw data is fed into GBDT to be analyzed, information gain of feature branch points must be calculated several times due to the dimension and size of the data sample. It causes the number of iterations to rise while slowing convergence and update rates. According to (Haidi Rao, 2019), it is suggested that employing Adaptive Boosting with classification Trees to optimize the initial data is fed into GBDT. The proposed approach ensures the correctness and efficiency of GBDT while forcibly reducing the initial feature dimensions of sample data and quickly creating a decision tree to determine the weight of features. (Haidi Rao, 2019)

1) **XGBOOST: eXtreme Gradient Boosting:** In this study another form of GBDT is used and therefore introduced under this chapter. Xgboost is short for eXtreme Gradient boosting package. Is a scalable and effective use of the gradient boosting framework (Tianqi Chen, 2017). Linear model solver and a tree learning algorithm are included in the package. It provides a number of objective operations, such as classification, regression and ranking. The package is developed to be extensible, so that the users are also allowed to define their own objectives. (Hui Chen, 2020).

3.1.6 LSTM

Long Short-Term memory (LSTM) is a type of Recurrent Neural network (RNN) with ability to store values from earlier stages for use in the future (Sima Siami-Namini, 2018). It is vital to get a general idea of how a neural network works before diving into LSTM.

1) **Artificial Intelligence (ANN):** A neural network consist of 3 three layers, an input layer, a hidden layer and an output layer. The dimensionality, or number of nodes in the input layer, depends on how many features are included in the data set. The nodes created in the hidden layer(s) are connected to these nodes via structures called “synapses”. For each node in the input layer, the synapses carries weights. The weights simply act as a decision maker to determine which input or signal should be allowed pass through and which shouldn’t. The weights also demonstrate the strength or extensive the hidden layer is. In basic terms, neural network learns by adjusting the weight for each synopsis. The nodes in the hidden layers uses an activation function, such as tangent hyperbolic or sigmoid to convert the weighted sum of the inputs to the outputs, or predicted values. The tanh and sigmoid functions both shape like an “S”, where the output of sigmoid range is (0,1) and the outputs of tanh vary between (-1,1) (Hui Chen, 2020). Their mathematical functions are defined below.

Sigmoid function:

$$S(x) = \frac{1}{1+e-x} \quad \textbf{(18)}$$

Tanh function:

$$T(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (19)$$

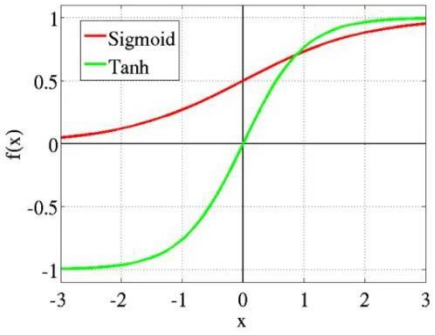

Figure 3, , graphical display of sigmoid and tanh function

The output layer generates a vector of probabilities for several outputs and chooses the one with lowest error rate or cost, that is by minimizing the difference between expected and predicted values, also known as the cost, using a function called SoftMax.values and subsequently in the cost. Model training occurs when the cost function is minimized. (Sima Siami-Namini, 2018)

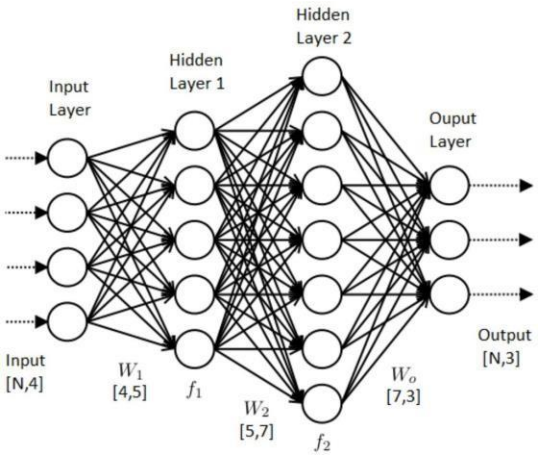

Figure 4, Graphical display of Artificial Neural Network

2) Recurrent Neural Network (RNN):

A recurrent neural network is particular type of neural network where the objective is to predict the subsequent observations in an ongoing sequence of observations in relation to the previous observation. RNNs have been developed to use sequential observations and learn from prior phases in order to predict future trends. As a result, when

predicting on the subsequent stages, the earlier stages data must be kept in mind. The information gathered in earlier phases of reading sequential data is stored in RNN's hidden layers, which act as internal storage. The reason RNNs are known as "recurrent" is because they carry out the same tasks for every element of the sequence and have the ability to forecast/predict future sequential data that has yet to be observed by the user. The primary issue with a typical Recurrent Neural Network is that these networks are not ideal for remembering longer data sequences, since the only remember a few earlier steps in the sequence. The "memory line" developed in the Long Short-Term Memory (LSTM) recurrent network is used to solve this difficult challenge (Sima Siami-Namini, 2018).

3) Long Short-Term Memory (LSTM) :

LSTM are a special kind of RNNs that have additional features memorizing the sequence of data. Data streams are gathered and stored in each LSTM's set of cell, or system modules. The upper line of each cell resembles a transport line that connects from one module to another, carrying data from the past and gathering it for the present module. Data in each cell can be removed, filtered, or added for the subsequent cell due to the work of some gates in each cell. As a result, the cells can choose whether to allow data to travel though or dispose of it using the gates, which are based on sigmoidal neural network layer. Numbers ranging from 0 to 1 are generated by each sigmoid layer, indicating how much of each segment of data should pass through each cell. More specifically, a value estimation of zero indicates "let nothing pass though", whereas a value estimation of value of one indicates "let everything pass though". Each LSTM utilize three different types of gates to regulate the state of each cell (Sima Siami-Namini, 2018). Following illustrates the three different types of gates.

- The Forget Gate generates a value between 0 and 1, with 1 indicating "completely keep this" and 0 indicating "completely forget this"
- Memory Gate makes decision to which new data should be stored in the cell. First, the "input door layer" of a sigmoid layer selects the value to be adjusted. A tanh layer creates a vector of potential a new values to be added to the state next.
- Each cell's yield is determined by the Output Gate. The cell state, together with

filtered and newly added data, will all be variables in the yielded value.

3.2 Cubic Spline Interpolation

The basic concept of cubic spline interpolation is based on the tool used by engineers to construct curved paths through a collection of points. Weights are attached to flat surface at the connection points of this spline. Then a flexible strip is bent across each of these weights to create an pleasingly smooth curve. In theory, the mathematical spline is similar. In this particular case, the points are numerical data. The coefficients of the cubic polynomials used to interpolate the data are weights. These coefficients "bend" the line, allowing it to pass through each data point without indicating any unpredictable behavior or discontinuities (Sky McKinley, 1998).

The important concept is to fit a piecewise function of the form:

$$S(x) = \begin{cases} s_1(x), & x_1 \le x \le x_2 \\ s_2(x), & x_2 \le x \le x_3 \\ s_{n-1}(x), & x_{n-1} \le x \le x_n \end{cases} \quad (20)$$

Where s_i is a third degree polynomial defined by:

$$s_i(x) = a_i(x - x_i)^3 + b_i(x - x_i)^2 + c_i(x - x_i) + d_i \quad (21)$$

For $i = 1, 2, \ldots, n-1$

The method relies primarily on the first and second derivatives of these n-1 equations which are:

$$s_i'(x) = 3a_i(x - x_i) + 2b_i(x - x_i) + c_i \quad (22)$$

$$s_i''(x) = 6a_i(x - x_i) + 2b_i \quad (23)$$

For $i = 1, 2, \ldots, n-1$. (Sky McKinley, 1998)

3.3 Pearson correlation coefficient

Assume that the data consist of nxm matrix, where n represent the number of instances and m represent number of attributes associated with each instance. Let X and Y represent instances with m characteristics (Ekasit Kijsipongse, 2011). The Pearson correlation coefficient, represented as $r_{x,y}$ between two occurrences X and Y described

mathematically as follows (Ekasit Kijsipongse, 2011):

$$r_{x,y} = \frac{\sum_{i=1}^{m} (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{m} (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^{m} (Y_i - \bar{Y})^2}} \qquad (24)$$

Where $\bar{X}$ and $\bar{Y}$ are defined as:

$$\bar{X} = \frac{1}{m} \sum_{i=1}^{m} X_i \qquad (25)$$

$$\bar{Y} = \frac{1}{m} \sum_{i=1}^{m} Y_i \qquad (26)$$

The linear relationship between two instances is quantified by the Pearson correlation coefficient. $r_{x,y}$ value falls between -1 and 1. In the case of two uncorrelated instances, it is close to zero. X and Y have a relationship when the value is positive. The correlation is stronger the higher the value. X and Y are negatively correlated if $r_{x,y}$ has a negative value. The correlation matrix, where each element represents the Pearson correlation coefficients, $r_{X,Y}$ of the various instances pairs (X, Y), can be used to express the correlation between all pairs of instances. Due to the ability to individually calculate each $r_{X,Y}$, the correlation matrix is very parallelizable.

3.4 Evaluation Metrics

The Mean Absolute Error (MAE) and Root-Mean-Squared error (RMSE) are two standards metrics used in model evaluation (Hodson, 2022). The MAE and RMSE are

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \qquad (27)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y}_i)^2} \qquad (28)$$

The mean squared error (MSE), as its name indicates, is the square root of the RMSE. The relative rankings of the models are unaffected by taking the root, but it yields a metric with the same units as y serve as a simple representation of the usual or "standard" error for normally distributed error. (Hodson, 2022). The Root-Mean-Squared error always lies between 0 and 1, where a higher R-squared score indicates a better model fit. It is difficult to provide rules regarding what R-squared should be, as it varies from study to study. R-squared values of 0.75, 0.50 or 0.25 can be interpreted as rule, where they indicates substantial, moderate or weak. (Sarstedt, M., & Mooi, E. (2014,p.211))

4 METHODOLOGY AND MATERIALS

In this chapter the focus will be on machine learning domain. Where we will aim to develop smart models based on data-driven algorithms that can accurately generate predictions without the explicit necessity to program them for that objective. It can be seen as training a function that maps input variables to output variables. Once the function are defined it can be used to generalize the learned behavior and make predictions given a new unseen instance of input variables. The data-driven approach depends on historical and existing data sets to infer the unknown function based on parametric or non-parametric algorithms. (Maddi Etxegarai, 2022).

In this thesis we propose the use of supervised learning, more specifically regression algorithms for the virtual sensor implementation, as illustrated in Figure 2. The proposed algorithm relies heavily on labelled datasets providing both input and output variables to infer the function. Furthermore, in regression problems, the output variables are continuous values instead of the categorical data type required in classification problems. (Maddi Etxegarai, 2022)

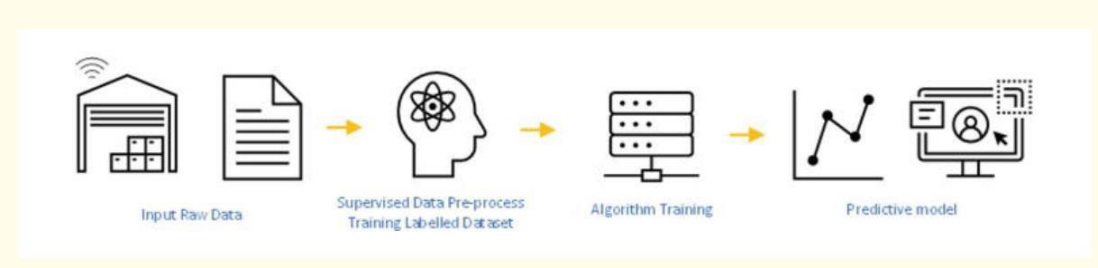

Figure 5, Supervised learning paradigm.

Cross Industry Standard Process for Data Mining (CRISP-DM) methodology provides flexible framework, and it is organized into seven phases as illustrated in figure 3. To successfully conduct data-driven projects, it necessary to follow the mentioned standardized method to translate business problems into tasks, provide means and suggest data transformation for evaluating the process and the final results. As illustrated in figure 3 the data mining process is a cycle. Since it is a necessity to go back and forth between stages until a valid solution is found and meets the quality criteria. (Maddi Etxegarai, 2022)

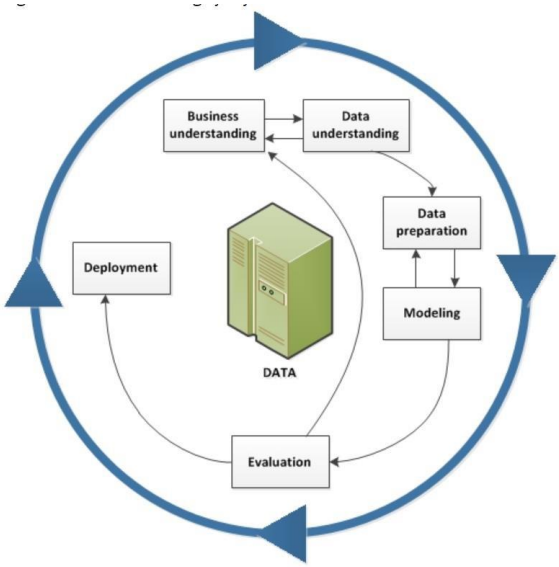

Figure 6, Phases of the CRISP-DM process

The Process of CRISP-DM starts with understanding the business requirements, perspective and objectives to create a method or a project plan together with the field expert. Once the objectives have been established, the initial data are gathered and analyzed. This initial analysis can help to detect interesting subsets and identify quality problems to enable hidden information. The final datasets will be constructed during data preparation phase and used to feed and validate the algorithms. This process typically requires a substantial amount of effort since it is the most-time consuming and complex phase that produces the training dataset, that generates one of the most critical outcomes. This is crucial since data in the field of data science provides as the foundation for all solutions, thus it must be dependable and consistent. Data cleaning, data scaling, feature selection or feature engineering are some of the common processes carried out in this stage and require experience for a successful implementation. (Maddi Etxegarai, 2022)

4.1 Data analytics for WWTP

The following is an overview of data analytics for wastewater treatment plant.

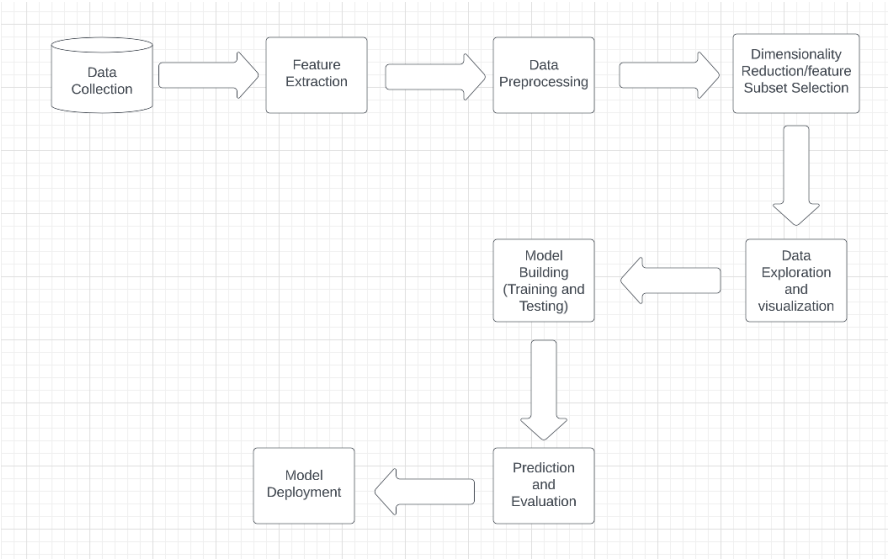

Figure 7, Flowchart of data analytics in WWTP using machine learning

The data used in this study is a timeseries dataset and was sent from the HIAS wastewater treatment plant. The data contains 8862 wastewater treatment historical data points from 1.december 2022 to 31.January 2023. The data set consist of a decent amount of data with a file size of 1.66 MB. The data contains a total of 20 columns which is illustrated in table 1. The Dataset used in this study is time series dataset, which requires a thoroughly analysis to prepare the data for the incoming task. The measurements in the dataset is measured frequently and has a frequency of 10 minutes.

The preparation, processing and model indexes in this study is performed using Excel and Python 3.10 Jupyter Notebook, windows 11. Jupyter Notebook is an open-source software containing equations, visualization and codes. Data cleaning, data visualization, statistical modeling, machine learning and other applications are included in the program by calling forth libraries containing and supporting these applications. The data set collection was analyzed, selected and prepared into a suitable dataset utilizing the program in order to develop a prediction model.

4.2 Data preparation

Reviewing the currently available data is essential, as is determining how it relates to the work and determining whether it is feasible to source new data that has been acquired especially for the intended task. Whether there is enough data to realistically produce the necessary machine learning results, also be evaluated. Data sets are frequently of

low quality, hence data quality should also be investigated into. Manual data collectors could not be very effective and it may arise human errors when assuring data accuracy. For instances, default values offered by a system have tendency to be substantially overrepresented in the data obtained. There is also a chance that automated data collection procedure will produce incomplete data or inaccurate. The precision of a measuring instrument may be lower than desirable. (Zahraa Said Abdallah, 2017)

The process of preparation usually involves many changes and conversions and needs to be repeated numerous times. Despite improvements in data processing tools, each of those conversions or changes still require a lot of manual work and usually consumes a significant amount of effort and time. Working with large data still remains a challenge. It is generally acknowledged that data preparations is the most time consuming aspect of data analysis. (Zahraa Said Abdallah, 2017)

To prepare the data for the next task the integration of multiple datasets is required in the datamining process. The integration of data from multiple datasets is known as data integration. Redundancies and inconsistencies in dataset can be reduced and avoided with thorough integration. The efficiency and accuracy of mining process can be improved by data integration (Wongburi, 2021). The difficulty with data integration is finding ways to align objects and structures from different datasets. In this study, the integration of data was done in Excel. There were a total of 3 data sets, where two of them are online data variables from the wastewater treatment plant HIAS. The two dataset are measurements from December 2022 and January 23, where the frequency is 10 minutes. December and January where collected into one dataset all together with the 3rd dataset which contains data on corrected sCOD values. Since the dataset with corrected sCOD values is measured every 5 minutes, it required manually work to get the same frequency (10 minutes) for all measurements.

4.3 Data Preprocessing

The collected treatment data often contain accidental errors and system errors. These errors may be caused by the sensor malfunctions. When predicting the output using non-preprocessed data it could greatly reduce the prediction accuracy. Therefore is preprocessing of data critical step to avoid such issues. Data preprocessing is how the data are encoded or transformed to a state that a computer can easily comprehend.

Preprocessing of data help the computer to understand the data (Wongburi, 2021). The following is a summary of data preprocessing steps used in this study.

1. Data cleaning:

Cleaning data includes several task, such as smoothing noisy data, removing or identifying outliers, correcting inconsistencies and filling in missing values. Data which is unclean can cause unpredictability for the mining process, resulting in inaccurate output. Thus, one of the most crucial methods of data preprocessing is the data cleaning process. To identify inconsistencies, noisy data and missing values several approaches were used in Jupyter notebook using python 3.10.

In figure 8 and 9 it is demonstrates missing values for each parameter in the dataset. The data set used in this study is collected into one dataset as mentioned. December is a month that are heavily effected by holidays, which is taken into consideration, since it can create unusual patterns in the data. For example, December might have increased organic load, since there is often an increase in food consumption, which may result in higher organic load In the wastewater. This could cause spikes in parameters, such as chemical oxygen demand (COD). There could also be increase in the concentration of certain parameters in the wastewater, due to heavily usage of household chemicals. This may result in higher concentrations of parameters, such as phosphate, ammonia and nitrate.

The missing values where identified using pandas, to be more specific the isnull.().sum() function. The mentioned function is used mainly for detection of missing values. The function where used to identify missing values for the time-series dataset, it is also noteworthy to mention that under the identification of missing values, was done seasonally, as well it was done for the whole dataset to get a better understanding of the data and the missing values. As illustrated in figure 8 and 9 you can see there are a larger amount of missing values in December. This could be because of the holiday season, were some industries may reduce their operations or even shutdown the operations during this period.

```
In [7]: december_df.isnull().sum()

Out[7]: Time                      0
        O2_sone_4                99
        O2_sone_5               100
        O2_sone_6               103
        O2_sone_8               102
        O2_sone_9               101
        Mengde_luft_sone_4      101
        Mengde_luft_sone_5      100
        Mengde_luft_sone_6      103
        Mengde_luft_sone_7      100
        Mengde_luft_sone_8      104
        Mengde_luft_sone_9      100
        Mengde_luft_sone_10     101
        Vannmengde_inn_linje_1  104
        NO2_sone_7_linje_1       99
        NO3_inn_bio             220
        SS_ut_bio               112
        SS_ut_diskfilter        112
        PO4_ut_diskfilter       628
        sCOD_korrigert            1
        dtype: int64
```

Figure 9, Amount of missing values in dataset (December)

```
In [8]: january_df.isnull().sum()

Out[8]: Time                      0
        O2_sone_4                 4
        O2_sone_5                 4
        O2_sone_6                 4
        O2_sone_8                 4
        O2_sone_9                 4
        Mengde_luft_sone_4        4
        Mengde_luft_sone_5        4
        Mengde_luft_sone_6        4
        Mengde_luft_sone_7        4
        Mengde_luft_sone_8        3
        Mengde_luft_sone_9        4
        Mengde_luft_sone_10       5
        Vannmengde_inn_linje_1    4
        NO2_sone_7_linje_1        4
        NO3_inn_bio            1060
        SS_ut_bio                 7
        SS_ut_diskfilter          3
        PO4_ut_diskfilter       254
        sCOD_korrigert           18
        dtype: int64
```

Figure 10, Amount of missing values in dataset (January)

```
Out[6]: Time                      0
        O2_sone_4               103
        O2_sone_5               104
        O2_sone_6               107
        O2_sone_8               106
        O2_sone_9               105
        Mengde_luft_sone_4      105
        Mengde_luft_sone_5      104
        Mengde_luft_sone_6      107
        Mengde_luft_sone_7      104
        Mengde_luft_sone_8      107
        Mengde_luft_sone_9      104
        Mengde_luft_sone_10     106
        Vannmengde_inn_linje_1  108
        NO2_sone_7_linje_1      103
        NO3_inn_bio            1280
        SS_ut_bio               119
        SS_ut_diskfilter        115
        PO4_ut_diskfilter       882
        sCOD_korrigert           19
        dtype: int64
```

Figure 8, Amount of missing values in the dataset (both December and January)

To fill in the missing values several interpolation methods were approached to try to create one or more functions to fit the discrete datapoints. Interpolations functions such as Linear interpolation, Lagrange interpolation and newton interpolation are used to draw the corresponding curves or lines, and also can obtain derivatives of different data points by executing derivative functions. By using cubic spline interpolation one can fit the data points so as to obtain the derivatives of the points on a smoother curve with less error between the actual function and the fit function, therefore in this study cubic spline interpolation is selected to calculate and estimate the missing values. Cubic spline interpolation can transform the data points of time series into the ones on a smooth curve. (Hailin Li, 2014)

2. Handling outliers

To successfully process data it is necessary to identify potential outliers in the dataset. An outlier is another type of data irregularity or inconsistency that requires attention in the cleaning process. Data that deviates from the normal distribution of data are considered outliers. (Zahraa Said Abdallah, 2017). When analyzing the data outliers may appear and can be seen from two different perspectives. Either they might be

discovered some interesting elements that could potentially represent significant elements in the data or they might only be some glitches that may appear when cleaning the data. To get a better understanding of reasoning behind the occurrence of outliers and to classify them, one must define what the usual behavior of the data is and how significant or different the outliers are relative to the usual behavior of the data being processed.

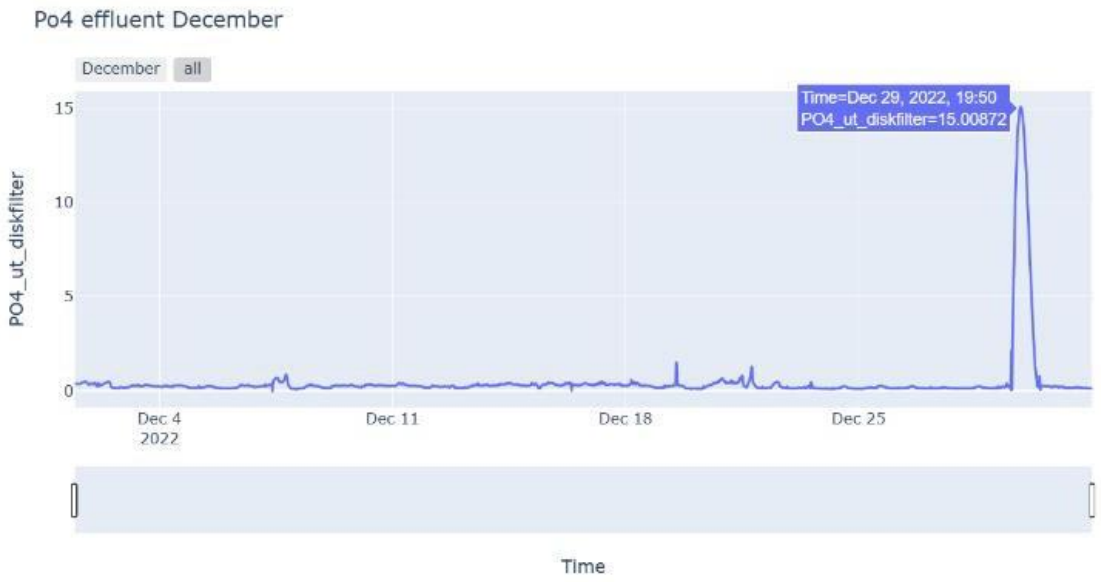

Figure 12, Visualization of PO4 in effluent (December)

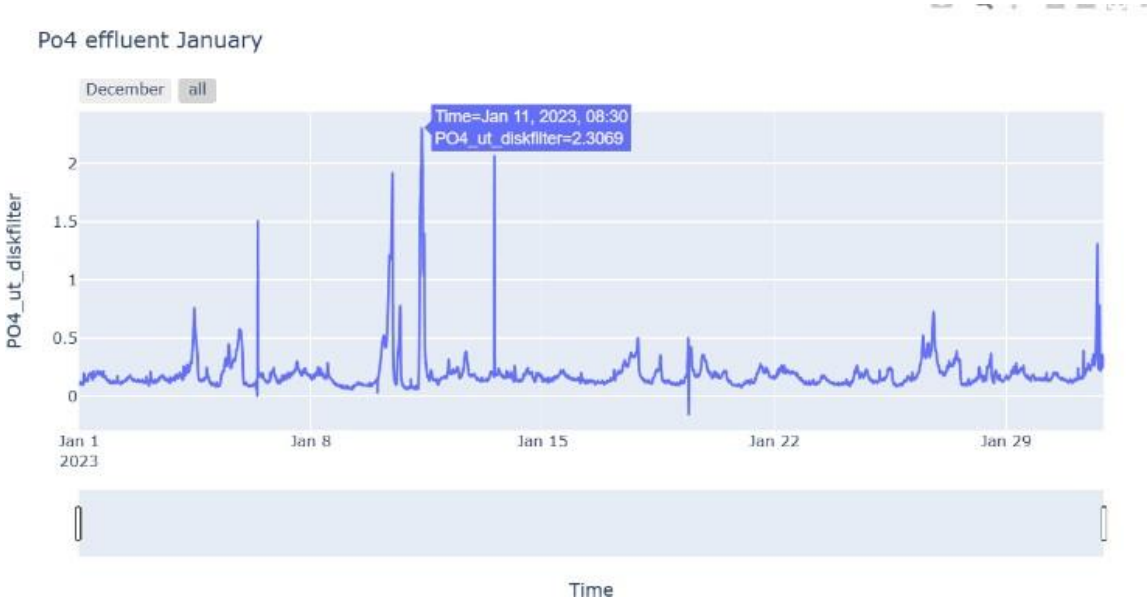

Figure 11, Visualization of sCOD in effluent (December)

The process of handling outliers is done using Python 3.10 Jupyter Notebook. Figure 11 and 12 illustrates irregularity and inconsistency when visualizing the data set. These are outliers and might be some glitches that may have appeared when interpolating the missing values. This is rather the case, since the usual behavior of the data set and how significant the outliers are do not relate to each other. After identifying the outliers one can remove them by using pandas to remove the outliers identified. This is done by calculating the absolute deviation of each value in the Data Frame from its mean. Then we check if the absolute deviation of each value is greater than three times the standard deviation of the Data Frame. Thereafter, it is created a mask where values that meet the conditions we set are set to NaN, also known as missing values. To fill the missing values in the Data Frame forward fill method is used. The forward fill method (ffill) fills the missing values with the last non-NaN value in the same column. For now only values forward in the Data Frame are handled, by handling the other missing values backwards in the Data Frame the backward fill method is used to fill the missing values with the next non-NaN value in the same column. This approach to remove outliers is a useful method for dealing with excessive values that can be affecting the modeling or analysis.

3. Data mining

The preprocessed data were used to extract information using a variety of statistical methods, such as seaborn pairplot and correlation matrix. Seaborn pairplot is a tool for visualizing the pairwise relationship between multiple variables in a dataset. Correlation matrix is a table that demonstrates the correlation coefficients between multiple variables in a dataset. It provides direction and shows the importance of those correlations, And helping in directing further analysis and modelling

4.4 Time lag calculation

Due to dynamic characteristics of the wastewater treatment processes' multiple flows of both water and sludge, there are lags in the time at which the water in the process streams reaches different meters (Dong Wang S. T., 2021). The original dataset, though, are time series. In order to interpret the machine learning models in terms of WWTP processes, the original time-series data must be shifted. To handle this lag in a WWTP involves shifting the values of the parameters forward in time by a certain number of time steps. This is done to take into consideration the possibility that changes in some factors, like influent flowrate or pollutant concentrations, may take some time to filter through the treatment flow process and causes changes in the effluent quality.

By using Python 3.10 in Jupyter notebook, it possible to handle such task by first defining the number of time steps to lag the features. In this study, after several discussions with supervisor, it was concluded to use a range of 5 to 10 hours, with each time step being 6 minutes. Then the shift method in Pandas is used to shift the values of the
parameters/features forward in time by the specified number of time steps.

4.5 Models development

The development of the proposed models is established using regression models, time-series forecasting models and data-driven multi-classification models. The proposed approach in this study comprises of 5 different models. Linear regression, Lasso regression, Ridge regression, decision tree, GBDT, xgboost and
LSTM are the models used to predict PO4 in the effluent. After making predictions for all 5 models, the evaluation of their performance will be executed using evaluation metrics proposed in this study, such as mean absolute error, mean squared error and r2. Visualizations of the performance will also be implemented utilizing scatter plots to gain insights into the performance off the models.

In this study, the training rate and network structure in wastewater treatment of HIAS was analyzed using the parameters presented in table 1. The WWTP online data variables presented in table 1 were used as input data in the models, as illustrated in Figure 13.

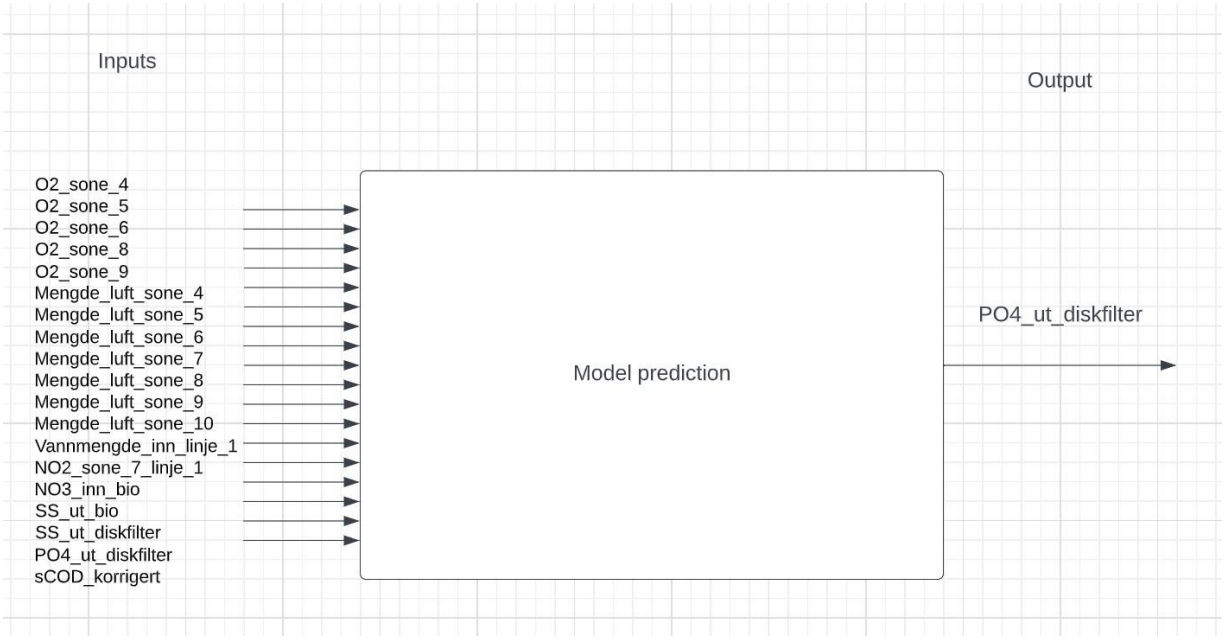

Figure 13, Block diagram of input and output for virtual measurement of PO4 in the effluent

Table 1, Parameters from dataset of online measurements.

Name	Name Tag in excel	Unit of measurement
O2 zone 4	O2_sone_4	mg/l
O2 zone 5	O2_sone_5	mg/l
O2 zone 6	O2_sone_6	mg/l
O2 zone 8	O2_sone_8	mg/l
O2 zone 9	O2_sone_9	mg/l
Aeration rate zone 4	Mengde_luft_sone_4	Nm/h
Aeration rate zone 5	Mengde_luft_sone_5	Nm/h
Aeration rate zone 6	Mengde_luft_sone_6	Nm3/h
Aeration rate zone 7	Mengde_luft_sone_7	Nm3/h
Aeration rate zone 8	Mengde_luft_sone_8	Nm3/h
Aeration rate zone 9	Mengde_luft_sone_9	Nm3/h

Aeration rate zone 10	Mengde_luft_sone_10	Nm3/h
Wastewater flow in	Vannmengde_inn_linje_1	l/s
NO2 zone 7	NO2_sone_7_linje_1	mg/l

NO3 inlet	NO3_inn_bio	mg/l
SCOD inlet	sCOD_korrigert	mg/l
Suspended solids out	SS_ut_bio	mg/l
Suspended solids after disc filter	SS_ut_diskfilter	mg/l
PO4 in the effluent	PO4_ut_diskfilter	mg/l

4.5.1 Regression

In this section we'll go through the development of the predictive models for the PO4 effluent using linear regression, Lasso regression and Ridge regression. Since the target value is PO4 in the effluent, the data of PO4 was initially preprocessed to remove it from the dataset. The PO4 effluent variables was assigned to the y variable, whereas the

remaining variables were assigned to the X variables. Y variable, also known as dependent variable (output) can be seen as the state, target pr final goal we study and try to predict. X variable or the independent variable (input) can be seen as the cause of those states. Thereafter the data was split into training and testing sets using a 80/20 split. The random state parameter was set to 42 for reproducibility and consistency.

In this analysis, the three regression models as mentioned above were developed. The development of these models utilized the features of sklearn library. The following subsection describe the model development process for each regression model.

1) Linear regression:

Linear regression is a commonly used method for predicting continuous variables. It works by finding a linear relationship between the input features and the output variable, allowing the algorithm to make predictions for new data points. The model was

developed using sklearn's LinearRegression function. The model was then trained using fit method with the training set of data. During the training, the model learned the weights for each feature that best predicted the output variable. Once the model was trained, the predict method was used to make predictions on the testing data. Evaluation metrics, such as means squared error and R-squared were used to evaluate the performance of the linear regression model. The R2 evaluate how well the model fits

the data in comparison to a simple average of the target variable, while MSE evaluate the average squared difference between the predicted variable and actual values. These metrics provide a measure of how well the model is able to generalize to new data.

2) Lasso regression:

Lasso regression is a type of linear regression that performs both feature selection and regularization by reducing the coefficients of the less important features to zero. When dealing with high-dimensional data, it is practically useful, since it can help prevent overfitting and improve the models generalization performance. In this study, Lasso regression was used to predict PO4 in the effluent. To develop the model, sklearn library was also used here. The fit method, which identifies the coefficient that reduce the sum of squared error between predicted value and actual values, was used to train the model on the training data. Once the model has been trained, it was used to predict the values of PO4 variable for the testing data using the predict method. The performance of the model was also evaluated by using the same metrics, such as MSE and R2. These metrics indicates how well the model correlates with the data and can be used to compare how well various models perform.

3) Ridge regression:

Ridge regression is also a type of linear regression that use L2 regularization, which is mentioned in chapter 3. It uses L2 regularization to prevent overfitting. It works by reducing the variable coefficients closer to zero, which improves the variance of the model. In contrast to Lasso regression, Ridge regression decreases each coefficients in the direction of zero rather than setting any to zero. Ridge regression model was developed to predict the PO4 in the effluent. The model was also developed by using sklearn's Ridge function. The model was fit to the training data using the fit method. Once the model fitted, the predict method was applied in order to generate prediction for the testing data. The evaluation for this method is similar to the previous ones.

4.5.2 Data-driven multi-classification

Data-driven multi-classification is widely used method in data science, where the aim is to predict the class label of a sample based on its features. In many real-world scenarios, in this case prediction of the effluent of PO4 in HIAS wastewater treatment plant, it is needed to predict multiple class labels for a given input, which is known as multiclass-

classification. In this study it is investigated various methods to develop data-driven multiclass classification models for the prediction of PO4 in the efluent, such as decision tree, gradient boosting and XGBoost. These algorithms have illustrated in literature promising results in predicting class labels accurately. To decide which algorithm provides the best results for the data set given, evaluation of each methods performances will be compared with the methods used in this study.

To develop the data-driven multi-class classification models, we will be using the same dataset as in the linear regression. The train_split function from sklearn will also be utilized for splitting the dataset into training and testing sets. Once the dataset is split, the three different algorithms presented below will be used to train our models on the training set. The trained models will then be used to make predictions on the testing set and evaluated using the same evaluation metrics used for regression. The objective is to determine the algorithm that performs the best in terms of efficiency and accuracy for predicting the labels of PO4 in the time series dataset.

1) Decision Tree:

For the decision tree model, the DecisionTreeRegressor function from sklearn library will be used. The decision tree has internal nodes that represents a decision on a feature, each branch indicates the decision's outcome, and each leaf node represents a prediction for the target variable, which is also described in chapter 3. Decision trees are a sort of algorithm that generates a tree like model of decisions. For the development of this model the DecisionTreeRegressor function will be used to fit the model to the training set. During this process, the algorithm will search for the optimal set of decision rules that will result in the most accurate predictions of the target variable. Once the model is fit to the training data, it will be used to make predictions on the testing data using the predict method. This allow us to evaluate the performance of the model and determine its accuracy in predicting the target variable for new data points. To evaluate the effectiveness of the decision tree model, evaluation metrics proposed in this study will be used to compare the predicted class labels to the actual class labels in the testing data.

2) Gradient boosting decision tree (GBDT):

For the GBDT model, the GradientBoostRegressor function from sklearn library will be utilized. A prediction is made using the ensemble learning method of gradient boosting by combining several decision trees. It functions by developing each tree one at a time, with each new tree attempting to correct the errors produced by the previous trees. According to literature this creates an accurate prediction model that can handle complex datasets. To develop the gradient boosting model for our data-driven multi-class classification problem, a similar methodology for decision tree model will be used. Using the sklearn library and the train_test_split function, we have to split the dataset into training and testing sets. The model will then be fitted to the training set of data utilizing GradientBoostingRegressor function. This includes developing multiple decision trees repeatedly and minimizing the loss function, which evaluates the difference between the predicted values and actual values of the target variable, in this case PO4 in the effluent. Using the same method for evaluation, the predict method will be utilized to make the predictions on the testing data once it has fitted to the training data. This allows evaluation of the model's performance and evaluation of its accuracy in predicting the target variable for new data points.

3) XGBoost:

For the XGBoost model, similar approach as the previous model will be utilized as well. Were XGBRegressor function will be used from the XGBoost library, which is a gradient boosting framework. In contrast to conventional gradient boosting, XGBoost minimizes overfitting with a more regularized model formalization, which can improve the performance. The development of the XGBoost model has the same structure as decision tree model and GBDT model, were the sklearn's train_test_split function will be used again to split the data into training and test sets. The model will then be fitted to the training data through the XGBRegressor function. This will include optimizing a loss function to minimize errors between the predicted values and actual values. When the model is fit to the training data, the predict method will be used to make predictions on the testing data. As stated previously this will allow evaluation of the model's performance and evaluate its accuracy in predicting the target variable for new data points.

4.5.3 LSTM

In this chapter the methodology used to develop an LSMT model for time series forecasting, will be discussed. The LSTM model is an effective tool for predicting/forecasting time series data, and its ability to model long-term dependencies makes it suitable for this research goal. The objective is to develop a model using historical data to accurately predict future values of PO4 in the effluent. In order to develop such a model, various steps will be involved, such as data preprocessing, model architecture, training/test, and evaluation.

The development of the LSTM model for time series forecasting/predictions, Keras library, which is a deep learning library in python will be used. Further it necessary to preprocess the data and split the it into training and test sets with the train_test_split function available from the sklearn library. This will allow us to avoid overfitting and evaluate the model's performance on unseen and new data.

To ensure that the LSTM model is able to learn from the input data, it is required to apply feature scaling. This is done by utilizing the StandardScaler function from sklearn library. For models that rely on gradient-based optimization methods such as the one utilized in this study, the scaling technique will help to standardize the range of features. After scaling the data, it was reshaped into a three-dimensional array that fit the input shape requirements of LSTM model. The data was reshaped specifically to have n features and 11 time steps. The LSTM model takes in a sequence of past values as in input and learns to predict future values based on those inputs. We were able to provide the LSTM with the information it required to accurately forecast/predict future values of the target by preparing the data in this manner.

Following that, we proceed to the development of the LSTM model by utilizing the Keras Library. The model architecture is comprised of two LSTM layers, each 256 units, which also included a dropout layer to reduce overfitting and a dense layer to generate the predicted value. The complexity of the model and its ability to recognize patterns in the input data is usually depending on the number of units in the LSTM layer. In broad terms increasing the number of units allows the model to learn more complex representations of the input data, but it also increases the risk of overfitting. Therefore is the dropout layer included in the development of this model. The choice of units used in each LSTM

layer in this study is based on literature and visual demonstrations regarding this field. 256 is frequently used value that has been shown to perform well for a variety of applications, including time series prediction/forecasting (XIN LIU, 2021).

To compile the model, the mean squared error loss function and Adam optimizer is used. Mean squared error loss function measures the averaged squared difference between the predicted and actual value. The Adam optimizer is an effective stochastic gradient descent algorithm that can handle large datasets. It is also implemented early stopping to improve training efficiency and prevent overfitting. This method monitors the validation loss while training and interrupts the process if it fails to improve over a certain number of epochs (XIN LIU, 2021).

To train the model, the same approach used for the other models is used for this model as well, were the data is split into testing and validation sets using the train_test_split function obtained from the sklearn library. 80% of data were utilized for training and 20% were used for validation. Under the training process, the fit method from Keras was used to train the model on the training data and monitor the validation loss to avoid overfitting. Early stopping then was implemented to interrupt the training when the validation loss did not improve for five continuous epochs. After the training is done, the performance of model will be evaluated using the same evaluation metrics is proposed in this study, but to evaluate the performance of the model, the function evaluate(), which is obtained from the Keras library will be utilized instead.

5 Results

The findings of our research of time series prediction of PO4 in the effluent utilizing various machine learning models are presented in this chapter. Firstly we start by discussing the data preprocessing methods that were applied to prepare the data for modelling. Further the performance of several kinds of regression models, decision Tree,
GBDT, XGBoost, LSTM is then evaluated. For each model it will be presented the evaluation metrics score used in this study, which are MSE, MAE and R2. In addition to this, visualization of the predicted value and actual value will be illustrated to better understand the performance of each model. By analyzing and comparing the results, the research aim is to identify the most effective model for predicting future values in the

time series dataset provided by HIAS.

5.1 Data Preprocessing

In order to prepare dataset for analysis for, several data preprocessing steps, were utilized as described in chapter 4. This needed several tasks, such as getting rid of missing values, smoothing noisy data, identifying and removing outliers. It is necessary to keep in mind that unclean data can have significant effect on the results of the mining process, which can lead to inaccurate output. To assure the quality and accuracy of the results of the analysis. Data cleaning is therefore an essential step in the data preprocessing process. To identify the missing values we utilized python 3.10 and Jupyter Notebook. The cubic spline interpolation method was used to identify and fill in missing values using Pandas library as illustrated in figure 13.

```
Out[156]: O2_sone_4                0
          O2_sone_5                0
          O2_sone_6                0
          O2_sone_8                0
          O2_sone_9                0
          Mengde_luft_sone_4       0
          Mengde_luft_sone_5       0
          Mengde_luft_sone_6       0
          Mengde_luft_sone_7       0
          Mengde_luft_sone_8       0
          Mengde_luft_sone_9       0
          Mengde_luft_sone_10      0
          Vannmengde_inn_linje_1   0
          NO2_sone_7_linje_1       0
          NO3_inn_bio              0
          SS_ut_bio                0
          SS_ut_diskfilter         0
          PO4_ut_diskfilter        0
          sCOD_korrigert           0
          dtype: int64
```

Figure 14, Display of the NaN values handled

After handling the missing values, it is crucial to identify potential outliers in the dataset. To identify outliers, we visualized the dataset and noticed inconsistence and irregularity, which was displayed in figure 11 and 12. By using Pandas library the outliers was removed. Figure 14 and 15 illustrates the results of handled outliers.

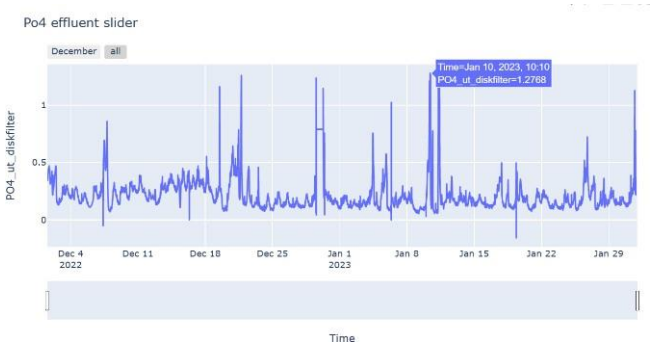

Figure 15, PO4 effluent visualization after outliers are handled

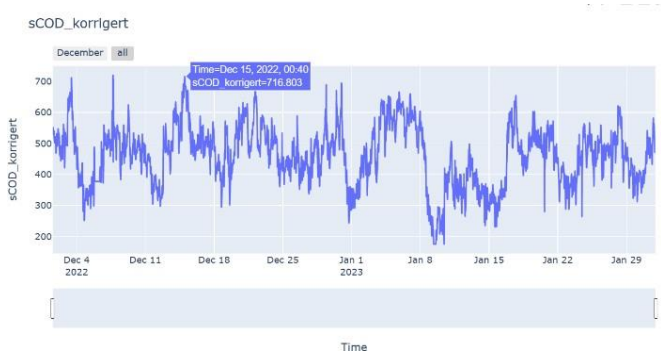

Figure 16, sCOD handled outliers

Next, correlation matrix was created. The correlation matrix provided a table of correlation coefficients between multiple parameters in the dataset. The correlation matrix allowed us to identify which parameters were highly correlated to each other, which parameters were not, and which parameters had a negative or weak correlation. The correlation was further visualized as a heatmap, where red indicates positive and high correlation, blue indicates negative and weak correlation, and colors between indicates no correlates between the parameters, which is also displayed in figure 16. The correlation matrix heatmap, enabled us to extract information to identify the most relevant parameters for the analysis and modeling executed in this study.

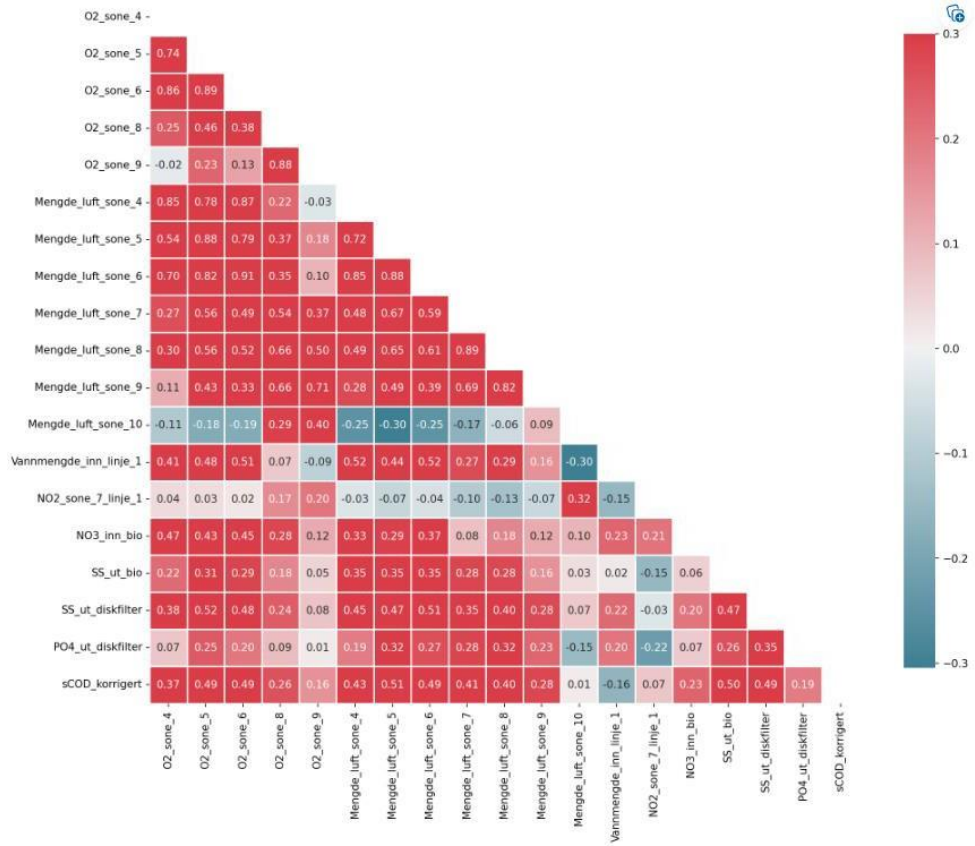

Figure 17, Correlation Matrix heatmap

In addition to the correlation matrix, a pairplot was also created with to visualize the pairwise correlation between the parameters in the dataset. The pairplot allowed us to study how different the parameters were correlated with each other, and to see whether there are any interesting correlation between the parameters. From the pairplot it was observed that some parameters had high positive correlations with each other and others had negative correlation. A visualization of every pairwise correlation between the parameters used in this study are illustrated in figure 17.

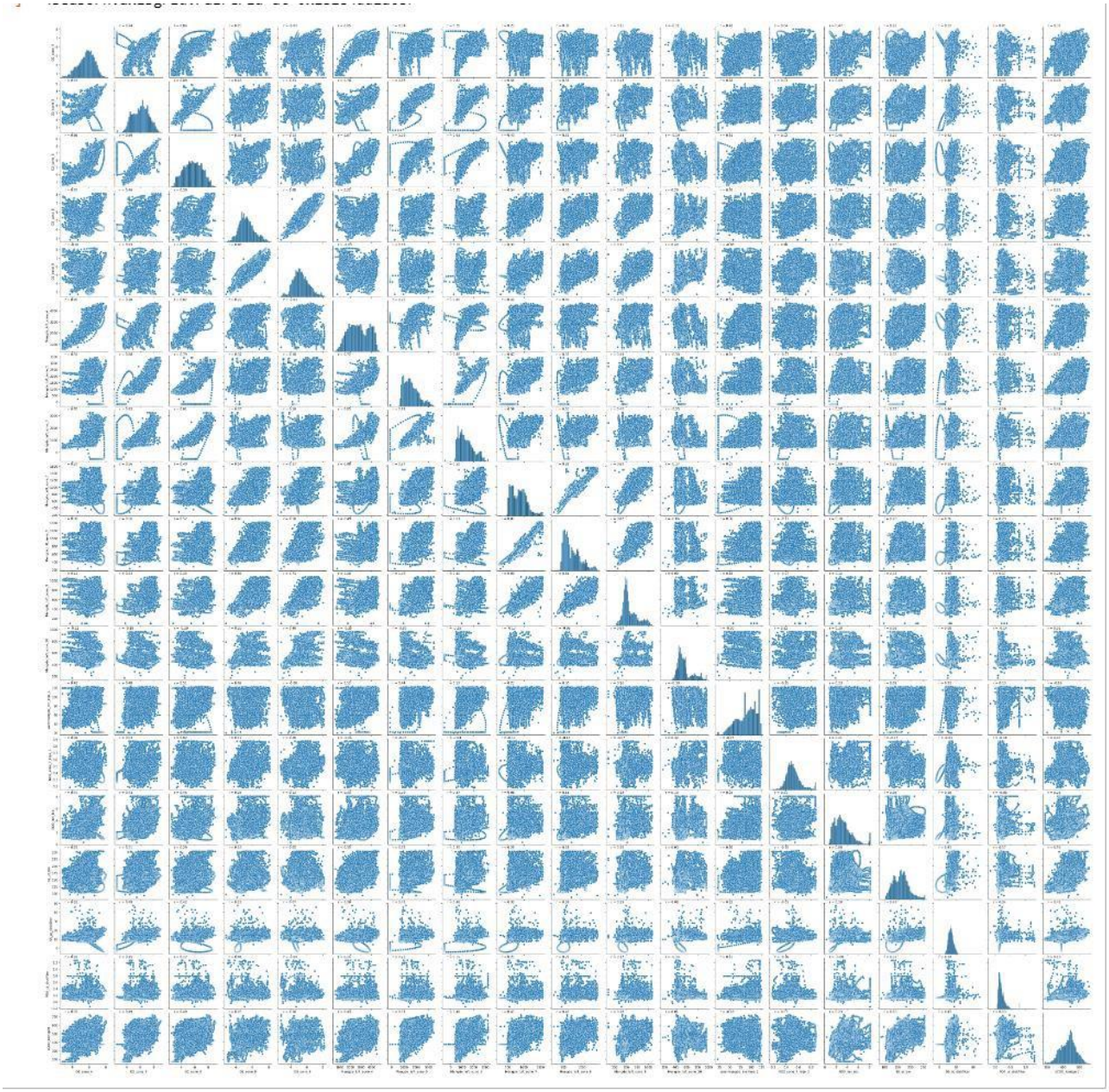

Figure 18, Seaborn pairplot

5.2 Regression

In this section, the results of regression predictive models for predicting the PO4 effluent are displayed. These model's objective is to accurately predict the PO4 in the effluent based on the other parameters. There were several experiments conducted, but it is worth nothing that the results displayed in table 2 are the best ones.

Table 2, Evaluation score of each metrics for regression

Regression	R2	MAE	MSE
Linear 1	0.3642	0.0702	0.0137
Lasso 1	0.0832	0.0802	0.0198
Ridge 1	0.3644	0.0701	0.0137
Linear 2	0.5275	0.1198	0.0756
Lasso 2	0.0155	0.1322	0.1576
Ridge 2	0.5275	0.1197	0.0756

Linear 1 achieved an R2 score of 0.36, MAE score of 0.07 and an MSE score of 0.014, as demonstrated in table 2. Linear 2 performed better on the second attempt, with an R2 score of 0.527, MAE score of 0.119 and an MSE score of 0.075. These results demonstrates that linear 2 was able to capture a larger proportion of the variation in the PO4 effluent and created a more accurate prediction compared to Linear 1.

Next, the predictive performance of the Lasso regression models (Lasso 1 and Lasso 2)was evaluated. The results of Lasso 1 and Lasso 2 are shown in table2, where Lasso 1 displays R2 score of 0.0832, MAE score of 0.0802 and an MSE score of 0.0198. However, Lasso 2 displayed a decrease in performance, obtaining an R2 score of 0.0155, MAE score of 0.1198 and an MSE score of 0.1576. These results show that the Lasso regression models encountered challenges in capturing the correlations between the input variables and the PO4 effluent. The Lasso regression models were unable to produce satisfactory results in this setting, demonstrated by the relatively low R2 score and higher MAE and MSE scores.

Lastly for regression models, the predictive performance of Ridge regression model was evaluated. Ridge 1 displays an R2 score of 0.0364, MAE score of 0.0701, and an MSE score of 0.0137. Ridge performance achieved a R2 score of 0.5275, MAE score of 0.01197 and an MSE score of 0.0756. Displayed in table 2, the results of Ridge performed similarly to the linear regression model, but an important observation is that Linear 2 and Ridge 2 showed improved performance compared to Linear 1 and Ridge 1.

Additionally to the evaluation scores, scatter plots of the regression models for both attempts are presented. The scatter plots visualize the correlation between the predicted values and the actual values of PO4 effluent for each model. The more linear the data points are, the better the prediction of the model are. Linear 1, Lasso 1 and Ridge 1 are displayed below:

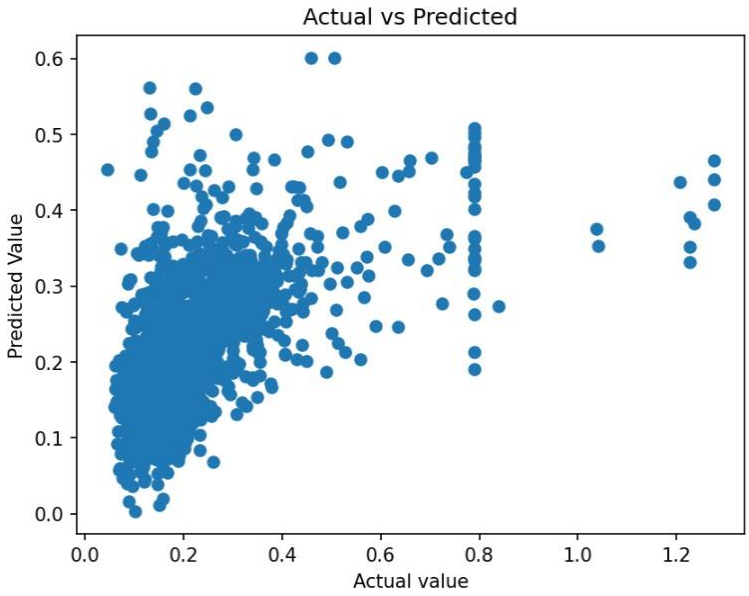

Figure 19, Scatter plots of Linear 1

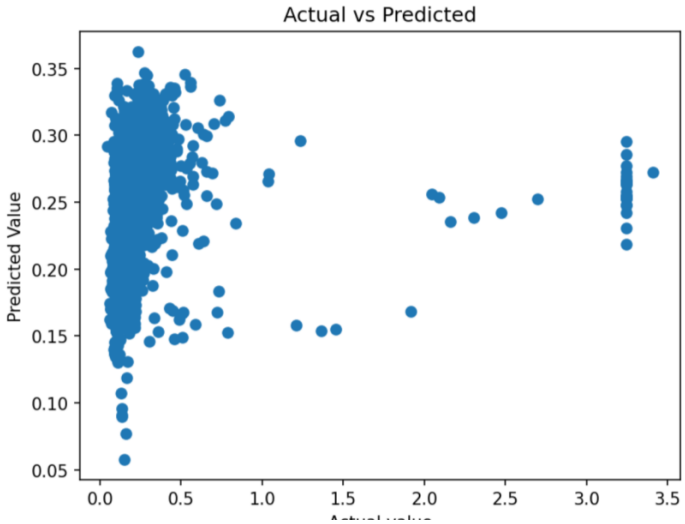

Figure 20, Scatter plots of Lasso 1

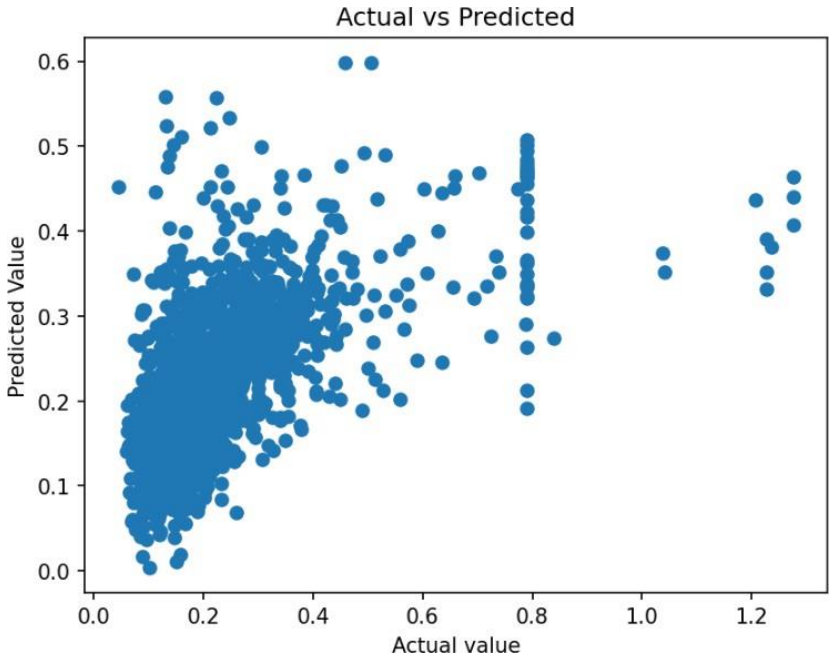

Figure 21, Scatter plots of Ridge1

The scatter plots demonstrates a somewhat good fit for the Linear and Ridge regression models , with the predicted values being relatively near the actual values. On the other had the Lasso regression model shows a poor fit , a wide spread of data points. In the scatter plots, there are some vertical lines that deviates from the values and do not follow the general pattern of data, which indicating outliers.

The scatter plots for each model for the second attempt are displayed below:

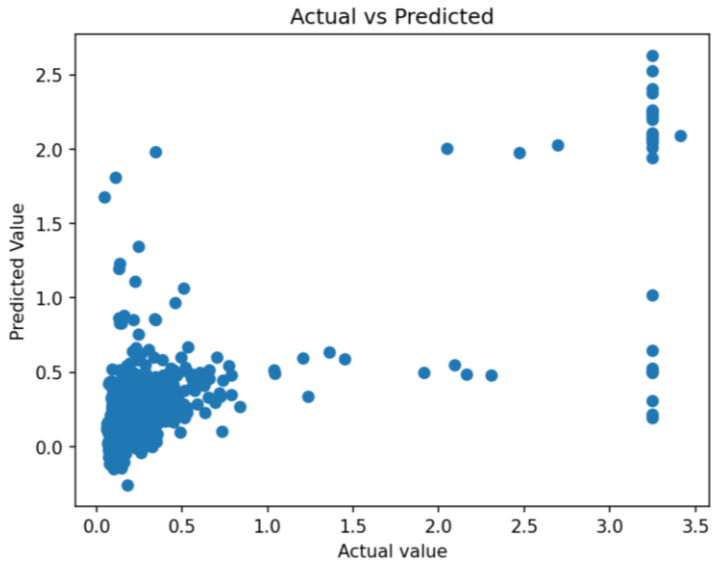

Figure 22, Scatter plots of Linear 2

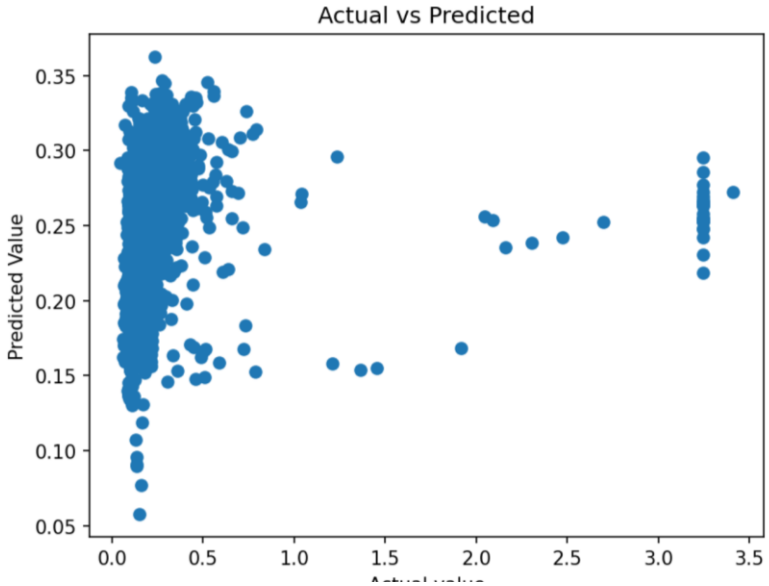

Figure 23, Scatter plots of Lasso 2

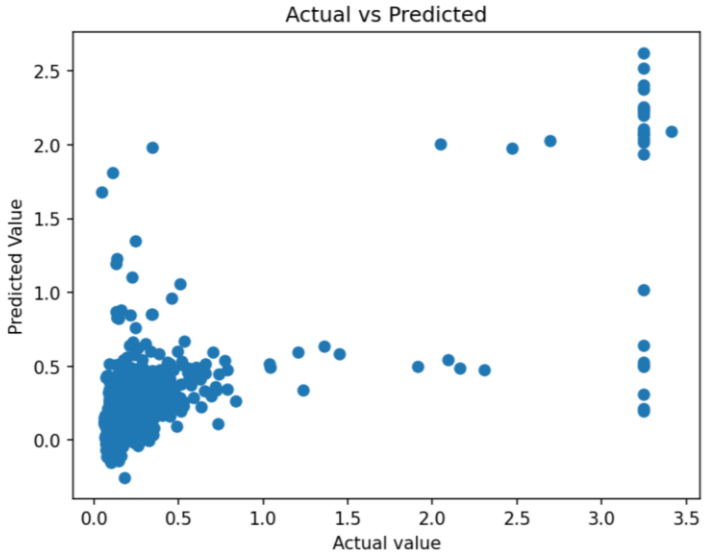

Figure 24, Scatter plots of Ridge 2

The Scatter plots for the second attempt demonstrates an improvement in the predictions accuracy, compared to the first attempt, where the Linear and Ridge regression models displays a better fit, with the predicted values nearer to the actual values. However, the scatter plots of Lasso regression model displays a poor fit, with a clear underprediction of the target variable.

5.3 Data driven multi-class classification

In this study, Data driven multi-class classification models was also evaluated to see if it accurately can predict PO4 effluent using the same parameters, and a similar approach used for the regression models was used for DT and GBDT models. Several experiments were conducted to evaluate the performance of the DT model and GBDT model. However, note that the result presented in table 3 represent the best performance achieved in this study as well.

Table 3, Evaluation score of each metrics for Classification

Classification	R2	MAE	MSE
DT 1	0.708	0.0310	0.0063
GBDT 1	0.757	0.0459	0.0052
XGBoost	0.567	0.058	0.0096
DT 2	0.736	0.040	0.042
GBDT 2	0.869	0.059	0.020

DT 1 achieved an R2 score of 0.708, MAE score of 0.0310 and an MSE score of 0.0063. GBDT 1 performed better, with an R2 score of 0.757, MSE score of 0.058 and MSE score of 0.0096. For DT 2 and GBDT 2, we see an increase in performance, where DT 2 displays an R2 score of 0.736, MSE score of 0.040 and MSE score of 0.042. However, for the second attempt , GBDT achieved a significant increase in performance with an R2 score of 0.869, MAE score of 0.059 and MSE score of 0.020 and outperformed the DT model. Overall the result displays that both DT and GBDT models shows promise to develop accurate data-driven multi-class classification models for predicting the PO4 effluent.

Next, the results for the XGBoost was also displayed in table 3, since it provided a decent result. Only one attempt is displayed, because the others attempts neither showed increase or decrease in performance. The XGBoost model achieved an R2 score of 0.567, MSE score of 0.058 and MSE score of 0.0096. The result demonstrates that the XGBoost showed a moderate performance in predicting the labels of the PO4 effluent variables.

In addition to the evaluation scores, the scatter plots are also presented in this section to visualize the correlation between the predicted values and the actual values of PO4 effluent. Scatter plot of DT, GBDT and XGBoost are all displayed below:

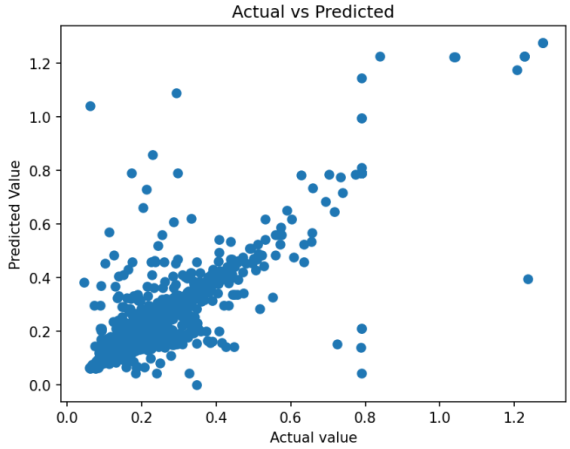

Figure 25, Scatter plots of DT 1

The scatter plot for DT 1 displays a clear correlation between the predicted values and the actual values of PO4 effluent values and a somewhat linear line. Where the scatter plot shows a satisfactory level of prediction accuracy for the model.

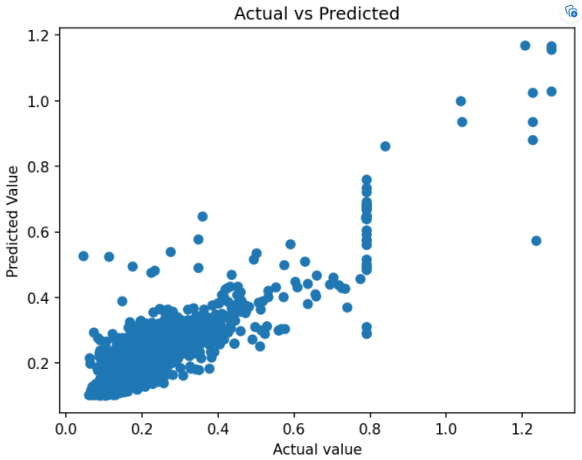

Figure 26, Scatter plots of GBDT 1

The scatter plot of GBDT 1 displays a better result, where it demonstrates a good fit and better alignment of the predicted values and actual values. Compared to DT 1, this shows an improved accuracy in the predictions.

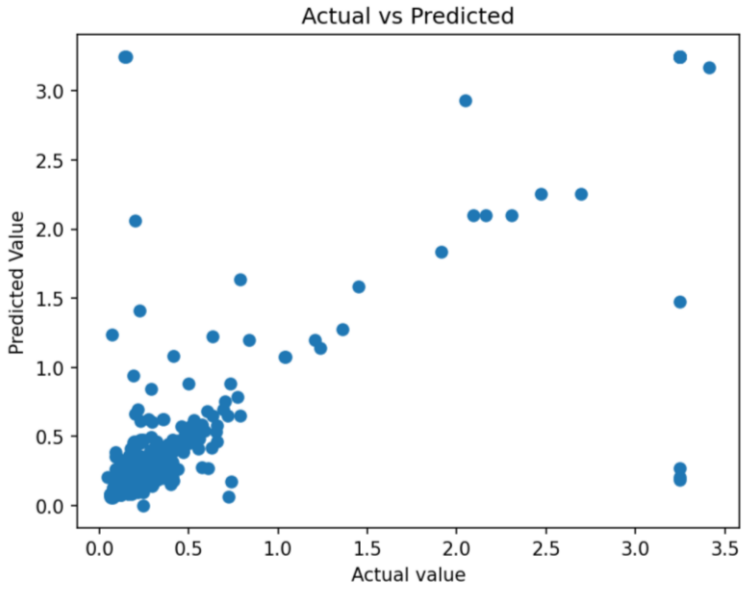

Figure 27, Scatter plot of DT 2

DT 2 also demonstrates how well the model fits the data. Since the R2 score of DT 2 is on a satisfactory level and shows a strong alignment of the predicted values and actual values, which is indicating a high level of accuracy in the model's prediction.

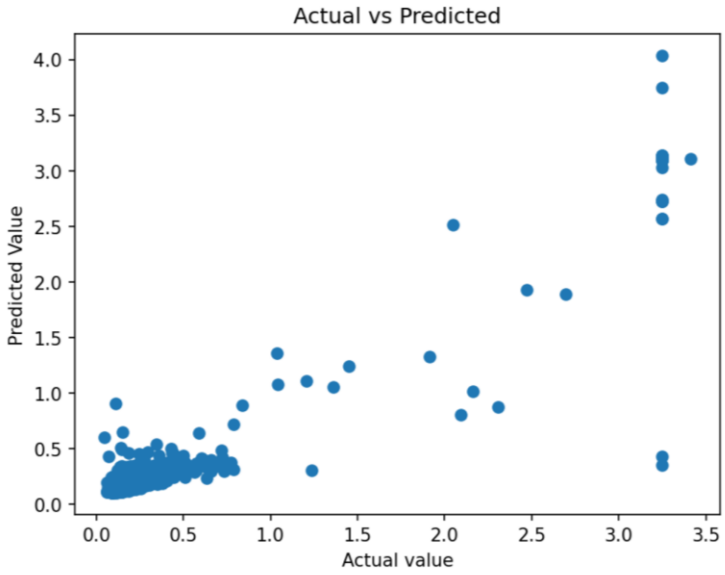

Figure 28, Scatter plot of GBDT 2

GBDT 2 demonstrates great improvement in the model's performance, where the R2 score and the scatter plot illustrates a better fit to the data and a stronger predictive capability.

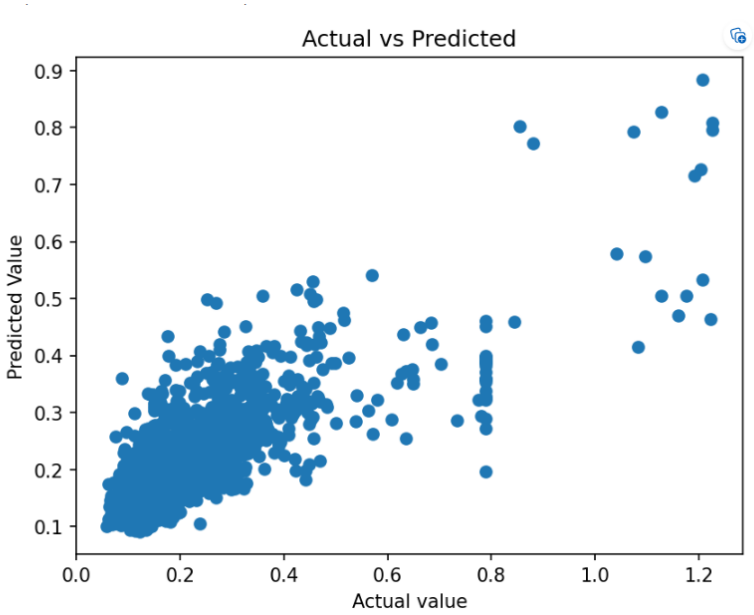

Figure 29, Scatter plot of XGBoost

The scatter plot of XGBoost model displays a decent alignment between the predicted values and the actual values, which demonstrates a moderate level of accuracy in the prediction of the model.

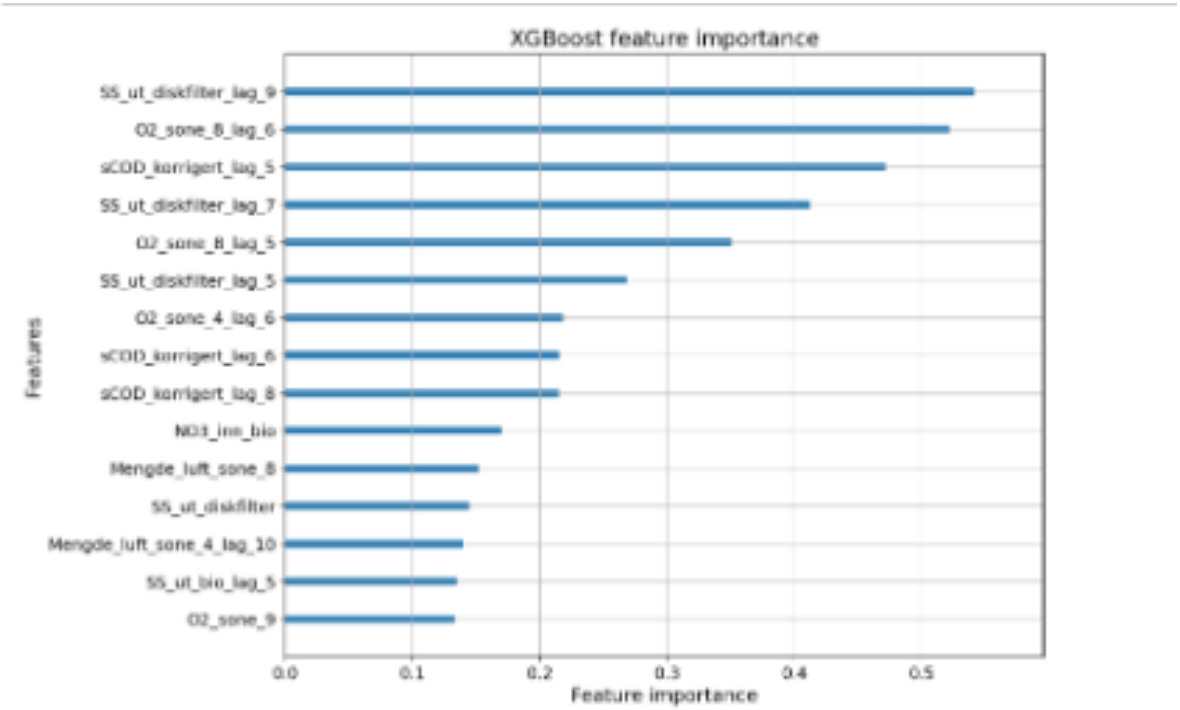

Figure 30, Display of parameters importance in the prediction of XGBoost model.

5.4 LSTM

LSTM was the last model conducted to observe if it was able to predict the PO4 effluent based on a time series dataset. Therefore, in this section the results of LSTM predictive model are displayed. There were several experiments conducted, but only the best result is worth displaying. The results are displayed on table 4.

Table 4, Evaluation score of each metrics for LSTM

LSTM	R2	MAE	MSE
Attempt 1	0.8678	0.0322	0.0534
Attempt 2	0.9259	0.0223	0.0399

The results displayed how effective the LSTM model is in predicting values of PO4 in the effluent. The first attempt illustrated in table 1, shows that the model achieved an R2 score of 0.8678. Based on the R2 score the model is indicating a satisfactory level of accuracy and the ability to capture the patterns and trends in the data. The models also achieved an MAE score of 0.0322 and MSE score of 0.0534, where the score of MAE illustrates a reasonable average deviation between the predicted values and the actual values. On the other hand, the MSE is indicating reduction in the overall squared difference between the actual and predicted values.

Attempt 2 displays an experiment with further training of the same model. Attempt 2 showed improved performance compared to Attempt 1, where R2 score increased to 0.9259 and demonstrates high level accuracy in capturing patterns and trends of the data. The MAE decreased to 0.00223 and shows a reduction of average deviation between predicted values and actual values. The MSE also shows a decrease, where the core is 0.0399 indicating more reduction in the overall squared difference between actual and predicted values. Scatter plots is also utilized for LSTM to visualize the alignment between predicted values and actual values of PO4 in the effluent.

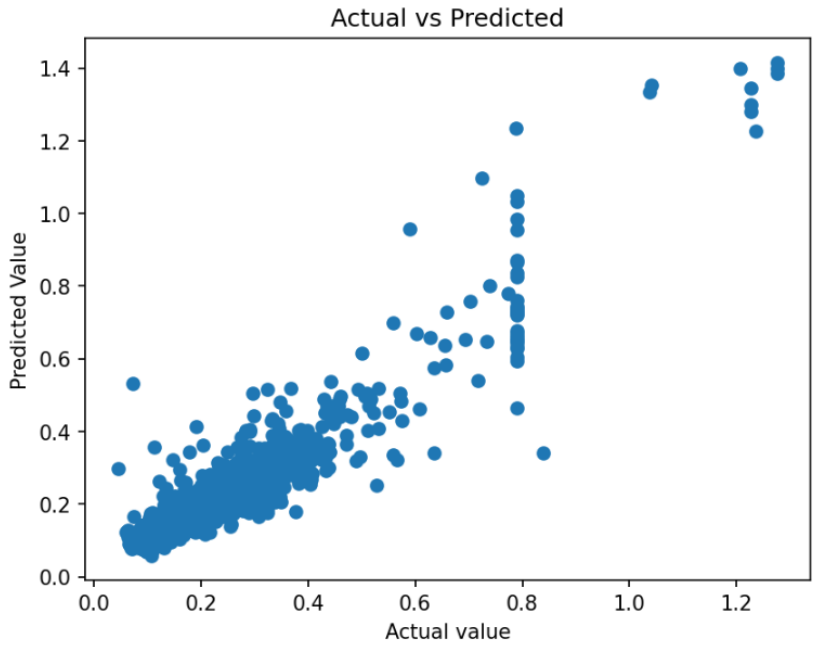

Figure 31, Scatter plot of Attempt 1 (LSTM)

The scatter plot for Attempt 1 displays the predicted values plotted against the actual values of PO4 in the effluent. The scatter plots demonstrates a great sequence between the predicted values and the actual values, which indicates a high satisfactory level of accuracy in the model's prediction .

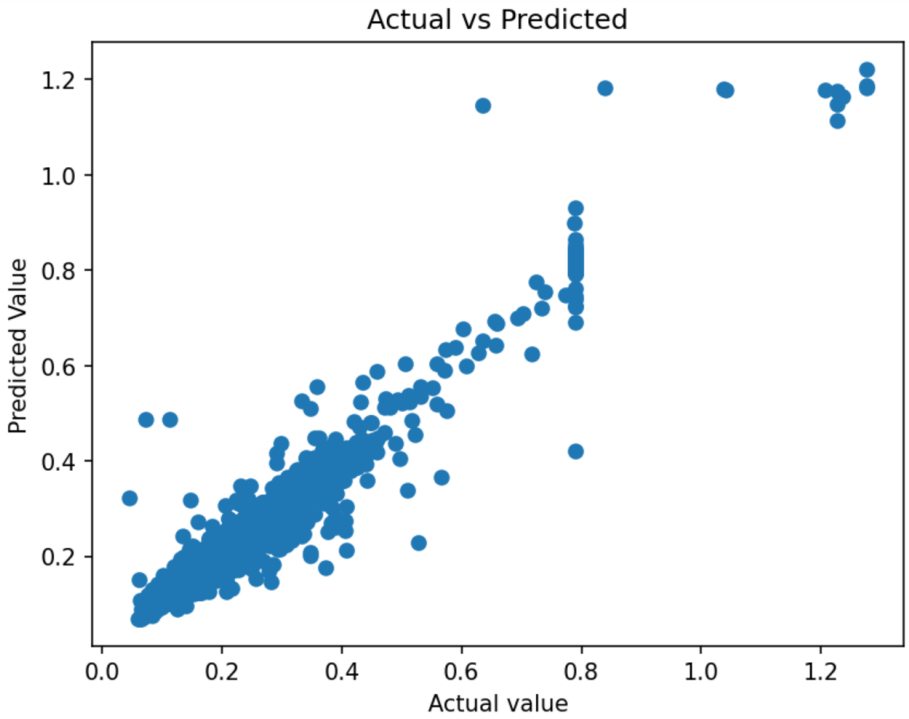

Figure 32, Scatter plot of Attempt 2 (LSTM)

For the Attempt 2 the scatter plot also displays how well the predicted values are plotted against the actual values of PO4 effluent. Attempt 2 demonstrates an even better sequence between the predicted values and the actual values, which suggest an improved performance of the model predictions compared to Attempt 1.

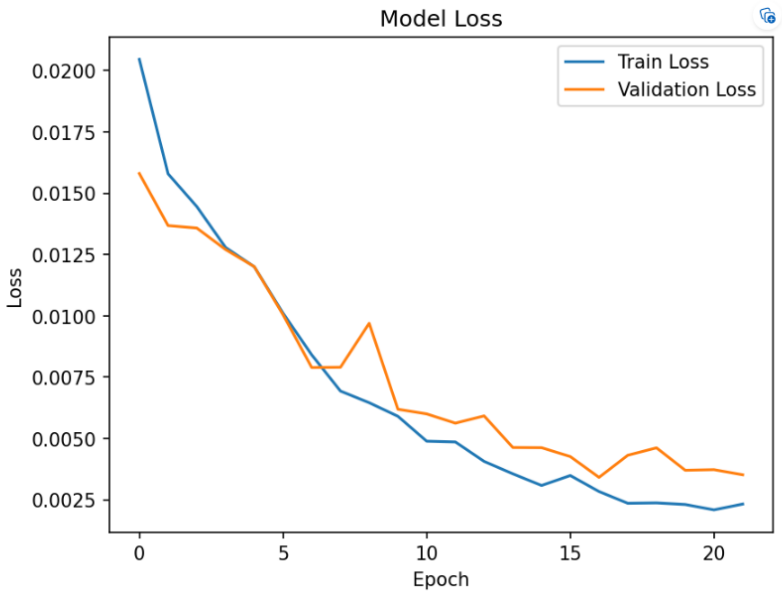

Figure 33, Display of Model Loss of Attempt 1 (LSTM)

In addition to the scatter plots and evaluation of the performance of the LSTM model, the model loss was also displayed. Figure 34 demonstrates the Train and Validation Loss throughout the training process. The Train loss curve illustrates the model's ability to fit the training data over consecutive epochs. As displayed in figure 34 the Train Loss curve starts with a higher value as the model begins learning and adjusting to the parameters. As the training continues, the Curve gradually decreases, which indicates that the model is improving its predictive performance on the training set.

The Validation Loss curve, on the other hand, gives an estimate of how well the model performs on unseen data. As displayed in the figure above, the validation loss follows a similar decreasing trend as the Train Loss curve. There are few increase in the trend, which may indicate that model is overfitting to the training data.

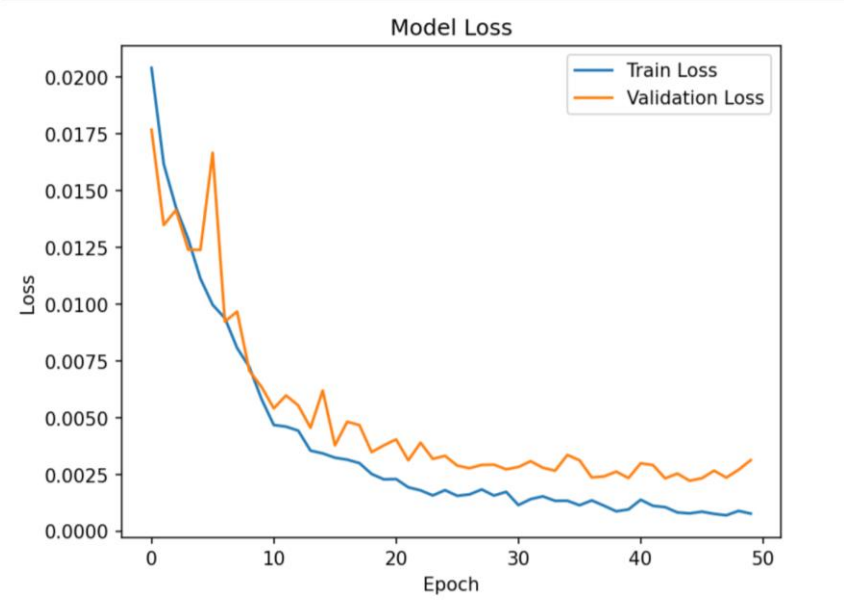

Figure 34, Display of Model Loss of Attempt 2 (LSTM)

The Model Loss of Attempt 2 demonstrates something similar to Attempt 1, where the decreasing trend of both Train and Validation Loss is even greater in figure 34. There are also some few increase in the trend, which also indicates overfitting.

6 Discussion

The objective of this study was to predict the PO4 concentration in the effluent of HIAS wastewater treatment plant using various machine learning models. In this chapter the effectiveness and performance of the three different regression models are proposed in this study. Additionally, the data driven multi-class classification methods are explored for predicting PO4 concentration in the effluent and lastly, the investigation of the LSTM model, for predicting the same target parameter will all be discussed. The performance of these models are evaluated to gain insight into their strengths, weaknesses and to see if there are any notable differences in accurately predicting the PO4 in the effluent. Through this chapter, we aim to identify the most suitable model for predicting the PO4 concentration in the effluent based on time series dataset.

6.1 Regression

Linear, Lasso and Ridge regression models were all evaluated based on their performance in Predicting PO4, utilizing the evaluation metrics. The Result based on table 2 showed that Linear 2 and Ridge 2 had the best performance accuracy when predicting PO4 in the effluent. Both Linear 2 and Ridge 2 achieved an R2 score of 0.5275, MAE score of 0.1198 and MSE score of 0.075, which are decent. Since the R2 score ranges from 0 to 1, where 0 indicates poor fit and the model does not explain any the variance in the target variable, 1 on the other hand indicates a perfect fit. Therefore we want to achieve a relative high R2 score to know whether the model to the data is a good fit. An R2 score of 0.75 indicates a great or substantial fit, an R2 score of 0.50 indicates a moderate fit, anything under is considered weak. Since Linear 2 and Ridge 2 demonstrates a relatively moderate fit, it suggest that Linear and Ridge regression is capturing a sizeable percentage of the variation in the PO4 effluent based on the parameters used in this study. It also suggest that there are still some variations in the sequence that are unexplained and influenced by factors not included in the model, which may be the reason that the accuracy of the prediction is not on a satisfactory level. It is worth noting that the R2 score of 0.5275 is our best result for Linear and Ridge. In our first attempt we achieved an R2 score of 0.364, which is below moderate. The increase in score happened after training the model several times, which allows them to learn from data and make adjustments to the provided parameters. A thought, is that during the first attempt, model was not able to capture the patterns in the data, and is the reason for the weak score.

Lasso 1 and Lasso 2, however demonstrates a very poor and weak fit, where the R2 score is 0.0832 and 0.0155. These result indicates that they were unable to capture the correlation between the input parameters and the PO4 concentration in the effluent. The fact Linear and Ridge regression were able to capture non-linear correlations between the input parameters and the target parameter may be one explanation for their stronger performance. The reason for this may be because these models utilized regularization methods to prevent overfitting, while the Lasso regression model may have removed important parameters from the model due to their L1 regularization penalty and also may struggle when correlations are complex. Over all Linear and Ridge Regression models demonstrated better performance than Lasso regression in predicting the PO4 concentration In the effluent.

6.2 Data driven multi-class classification

When comparing the performance of DT and GBDT models, it is apparent that both models demonstrate potential in developing accurate data-driven multi-class classification models for predicting PO4 in the effluent. DT 1 displayed a R2 score of 0.708, on the other hand GBDT achieved an R2 score of 0.757. Both DT 1 and GBDT demonstrates a moderate fit of the model, where they was successfully able to capture a larger proportion of the variation in the PO4 effluent. Even if they both demonstrated good results, GBDT 1 outperformed DT 1, which indicates that GBDT is able to capture an even bigger proportion of the variation in the PO4 effluent and generates more precise predictions compared to DT 1. In the second attempts, notable improvements in performance was observed for both DT 2 and GBDT 2 models and demonstrated an enhanced predictive ability compared to the previous attempts. However, It is worth nothing that GBDT 2 showcased significant improvements and outperformed the other attempts with an R2 score of 0.869. These results imply that GBDT model is capable to identify the underlying correlations and patterns in time-series data, which provides more accurate predictions of the PO4 concentration in the effluent. Nevertheless, the XGBoost model displayed a moderate performance where it only achieved an R2 score of 0.567 when predicting the labels of the PO4 effluent variables. The XGBoost model was able to provide a decent accurate prediction, but it did not achieve the same level of performance as the GBDT model.

The DT model has notable strengths in terms of simplicity and comprehension. The decision making process is simple to comprehend and visualized, which is beneficial for gaining insights about the predictive factors for the PO4 effluent. The risk of overfitting with DT models must, however, be taken into consideration, especially when the three depth is not well controlled. Overfitting can lead to excessively complex models that may not generalize well to new data. Therefore to reduce the risk of overfitting in DT models, careful parameter tuning and regularization methods are necessary. GBDT model, in contrast display significant abilities to handle complex correlations and identify non-linear patterns in the data. The iterative nature of

GBDT allows it to correct errors made by previous trees, resulting in increased overall performance. However, due to sequential nature of tree construction, it is important to consider that GBDT models can be computational demanding, especially when dealing with large datasets. Furthermore, without a proper regularization methods, GBDT can be vulnerable to overfitting, which could reduce the generalization ability. As a result, careful regularization and tuning of hyperparameters are important to balance between the performance in GBDT models and model complexity. Fortunately, the GradientBoostingRegressor library available in python solve majority of this.

In this study, the XGBoost model demonstrated a moderate level of accuracy in predicting the PO4 concentration in the effluent. While XGBoost has attributes such as scalability, efficiency and the ability to handle large datasets, its performance fell short compared to LSTM, DT and GBDT. There could be several reasons for the poor performance of XGBoost in this study, such as insufficient hyperparameter tuning and optimizers. The XGBoost model requires careful hyperparameter tuning to achieve optimal performance. One possibility in this study, may be that they were not finely tuned, which resulted to the moderate performance of XGBoost. Further experimenting and investigation with different combinations of hyperparameters and optimizers could potentially improve the model's performance.

6.3 LSTM

The LSTM model utilized in this study demonstrated significant promise in accurately predicting the PO4 concentration in the effluent based on the time series dataset, where in Attempt 1 we achieved an R2 score of 0.868 and achieved an R2 score of 0.926 in Attempt 2. If we follow the benchmark set for the R2 score proposed in this study, they both demonstrates substantial and excellent model fit, where 1 indicates perfect model fit. Specifically notable was the improvement noticed in Attempt 2 compared to Attempt 1, demonstrating the model's ability to identify complex patterns and trends in data. One of the advantages of the LSTM model is its ability to handle time-series data by considering temporal dependencies. The improvement achieved can be because to the iterative training process used for the LSTM model. The model obtains the ability to identify and make use of the temporal patterns present in the data by being trained multiple times. The iterative training allows the LSTM model to improve its internal representations and identify dependencies over time. As a result, the model becomes better at comprehending the complex correlations and patterns of the time-series dataset, which improved its ability to predict the PO4 concentrations in the effluent accurately.

Sequential the data must be incorporated by the model in order to accurately estimate the PO4 in the effluent, which is enabled by the recurrent architecture. By identifying long-term dependencies, the LSTM

model can effectively learn and understand the underlying patterns in the data. However, like any model, the LSTM model also has its weaknesses. One important consideration is necessary to optimize the model to fullest, where it requires rigorous parameter tuning. Selecting suitable hyperparameters, such as the number of LSTM layers, the learning rate, number of hidden units, is all essential for achieving optimal performance. Since the accuracy of the model's prediction can be affected by unsuitable parameter selection, which can result in overfitting or underfitting. Furthermore, the LSTM model can be computationally demanding, especially when working with large datasets. The model is computationally more demanding than other models due to its recurrent nature and the requirement to convey information over multiple steps. Therefore, it is important to implement and assign computational resources in order to effectively train LSTM models. In our LSTM model various computational resources was utilized to handle the complexity, such as StandardScaler function from sklearn library, Reshaping and optimizers like Adam, which are all described in chapter 4.5.3.

6.4 Outliers

In the investigation of scatter plots for regression, data-driven multi-class classification and LSTM model, we were able to observe sudden vertical lines of data points that was spread away from the rest of the scatter plot. After further examination, this random vertical line of data points can indicate outliers that may have appeared in the model's predictions. The appearance of outliers can have several effects on the model. One potential reason for the appearance of outliers may be caused under the prediction of PO4, where a sample of data frequently deviates in its predictions. These deviations could develop from errors in the data collection or patterns in the dataset that the model fails to identify accurately. The effectiveness of the model and the accuracy of its predictions can be affected by the presence of outliers or in this case a clear vertical line in the scatter plot. Outliers may have an effect on the models' fitting process, which could result in inaccurate predictions and biased estimates. For the model to be robust and generalizable, these outliers must be identified and dealt with. Therefore , further investigation into the causes of these outliers and the development of robust outlier detection and handling methods are recommended, where the removal of outlier could improve the model's accuracy and ensure more reliable predictions.

7 Further work

Although the research in this study has proved valuable insights into predicting the PO4 concentration in the effluent using different machine learning models, there are still several areas that need further research and development. To improve the findings and open up new areas for investigation, the following suggestions are presented below:

1. **Feature selection**: Predefined Parameters were used as inputs in this model. However, investigating which parameters could optimize the prediction accuracy, since having to many features may lead to overfitting. Feature selection overall could help to remove redundant or irrelevant features that may negatively affect the model's performance. Feature selection methods could be applied to identify the most informative features for prediction.

2. **Hybrid Model:** Investigating potential hybrid models that could work together with different machine learning models could be a research for further work. For example, combining MPC model with LSTM model, where the integration of these two models could potentially enhance the accuracy and control the abilities of the system.

3. **Separate the Dataset:** In this study, the dataset use for analysis consist of December and January. December have some unique challenges due to the potential factors and missing values that affect variables in a wastewater treatment plant. Factors, such as the industrial activities during the holiday season, change in water usage pattern, weather conditions, increased organic loads etc. are commonly observed during this month. By predicting each month individually, the model can capture these seasonal patterns more accurately, which may result in improved predictions. The months have their characteristics, and modeling them separately allows better understanding of the underlying dynamics.

8 Conclusion

In this study, we investigated various machine learning models for predicting the PO4 concentration in the effluent of a wastewater treatment plant. Regression model, such as Linear, Lasso and Ridge were evaluated, where Linear and Ridge demonstrated the best performance among the regression models achieving a moderate fit with an R2 score of 0.5275. Lasso showed weak performance, suggesting that its L1 regularization penalty may have removed important parameters from the model. Data driven multi-class classification models (DT and GBDT) were also evaluated, where GBDT outperformed the other classification models, with an R2 score of 0.869, indicating a good fit. The XGBoost on the other hand demonstrated a moderate accuracy compared to other classification models, highlighting the need for careful hyperparameter. The LSTM model displayed significant promise in accurately predicting the PO4 in the effluent based on time-series dataset. It achieved an substantial or excellent fit with R2 score of 0.926. The LSTM model's ability to capture temporal dependencies and identify complex pattern in time-series data contributed to its improved predictive ability.

It is important to note that outliers were observed in the scatter plots of the regression, classification and LSTM models. These outliers could affect the model's accuracy and prediction abilities. Therefore, to increase the accuracy of the model's predictions, further investigation into what causes the outliers is suggested as well as the development of robust outlier detection.

In conclusion, the LSTM model proved to be the most suitable and effective model for predicting the PO4 concentration in the effluent of the wastewater treatment plant, considering the complex environment and the time-series nature of the data. Its ability to capture complex correlations, handle temporal dependencies and identify patterns made it dependable choice for accurate predictions.

9 References

Anthony J.Myles, R. N. (2004). An introduction to decision tree modeling. *JOURNAL OF CHEMOMETRICS*, 275-285.

Didrik Villard, T. S. (2022). Spatial fractionation of phosphorus accumulating biofilm: stratification ofpolyphosphate accumulation and dissimilatory nitrogen metabolism. *BIOFOULING*.

Dong Wang, S. T. (2021). A machine learning framework to improve effluent quality control in. *Science of the Total Environment*.

Dong Wang, S. T. (2021). A machine learning framework to improve effluent quality control in wastewater treatment plants. *Science of the Total Environment*.

Ekasit Kijsipongse, S. U.-r. (2011). *Efficient Large Pearson Correlation Matrix Computing using Hybrid MPI/CUDA.* IEEE.

Haidi Rao, X. S. (2019). Feature selection based on artificial bee colony and gradient boosting decision tree. *Applied Soft Computing* , 634-642.

Hailin Li, X. W. (2014). Dynamic time warping based on cubic spline interpolation for time series data. *IEEE*.

Hui Chen, L. J. (2020). An Efficient Hardware Architecture with Adjustable Precision and Extensible Range to Implement Sigmoid and Tanh Functions. *Electronics* .

Icke, O., Es, D. M., Koning, M. F., Wuister, J. J., Ng, J., Phua, K. M., . . . Tao, G. (2020). Performance improvement of wastewater treatment processes by application of machine learning. *Water Science & Technology*.

Knut Rudi, I. A. (2019). Microbial ecological processes in MBBR biofilms for biological phosphorus removal from wastewater . *Water Science & Technology*.

Maddi Etxegarai, M. C. (2022). Virtual Sensors for Smart Data Generation and Processing in AI-Driven Industrial Applications. *Industry 4.0 - Perspectives and Applications*.

Mahesh, B. (2018). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*.

Mayooran Thevaraja, A. R. (2019). Recent Developments in Data Science: Comparing Linear, Ridge and Lasso Regressions Techniques Using Wine Data. *International Conference on Digital Image and Signal Processing* .

Mohamed Sherif Zaghloul, G. A. (2022). Application of machine learning techniques to model a full-scale wastewater treatment plant with biological nutrient removal. *Journal of Environmental Chemical Engineering*.

Nashia Deepnarain, M. N. (2019). Decision tree for identification and prediction of filamentous bulking at full-scale activated sludge wastewater treatment plant. *Process Safety and Environment Protection* , 25-34.

Nguyen Duc viet, A. J. (2023). Machine learning-based real-time prediction of micropollutant behaviour in forward osmosis membrane (waste)water treatment. *Journal of Cleaner*

Production.

Quang Viet Ly, V. H. (2022). Exploring potential machine learning application based on big data for prediction of wastewater quality from different full-scale wastewater treatment plants. *Science of the Total Environment*.

Raed Jafar, A. A. (2022). Predicting Effluent Quality in Full-Scale Wastewater Treatment Plants Using Shallow and Deep Artificial Neural Networks. *Sustainability*.

Rui Wang, Y. Y. (2021). Model construction and application for effluent prediction in wastewater treatment plant: Data processing method optimization and process parameters integration. *Journal of Environmental Management*.

Sima Siami-Namini, N. T. (2018). A Comparison of ARIMA and LSTM in Forecasting Time Series. *IEEE International Conference on Machine Learning and Applications*.

Sky McKinley, M. L. (1998). *Cubic Spline Interpolation.* College of Redwoods.

Thulane Paepae, P. N. (2022). A Virtual Sensing Concept for Nitrogen and Phosphorus Monitoring Using Machine Learning Techniques. *Sensors*.

Tianqi Chen, T. H. (2017). *xgboost: eXtreme Gradient Boosting.*

Ugo Barry, J.-M. C.-p. (2017). A one dimensional moving bed biofilm reactor model. *Bioprocess Biosyst Eng*.

Wei Lin, Y. H. (2022). Prediction of wastewater treatment system based on deep learning. *Frontiers in Ecology and Evolution*.

Wongburi, P. (2021). Big Data Analytics from a Wastewater Treatment Plant. *Sustainability* .

XIN LIU, Q. S. (2021). Using LSTM Neural Network Based on Improved PSO and Attention Mechanism for Predicting the Effluent COD in a Wastewater Treatment Plant. *IEEE Access*.

Zaghloul, M. S. (2022). A review of mechanistic and data-driven models of aerobic granular sludge. *Journal of Environmental Chemical Engineering*.

Zahraa Said Abdallah, G. I. (2017). Data Preparation. *Encyclopedia of Machine Learning and Data Mining*.

Zhang, Y. (2022). Development of hybrid machine learning model for simulation of chemical reactors in water treatment applications: Absorption in amino acid. *Environmental Technology & Innovation*

APPENDIX

The appendix for this study are available as a separate zip file. The zip file contains materials, such as code files and its results for all the models. The code files included in the appendix can be opened and executed in Jupyter Notebook. Please refer to the provided zip file for the complete set of appendices with this study.