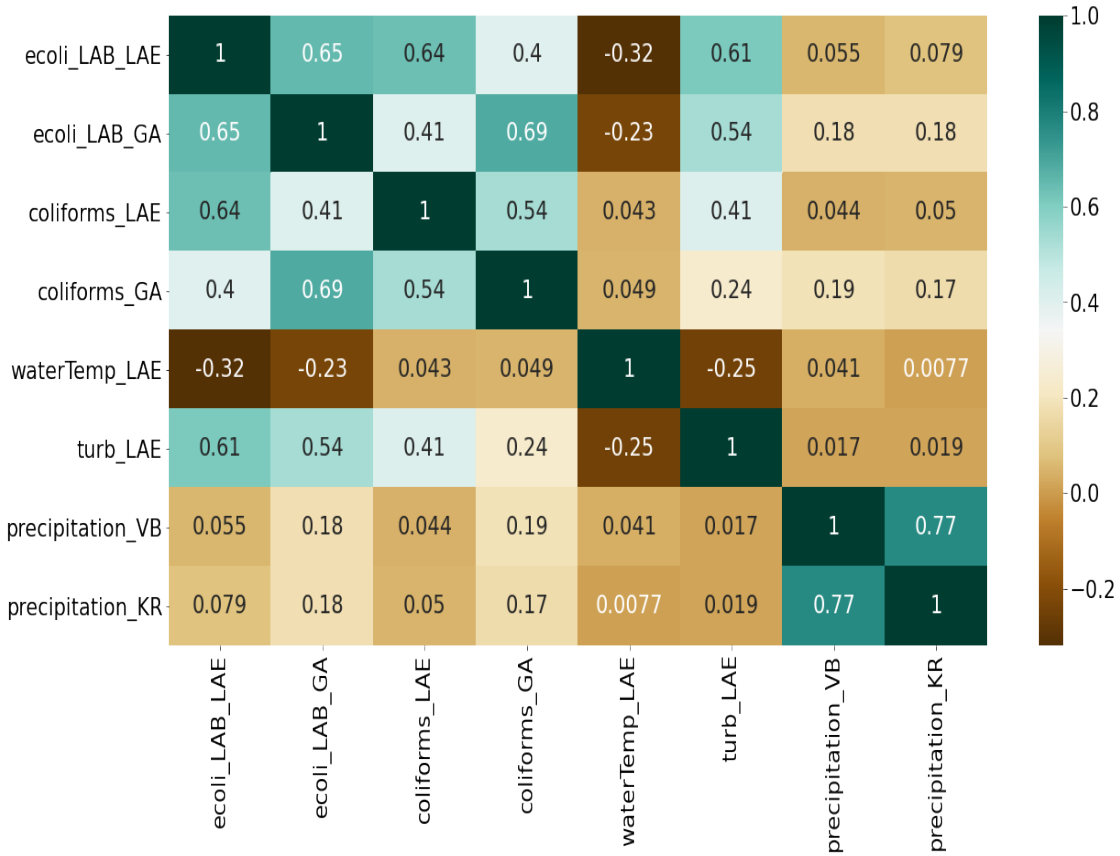


Predictive analysis of *E. coli* levels to assess water quality in the river Göta älv.

Master's Thesis in the Master's Programme Industrial Ecology

Ramachandran Ravishankar

DEPARTMENT OF ARCHITECTURE AND CIVIL ENGINEERING
DIVISION OF WATER ENVIRONMENT TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

Master's thesis 2021

Predictive analysis of *E. coli* levels to assess water quality in the river Göta älv

Ramachandran Ravishankar

Master's Thesis in the Master's Programme Industrial Ecology

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Architecture and Civil Engineering
Division of Water and Environment Technology
Chalmers University of Technology
Gothenburg, Sweden 2021

Predictive analysis of *E. coli* levels to assess water quality in the river Göta älv.

Master's Thesis in the Master's Programme Industrial Ecology
Ramachandran Ravishankar

Examensarbete ACEX30
Institutionen för arkitektur och samhällsbyggnadsteknik
Chalmers tekniska högskola, 2021

Department of Architecture and Civil Engineering
Division of Water Environment Technology
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: correlation matrix between features in the dataset.

Typeset in LaTeX
Gothenburg, Sweden 2021

Predictive analysis of *E. coli* levels to assess water quality in the river Göta älv.
Master's Thesis in the Master's Programme Industrial Ecology

Ramachandran Ravishankar
Department of Architecture and Civil Engineering
Division of Water and Environment Technology
Chalmers University of Technology

Abstract

Water quality is one of the most important factors in a clean and hygienic environment. Sewage waste from the city contains harmful faecal pathogens that when led to the river may contaminate the quality of the water. In this study, a widely used faecal indicator, known as *Escherichia coli* or *E. coli*, is predicted at Lärjeholm drinking water intake plant.

An initial dataset was compiled using the raw data points obtained from Göteborg Kretslopp och Vatten and Swedish Meteorological and Hydrological Institute (SMHI). Data preprocessing steps, such as $log_{10}(x+1)$ transformation, time indexing, removing duplicate values, filling missing values, and defining lag values were carried out on the initial dataset. After preprocessing, the initial dataset was split into baseline and complex datasets. The baseline dataset contains lag values of precipitation at Komperöd and Vänersborg and water temperature at Lärjeholm to predict *E. coli* levels at Lärjeholm, while complex dataset, an upgraded version of the baseline dataset with additional features such as lag values of *E. coli* at Garn, turbidity at Lärjeholm, coliforms at Lärjeholm and Garn.

Linear models Multivariate adaptive regression splines (MARS), and Elasticnet regression and a non-linear tree-based model Extreme Gradient Boosting (XGBoost) regression were used for the prediction of *E. coli* levels. Elasticnet regression was the most efficient algorithm with a mean absolute error of 77 (CFU/100 ml), root mean squared error of 125 (CFU/100 ml) and R^2 score of 0.46. MARS was the least efficient with an mean absolute error of 86 CFU/100 ml, root mean squared error of 154 (CFU/100 ml) and R^2 score of 0.22. Though XGBoost was expected to perform better than linear model such as Elasticnet, it failed to do so. However, the relative error change (Δ error) for XGBoost was around 43% from baseline to complex dataset, the highest improvement rate among all three models with the addition of new features into the dataset.

The study uses machine learning algorithms as a complement to expensive lab analysis to analyse and predict *E. coli* levels to take precautionary actions if the levels exceed a certain threshold. The study can be expanded to include other faecal and physio-chemical indicators to improve the accuracy of the models. Further enhancements, can include other machine learning/deep learning algorithms to predict *E. coli* levels.
Keywords: *E. coli*, Elasticnet regression, predictive analysis, MARS, XGBoost.

Acknowledgements

This thesis was conducted within the research project “ClimAqua – Modelling climate change impacts on microbial risks for a safe and sustainable drinking water system” grant number 2017-01413 funded by Formas – the Swedish Research Council for Environment, Agricultural Sciences and Spatial Planning. I would like to thank my examiner Mia Bondelind and supervisor Ekaterina Sokolova. This project would not have been possible without the support and guidance you have provided throughout my thesis work. I would extend my thanks to Göteborg Kretslopp och Vatten and Swedish Meteorological and Hydrological Institute (SMHI) for providing access to the data needed for the analysis. I would like to thank Oscar Ivarsson, from Data Science Research Engineers, for guiding and helping me throughout the project. I appreciate that you always been a message away for interesting discussions and feedback. A final thank you to all my friends, teachers and fellow students, it was a pleasure to work with you all.

Ramachandran Ravishankar, Gothenburg, June 2021

Contents

List of Figures

List of Tables

1
Introduction

Water is a natural resource available to humans and is a necessity for life. Access to clean water plays an important role in deciding the economic status of a nation. Water quality is a significant aspect to consider for drinking as well as recreational purposes. Increased levels of faecal contamination reduces the quality of water. Faecal pathogens mainly from human excretion can cause stomach infections and respiratory problems (WHO, 2020). This calls for regular monitoring of faecal contamination in water both for recreation and drinking purposes. To aid faecal contamination monitoring, two main indicators were developed, i.e. *Escherichia coli* or *E. coli* and enterococci (Price & Wildeboer, 2017). Lack of efficient indicators from the coliform group combined with systematic testing methods of finding *E. coli* levels were the reason for increased usage of *E. coli* as an indicator of faecal contamination in water (Odonkor & Ampofo, 2013).

Water quality assessment involves detecting, evaluating, and quantifying water quality problems. Assessing water quality is a complex task that requires optimal and efficient extraction of information and monitoring any deviation from normal values. Artificial Intelligence (AI) has helped to address this issue with user-friendly code development, easy integration with technical systems and transferability into insightful solutions (Strobl & Robillard, 2006). AI is an umbrella term that consists of computer algorithms that mimic how a human brain works. It comprises several techniques such as neural networks - where machines learn from observation data, and try to figure out a solution, deep learning - a method of multiple layers of computational models with various levels of abstraction to mimic the human brain, and machine learning - algorithms that learn from the past data and predict a consequent set of values. The advancements of AI in the past decade along with a vast amount of data has helped to solve many problems that conventional numerical methods could not solve (Reis et al., 2019).

1.1 Aim

In the context of achieving the UN Sustainable development goal on clean water availability, this master thesis tries to address water quality problems that arise due to faecal contamination, i.e. *E. coli* levels (UNSDG, 2019). Assessing the levels of *E. coli* in water is useful to prepare technological measures to decrease faecal contamination and ensure a safe water supply for drinking and recreational purposes. To analyse and predict the *E. coli* levels, the following approaches were

implemented:

- To collect, prepare and process raw data into a dataset with physio-chemical and faecal indicators as an input variable and *E. coli* as the target variable.
- Perform an exploratory data analysis to examine the trends, relationships, outliers and statistical significance of features.
- Incorporate supervised machine learning algorithms to predict *E. coli* levels at the Lärjeholm water intake facility.
- Evaluate the performance and accuracy of prediction using error metrics.

1.2 Hypotheses

This study was based on reasoning the hypotheses given below:

- Supervised machine learning algorithms are expected to predict *E. coli* levels significantly better with a larger dataset (with many indicators as input variables than on a smaller dataset (with limited number of indicators as input variables).
- The performance of the non-linear model supervised machine learning model (XGBoost) is better compared in predicting *E coli* levels at Lärjeholm, than that of linear supervised machine learning models (MARS and Elasticnet regression).

2
Theory

2.1 *E. coli*

Escherichia coli often termed as *E. coli* are rod-shaped bacteria from the Enterobacteriaceae family. Most *E. coli* strains are harmless and are found in the small intestine of humans and warm-blooded animals. However, strains like (0157:H7) or EHEC can cause illness to humans such as diarrhoea, stomach cramps, vomiting and fever (*E. coli*, 2021). Due to its abundant nature in faeces of humans and animals, *E. coli* is termed as an efficient faecal indicator in water samples. The presence of *E. coli* in water samples often indicate recent faecal contamination, i.e. presence of additional faecal pathogens that are harmful to human beings (Edberg et al., 2000). The survival of *E. coli* depends on the water temperature, light intensity and salinity.

2.2 Supervised machine learning algorithms

A machine learning model is an algorithm that automatically analyses the given sample of data, to make predictions based on the characteristics of input dataset. According to Mitchell et al. (1997):

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

Supervised machine learning algorithms operate on labelled dataset that contains independent and dependent variables. The independent variables are the input to the process or phenomena under examination, while the dependent variable, also known as the target variable is to be predicted by the algorithm (Kotsiantis et al., 2007). The algorithm starts by learning the traits of the independent variables and predicts the outcome or the target variable. This is repeated for a set of iterations, where the target variable acts as a teacher or 'supervisor' to the machine learning model to increase its predictive accuracy and rectify its mistake (error) in each iteration that it learns on the dataset (Cox et al., 2020).

Linear models are the simplest form of supervised machine learning algorithms. Linear models are easy to train and fairly straightforward to implement in predictive modelling problems (Gross, 2021). However, linear models are not always accurate

in predicting the target variable when higher dimensions are involved, i.e. when the dataset has a large number of independent variables (Gross, 2021).

Tree-based models are used to address the drawback of linear models. To produce predictions or choices from one or more trees, tree-based models employ a sequence of 'if-then' criteria (Katie, 2021). A single decision tree works based on conditional splitting. The tree splits into several branches and in each split, a condition is checked. The sum of all the output at the final split provides the prediction of the decision tree. Decision trees can handle complex, non-linear relationships in the dataset (Clark & Pregibon, 2017). However, the main drawback of decision trees was that they are slow to train since trees are built sequentially.

2.2.1 Multivariate adaptive regression splines

A drawback of linear algorithms, is that they become unstable and sensitive to complex dataset. To solve this issue, linear models need to be tuned or modified to work with complex relationship in the dataset when the number of features increases. One such modified linear algorithm is called multivariate adaptive regression splines (MARS). MARS works by automatically searching for interaction and relationship with non-linear data (Pyearth, 2021). Each term in this model is a product of 'hinge functions' (Friedman, 1991). A hinge function is divided into two:

- Right function provides the value of the data point as its output when the argument is greater than zero and zero else where (Brownlee, 2021).

$$f(z) = z \ \ (when\ z > 0) \tag{2.1}$$

$$f(z) = 0 \ \ (when\ z < 0) \tag{2.2}$$

- Left function provides the value of the data point as its output when the argument is lesser than zero and zero else where (Brownlee, 2021).

$$f(z) = z \ \ (when\ z < 0) \tag{2.3}$$

$$f(z) = 0 \ \ (when\ z > 0) \tag{2.4}$$

where f(z) is the hinge function fitted to the data points. These functions are also called splines, from which the name of the algorithm has been derived (Brownlee, 2021). Hinge function is similar to rectified linear function in neural networks. MARS generates many of these functions for the data points in the independent variable. A linear model is then fit to the output of every hinge function and the target variable (Brownlee, 2021). Prediction is made by aggregating the weighted output of all the hinge functions in the model (Brownlee, 2021).

2.2.2 Elasticnet regression

A major problem with linear regression is that it becomes unstable when the dataset has large number of features. With the increase in dimension to deal with, the

model becomes complex and prone to large errors. This is when regularization plays an important role to reduce model complexity and penalize features with high coefficients (Zou & Hastie, 2005). Consider a dataset with n number of data points. The loss function to be reduced is given by:

$$loss = \sum_{i=0}^{n} (y_i - \bar{y})^2 \tag{2.5}$$

where y_i is the actual value at i_{th} instant, $\bar{y}$ is the predicted value at i_{th} instant, and $\sum(y_i - \bar{y})^2$ is the summation of errors for all the data points to be reduced.

Elasticnet works by penalizing higher coefficients. The loss function given in equation (2.5) is modified to include additional cost functions that penalize features with high coefficients. Elasticnet combines two penalty functions (i.e., *l1* and *l2* penalty function) to minimize the coefficients penalty functions (Sklearn, 2021a).

l1 penalty reduces the absolute coefficient values of the features, i.e. it reduces the coefficient values and even make some of the coefficients to zero, thereby removing the features from loss function equation (Brownlee, 2020a). Consider a dataset with n number of features, the *l1* penalty function for the features in the dataset is given by:

$$l1 = \sum_{j=0}^{n} |\beta_j| \tag{2.6}$$

where $\sum_{j=0}^{n}$ is the number of features in the dataset (j = 0 to n), β_j is the coefficient for the jth feature to be reduced.

l2 penalty reduces features with high coefficients values, by reducing the sum of squared values of coefficients. This way, *l2* penalty only reduces the high coefficient values without removing them from the model so that they become considerably small or closer to zero (Brownlee, 2020a). Consider a dataset with n number of features, the *l2* penalty function is given by:

$$l2 = \sum_{j=0}^{n} |\beta_j^2| \tag{2.7}$$

where $\sum_{j=0}^{n}$ is the number features in the dataset (j = 0 to n), β_j is the coefficient for j_{th} feature to be reduced.

To improve the accuracy of the Elasticnet regression, it is important to tune the hyperparameters of the model. Section 2.3 provides a brief introduction about hyperparamater tuning, and how it can be helpful to tune an algorithm. The most widely used hyperparameters for Elasticnet regression are *alpha*, and *lambda*. *alpha* provides the weights to each penalty function ($l1_{(penalty)}$ and $l2_{(penalty)}$) i.e. how much each penalty function can contribute to the net loss function, while *lambda* is helpful

to decide the magnitude of the penalty to be added to net loss function (Brownlee, 2020a). Equations 2.8 and 2.9, formulate the use of *alpha* and *lambda*, to include *l1* and *l2* penalty function in the net loss function equation (2.9).

$$elastic\ net\ penalty = (alpha * l1_{(penalty)}) + ((1 - alpha) * l2_{(penalty)}) \tag{2.8}$$

$$elastic\ net\ loss = loss + (lambda * elastic\ net\ penalty) \tag{2.9}$$

2.2.3 Extreme gradient boosting regression

Extreme gradient boosting, often referred as XGBoost, is a popular tree-based non-linear model, that works with an ensemble of decision trees (Chen & Guestrin, 2016). XGBoost works on the principle of gradient boosting algorithm. The technique of the gradient boosting algorithm is to minimize the loss function (actual-predicted) values for each tree split. This is achieved by reducing the slope/gradient of the loss function. The formula given below, shows the loss function to be reduced (mean squared error). y_i is the data point of target variable at i_{th} instant, y_i^p is the prediction at i_{th} instant, $\sum(y_i - y_i^p)^2$ is the total loss function (Chen & Guestrin, 2016).

$$Loss = MSE = \sum (y_i - y_i^p)^2 \tag{2.10}$$

XGBoost works a bit differently from gradient boosting. It takes into account how complex the model is. XGBoost bypasses the stress of finding the loss function for each split in decision tree by using Taylor expansion. Taylor expansion is used to find the first and second derivative of the loss function so that the time complexity is minimized (KDnuggets, 2021). XGBoost also introduces a concept called regularization that helps to specify the threshold of loss function based on a certain split criteria, above which they are penalized. The loss function along with the regularization term is called the objective function. Equation 2.11 shows the general equation of Taylor expansion obtained from KDnuggets (2021) for the objective function to be minimized for extreme gradient boosting .

$$L^t = \sum_{i=1}^{n} (g_i f_t(x_i) + h_i f_t(x_i)) + \Omega f_t \tag{2.11}$$

where,

$$g_i = \frac{dL_{t-1}}{dy_i^{p-1}}$$

$$h_i = \frac{d^2 L_{t-1}}{d^2 y_i^{p-1}}$$

y_i^{p-1} is the predicted value at the previous iteration and L_{t-1} is the loss function at the previous iteration, L^t is the loss function at t^{th} iteration. g_i and h_i are the first and second-order derivative of the loss function of previous iterations, respectively. Ωf_t is the regularization term to reduce the complexity of the model.

2.3 Hyperparameter tuning

Hyperparameters are input that helps to adjust the working of an algorithm according to the dataset considered. Hyperparameter tuning is a labour-intensive approach to find the best parameters that can be given as attributes to the machine learning algorithms. This requires extensive knowledge on the field of research and theoretical background about the algorithm to choose the right hyperparameters to be given as input (Bardenet et al., 2013). Different algorithms can have several input hyperparameters that can be given to tweak the performance of the algorithms since manually finding them would be a tedious process. *GridSearchCV* is a function, that fits the algorithms with a range of values for the input hyperparameters and finds the best value that can be given as the final set of input to the algorithms (Bardenet et al., 2013).

2.4 Performance metrics

The performance of supervised machine learning algorithms is assessed or measured using metrics based on the residuals obtained. This study uses three different error metrics namely mean absolute error, root mean squared error and R^2 score.

2.4.1 Mean absolute error

Mean absolute error (MAE), measures the accuracy of supervised machine learning algorithms. It takes the average sum of the absolute difference between absolute actual and predicted values (Naser & Alavi, 2020). It weighs each error values equally. Ahmed et al. (2019) provide the formula for calculating MAE as given below:

$$MAE = \sum_{i=0}^{n} \frac{|(y_{act} - y_{pred})|}{n} \tag{2.12}$$

where y_{act} refers to the data point at i_{th} instant, y_{pred} refers to the predicted value at i_{th} instant, and n refers to the number of data points in the dataset.

2.4.2 Root mean squared error

Root mean squared error (RMSE), is an alternative to mean squared error (MSE) to reduce the penalizing effect of MSE on larger residuals. RMSE metric scales the

predicted values closer to actual values by taking the square root on MSE (Ahmed et al., 2019). Equation 2.13 provided by Ahmed et al. (2019) show the formula for MSE, where y_{act} refers to the dependent (target) variable at i_{th} instant, y_{pred} refers to the predicted value at i_{th} instant, and n refers to the number data points in the dataset.

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n}(y_{act} - y_{pred})^2}{n}} \tag{2.13}$$

2.4.3 R^2 score

The R^2 score, known as the coefficient of determination, specifies how well the model fits the data (Ahmed et al., 2019). It ranges between 0-1 which indicates the correlation between predicted values and actual values, i.e. a score closer to one indicates a strong correlation between actual and predicted values and vice versa (Naser & Alavi, 2020).

2.4.4 Variation analysis

Variation analysis is used to find the improvement or deterioration in the model when new features are added to the dataset. Given the error metrics on the baseline dataset, the relative change in error (Δ *error*) report the change in model's performance when it is trained on the upgraded dataset (with more features) with the respect to the error on baseline dataset (with limited features). Relative change is given by:

$$\Delta\ error = \frac{G - H}{G} \tag{2.14}$$

where G and H are the error metric values of the model on the baseline and complex dataset, respectively. Δ error is the relative change in error value of the model from baseline to complex dataset.

3

Literature Review

This section deals with the research and study of scientific articles that have previously reported on the topic of water quality and bacteria predictions. The literature review provides a pivotal understanding of the research done on the topic, providing a strong foundation of knowledge that can be used for the methods used in this study. Several studies were performed on water quality predictions using machine/deep learning algorithms or statistical models for the analysis (Tornevi et al., 2014; Solanki et al., 2015). Sources of water such as rivers and lakes from different parts of the world were analysed and predictions were drawn based on faecal indicators such as *E. coli*, enterococci, total coliforms, and also physico-chemical factors such as pH, dissolved oxygen, microbial growth, nutrients, temperature, salinity, etc,. Most of the research focused on predicting either one or several of these indicators to assess water quality (Ahmed et al., 2019; Edberg et al., 2000).

The presence of certain pathogenic microorganisms in drinking water may be harmful to human wellbeing. To avoid the presence of harmful pathogens, it is necessary to evaluate and assess water quality to ensure safe and sustainable drinking water provisions. To do so, certain indicators were used to assess water quality effectively. For freshwater, *E. coli* is widely considered as an effective faecal indicator to measure the level of contamination in water (Odonkor & Ampofo, 2013). *E. coli* was found to be a better predictor of gastrointestinal illness than enterococci and other bacteria indicators in over 900 trials (Avila et al., 2018). An increase in *E. coli* levels in water means there is a risk of contamination from other pathogens since *E. coli* is present in the faeces of all mammals and birds and is commonly seen in contaminated water (Winfield & Groisman, 2003; Edberg et al., 2000).

To ensure safe drinking water provisions, World Health Organisation (WHO) has provided water quality standards to ensure proper monitoring of water quality with a minimum of one monthly sample taken and the results should be disclosed to the public (WHO, 2020). Water monitoring and treatment are based on laboratory analysis of water samples to detect the presence of faecal contamination. However, time constraints are imposed by the fact that water samples need to be collected and processed which requires 18 - 24 hours to obtain the results (Eleria & Vogel, 2005). The requirement of expensive tools for the analysis raises financial constraints. The introduction of statistical analysis and machine learning models are useful in cutting down cost, and time constraints imposed by expensive lab analysis. Most of the research that uses statistical analysis follows a set of methods such as data collection, cleaning and processing, exploring, applying (statistical/machine/deep learning al-

gorithms) and finally evaluating the performance of these models using different error metrics.

3.1 Previous work

This study comes under the research project "ClimAqua – Modelling climate change impacts on microbial risks for a safe and sustainable drinking water system" funded by Formas – the Swedish Research Council for Environment, Agricultural Sciences and Spatial Planning. The study was conducted in collaboration with the data science research engineers, a group of computer science engineers created by Chalmers and Area of Advance - Information and Communication Technologies.

The purpose of the study was to evaluate the suitability of data-driven models with diverse complexity, from simple linear models like least absolute shrinkage and selection operator (LASSO) regressor to complex models such as random forest on predicting *E. coli* levels. The dataset was prepared by combining indicators available from Göteborg Kretslopp och Vatten and Swedish Meteorological and Hydrological Institute (SMHI). Faecal indicators such as *E. coli* laboratory and, Colifast data at stations Garn and Lärjeholm, were obtained from Göteborg Kretslopp och Vatten. Additionally, physio-chemical indicators such as water temperature (at Lärjeholm), turbidity (at Lärjeholm) were obtained from Göteborg Kretslopp och Vatten. Precipitation (at Vänersborg and Komperöd), and flowrate (at Lilla Edet and Göteborg) were obtained from SMHI. The preprocessing steps were cleaning, manipulating, and altering raw data into an effective dataset that can be used for analysis. The target variable was *E. coli* at Läjerholm.

The first step was to check the observations with NaN (Not a Number) values in the dataset. To reduce missing observations and also to understand the effect of lag values of external predictors, different sets of lag values were defined based on domain knowledge.

The second step was to fit machine learning/statistical models into the dataset. Both univariate and multivariate analyses were conducted in this study. In univariate analysis, time series forecasting was performed to forecast the target variable (*E. coli* levels) along with its lag values and timestamp of observation. Baseline statistical models namely autoregressive integrated moving average (ARIMA), exponential smoothing, variable autoregression (VAR) were used for the forecasting procedures. In the multivariate approach, several external indicators were used for the prediction of *E. coli* levels. Models namely LASSO regression, random forest, and TPOT were used to predict *E. coli* levels. The conclusion drawn from the study, states that the multivariate approach was more accurate in predicting *E. coli* levels than the univariate approach.

3.2 Data preprocessing

Raw data can be inconsistent and noisy which becomes a problem when machine learning models are trained on the data. Data preprocessing helps to have structured and meaningful data by cleaning and transforming raw data into a useful format.

Ahmed et al. (2019) in their study on predicting water quality index (a number between 0-100) at Rawal water intake, Pakistan, use several steps to process the data. The data was cleaned by removing the outliers so that the machine/deep learning models do not overfit the data (the model performs better on training data but predicts poorly on test data). A box plot visualization was used to detect the outliers and to set a threshold value above which the values will be eliminated or replaced.

Normalization is an important step to make sure that the input features were comparable, i.e. are, on a common scale. This was helpful for machine learning models to make unbiased predictions on the target variable. Ahmed et al. (2019) in their study on predicting water quality index uses a technique called q-value normalization to convert the range of features to a normal distribution i.e. (mean = 0 and standard deviation = 1). Mohammed et al. (2018) on predicting total coliforms and *E. coli* levels, perform data normalization using the min-max normalization technique to have a common scale for the data.

Missing values can cause errors or less accurate results from algorithms. Hence one of the vital steps in preprocessing data was to handle missing values. Dheda and Cheng (2020) performed a multivariate analysis on dissolved oxygen prediction using recurrent neural networks (RNN). Data that were not regular in time interval, were removed and the missing values were interpolated. The authors conclude that filling missing values improved the accuracy of machine learning.

From the literature review, it was evident that there seems to be little emphasis on preprocessing steps since most of the studies try to implement algorithms that are capable of handling raw uncleaned data by themselves (Avila et al., 2018; Tornevi et al., 2014; Vijayashanthar et al., 2018). Data normalization, outlier detection, and filling missing values were common preprocessing steps (Ahmed et al., 2019; Dheda & Cheng, 2020; Joslyn, 2018).

3.3 Lag values of features

Several studies on predicting faecal bacteria were based on the effect of lag values of indicators, especially rainfall (Laureano-Rosario et al., 2019; Choi & Bae, 2018; Eleria & Vogel, 2005). Avila et al. (2018) mention that past 24 hr cumulative rainfall and flow rate were useful in predicting *E. coli* levels at Oreti River New Zealand. Tornevi et al. (2014) on predicting *E. coli* levels at Göta river, mention an increase in *E. coli* observed with 2-day lag values of rainfall. Vijayashanthar et

al. (2018) developed a model to predict *E. coli* levels by considering daily, 2-day and 7-day cumulative rainfall along with flowrate. He and He (2008) on predicting faecal bacteria, report that the concentration of faecal bacteria increased one day after rainfall before decreasing to lower levels. The studies mentioned above, use a limited number of features (i.e., only lag values of rainfall and flowrate) to predict faecal bacteria levels. One of the hypotheses framed in this study (section 1.2) is to analyse the accuracy of supervised machine learning models when a limited number of features were involved in predicting *E. coli* levels and to check if adding extra features can improve the accuracy of the models.

3.4 Exploratory data analysis

Supervised machine learning algorithms work on labelled dataset, i.e. a dataset with independent and dependent variables (Cox et al., 2020). The independent variables are the input for the process or phenomena being examined or analysed. Independent variables are also referred as predictors or features. The dependent variable, also termed as target variable is the output of the process or phenomena being analysed. To efficiently predict the target variable, it is important to investigate and examine the relationship between features and the target variable. Exploratory data analysis helps to understand the relationship between features and the target variable so that the right machine learning model can be used (Behrens & Yu, 2003). Exploratory data analysis includes trends, relationships, and statistical distribution of the data which help to summarize the characteristics of the entire dataset.

Dheda and Chenq (2020) utilizes Spearman's correlation matrix for explaining the relationship between features. The authors specify that it was useful to understand and differentiate features that had a positive and negative relationship with the target variable. The authors also mention that descriptive statistics were used to gain insights about the data, i.e. the minimum, maximum, mean, median, 25% (first quantile) and 75% (third quantile). Removing outliers (noise) in the data was helpful to have a structured dataset that can be used to train machine learning algorithms. Using descriptive statistics, the authors were able to remove outliers and clean the data. Box plot analysis was another technique that is used to visualize the distribution of data.

3.5 Train test split

An important step before feeding the data into the machine learning model is to split the data into training and test sets. The training and test data consist of independent and dependent variables. The training data is fed to the model to learn the traits (generalized property without noise) of the data and the corresponding outcome of the target variable. Once the training phase is done, the model is fed with the unknown test data to predict the output of the target variable. There are different ways of splitting the dataset into training and test set. Ahmed et al. (2019) in predicting the water quality index, used cross-validation, a technique that

splits the data into k subsets and iterates over all the subsets, considering k-1 subset as the evaluation split. Laureano-Rosario et al. (2019) on predicting enterococci, mention the use of leave-one-out-cross validation (LOOCV) before feeding the data into the ANN model to prevent overfitting. Another study on predicting *E. coli* levels in cascading dams, mention the importance of splitting the data randomly into a training set (80%) and test set (20%) (Abimbola et al., 2020).

3.6 Machine learning models

Several studies suggest the use of supervised machine learning algorithms for predicting faecal bacteria levels (Joslyn, 2018; Avila et al., 2018; Thoe et al., 2014). Joslyn, (2018) used both support vector machine (SVM) and support vector regression (SVR) along with XGBoost for predicting nine water quality factors. Though all three algorithms showed satisfactory results, the computational speed of XGBoost was faster since it runs an ensemble of decision trees in parallel. Avila et al. (2018) used a wide range of statistical models to predict *E. coli* levels using past observations of *E. coli*, accumulated 48 hr rainfall, and flow rate. The models include Dynamic regression, naive Bayes, random forest, decision trees, and Bayesian network. Tornevi et al. (2014) analysed short term variations in faecal bacteria levels in relation to precipitation, using time series regression and non-linear distributed lag models.

Jiang et al. (2013) on predicting faecal bacteria levels, claim that non-linear models were more efficient and accurate in predicting faecal bacteria levels than linear models. The authors report that linear models could not capture vital information such as variance in data, human interventions like land use which are considered to be non-linear. Furthermore, faecal bacteria concentrations vary depending on different factors like location, sources, and environmental factors and hence a non-linear approach may be more suitable for complex, non-linear relationship (Laureano-Rosario et al., 2019). A wide area of research was held on tapping the non-linear relationship between faecal bacteria and environmental factors (He & He, 2008; Thoe et al., 2014; Avila et al., 2018; Zhang et al., 2018).

Artificial neural networks (ANN) were widely used to predict water quality and faecal indicator bacteria since it can take non-linear relationships between different input variables and assess their relationship with faecal indicator bacteria (Jiang et al., 2013). Another study on predicting *E. coli* levels in Charles river, use models such as multivariate linear regression (MLR), and ANN (Motamarri & Boccelli, 2012). The result from the study suggests that ANN was the most efficient of the two models. However, the efficiency of ANN dropped when the input features were limited or small. Mohammed et al. (2018) on predicting faecal coliforms at Maridalen lake, Oslo, used two algorithms namely ANN and SVM by applying different variations in ANN and SVM models. Three types of ANN namely feed-forward ANN model, cascade-forward ANN model, and layer-recurrent ANN model were used for predicting faecal bacteria levels in the given sample. The results from the study suggest that though SVM and ANN yielded comparable results, ANN was more

efficient in estimating extreme variations of faecal indicator bacteria in raw water.

Another study was conducted to predict total coliform levels by using rainfall and flowrate data as the input (Choi & Bae, 2018). Self-Organising Linear output (SOLO) - ANN was used for predicting total coliform levels in Aliso Creek, California. SOLO-ANN is a parallel running system of classification and mapping layers, as opposed to hidden layers that are common in most ANN. Results from the study suggest that SOLO-ANN was able to reasonably predict total coliforms using rainfall data without additional inputs, however, it performs poorly during the summer season/ no-rain period.

From the literature review, it is seen that AI has been widely used to complement expensive lab analysis for predicting water quality levels. Normalization, filling the missing values and outlier detection and manipulation were the commonly seen data preprocessing techniques. Different models were used to predict faecal bacteria and water quality levels. Commonly used algorithms are linear, non-linear regressors and ANN.

Table 3.1: *Summary of the literature study (DO -dissolved oxygen, DS- dissolved solids, reg- regression, TC -total coliforms, FIB- faecal indicator bacteria, FC- faecal coliforms, OLS- ordinary least squares)*

Author	Target label	Algorithms	Data	Result
Abimbola et al. (2020)	*E. coli*	ANFIS, ANN, PCA, reg	Water temperature, flow rate, rainfall, animal density, grazing patterns	ANFIS-most accurate.
Avila et al. (2018)	*E. coli*	mul.logistic reg, Bayesian network, etc,.	*E.coli* levels - weekly data.	Bayesian network-most accurate.
Choi and Bae (2018)	TC	SOLO, ANN	Rainfall, flowrate	Model performs slightly better if rainfall is included in the data.
Dheda and Cheng (2020)	DO	RNN, LSTM	pH, DO, turbidity and conductivity	Single - step models were more accurate than the multistep models.
Elaria et al. (2019)	FC	log-reg, OLS	Charles river data.	OLS - 50 -60% accuracy.
He and He (2008)	FIB	ANN	Temperature, pH, turbidity	ANN- superior in prediction of *E.coli*
Laureano-Rosario et al. (2019)	Enterococci	ANN	Enterococci, and many more.	Turbidity,cuml.48 hr precipitation, MSL, SST were the most important features.
Mohammed et al. (2017)	*E. coli*, enterococci.	RF regression	pH, conductivity, color, seasons	Color and season were the most important features.
Mohammed et al. (2018)	*E. coli*	ANN, SVM(three solvers)	pH, DO, turbidity, color, conductivity	Raw water turbidity, color and alkalinity- coliform, *E.coli.*

Author	Target label	Algorithms	Data	Result
Motamarri and Boccelli (2012)	FC	LVQ, MLR, ANN	lag of bacteria,rainfall, log10 discharge.	MLR - high false negative rate, LVQ- best classifier
Solanki et al. (2015)	DO	ANN, linear reg	pH, DO, turbidity	ANN - highest prediction capability
Tornevi et al. (2014)	FIB	Time series reg, non-linear distributed lag models.	Turbidity, *E. coli*, precipitation 24h, 48h	Rainfall elevates microbial risks year-round in this river.
Vijayashanthar et al. (2018)	FIB	ANN	2 and 7 day cuml.rainfall, flowrate, turbidity.	86.5 % accuracy - ANN
Zhang et al. (2018)	*E. coli*	NARX, WA-NAR models	3 months of observed data	NARX models -best performance among the three models

4
Methods

This chapter addresses the methods followed in analysis and prediction of *E. coli* levels using water quality features. The following are the questions that are answered in the method section:

- What are the data used for the prediction?
- What are methods of cleaning, preparing and structuring data?
- What are the machine learning methods used and how are they assessed?

4.1 Study area

Göta älv is the longest-running river in the southern part of Sweden. It is also one of the largest rivers around Sweden with an annual discharge of 575 m^3/s ("The Göta älv Estuary", 2021). This study focuses on the part of the river that flows from Vänern down to Kattegat, which is around 93 km. It is estimated to be the drinking water for 700,000 people and around 1000 ships/month travel by the river ("The Göta älv Estuary", 2021). Though the river meets the conditions for normal drinking water standards (after treatment), faecal contamination poses a continuous threat to water quality and stresses the need for continuous water quality monitoring ("The Göta älv Estuary", 2021). Along the river, there are many water quality monitoring stations that continuously examine water properties and check for faecal contamination in Göta älv.

4.2 Dataset and preprocessing

The data for this study was obtained mainly from Göteborg Kretslopp och Vatten and Swedish Meteorological and Hydrological Institute (SMHI). The data was subjected to preprocessing techniques that involve transforming, adjusting, cleaning and preparing the data to fit the machine learning models. Preprocessing steps were performed using Python 3 as the programming language and Anaconda's Jupyter notebook as the development environment (IDE).

The Göteborg Kretslopp och Vatten provides data for faecal bacteria and physio-chemical indicators that are analysed using water samples obtained from the Göta älv. Faecal indicators such as coliforms (colony forming units (CFU/ 100 ml)), *E. coli* (CFU/100 ml), clostridia (number/100 ml), enterococci (number/100 ml), and coliphages (number/100 ml) were available. Physio-chemical indicators such as turbidity (formazin nephelometric unit (FNU)), water temperature (°C), conductivity

(Siemens/meter), and color (milligram/Liter Platinum-Cobalt units) were available. These indicators were measured at different stations: Garn, Södra Nol, Surte, and Lärjeholm (Figure 4.1). *E. coli* at Surte is measured once every two months, while *E. coli* at Södra Nol is measured at irregular intervals, approximately twice a month and hence only faecal indicators from Lärjeholm and Garn were considered in this study.

The data for precipitation (mm) was provided by SMHI. Precipitation data is available at two stations namely Vänersborg (upstream) and Komperöd (middle) (Figure 4.1). However, the timestamp of observations differs between these stations. Precipitation was observed on an hourly basis and are added together to obtain daily data. Figure 4.1 shows the map with water monitoring facilities and weather stations from which the data were collected for Göta älv.

E. coli laboratory data at Lärjeholm

The raw *E. coli* data was processed before it was labelled as target variable. The target variable, *E. coli* had 1290 data points during a time interval of 7 years (2012 to 2019). The data consist of *E. coli* observed on average three times a week. The timestamp of observations was parsed (yyyy-mm-dd) to be used as the index of the dataset. Additionally, duplicates (a single date had two observations) were removed and the column was named as '*E. coli*_LAB_LAE', i.e. *E. coli* laboratory observations at Lärjeholm. From Table 4.1 it was evident that *E. coli* data was skewed heavily on the right, i.e. 75% of the data points were less than 145 CFU/100 ml. Skewed distribution can adversely affect the performance of machine learning models. Log_{10}(x+1) transformation on the data was used to reduce the skewness in the data, by transforming them to normal-like distribution. Tables 4.1 and 4.2 show the statistical inference of the initial dataset and processed dataset, respectively.

Various studies suggest the importance of lag values in predicting *E. coli* levels (Tornevi et al., 2014; Avila et al., 2018). It was realistic to have lag values to predict *E. coli* levels since on given day, the input variables (features) may not be available to predict *E. coli* levels. The reason is that the timestamp of observations for the features were different from each other. Therefore, lag values of features were incorporated to predict *E. coli* levels. A lag-(1) observation represents the last observed *E. coli* level from the current observation, and lag-(2) represents the second to last observation and so on. Up to three lag observations were created for *E. coli* laboratory data and were labelled as '*E. coli*_LAB_LAE_lag1O' (i.e., 1-lag observation for laboratory *E. coli* values at Lärjeholm), '*E. coli*_LAB_LAE_lag2O', '*E. coli*_LAB_LAE_lag3O' were defined. The original data '*E. coli*_LAB_LAE' was labelled as the target variable.

E. coli laboratory data at Garn

E. coli at Garn had 924 data points during time interval of 7 years (2012 to 2019). The column was labelled as '*E. coli*_GA', i.e. *E. coli* at Garn. *E. coli* were observed approximately two times a week. The preprocessing techniques used were similar to that of *E. coli* at Lärjeholm. The original data '*E. coli*_GA' was removed after three lag observations were extracted. The lag values were labelled as '*E. coli*_LAB_GA_lag1O', '*E. coli*_LAB_GA_lag2O', and '*E. coli*_LAB_GA_lag3O'.

Coliforms at Lärjeholm and Garn

Coliforms at Lärjeholm and Garn was processed with the following technique. The columns were labelled as 'coliforms_LAE' and 'coliforms_GA'. Coliforms at Lärjeholm had 1331 data points for a period of 7 years (2012 to 2019). Coliforms at Garn had 937 data points for the same period. Coliform observations at Lärjeholm and Garn were available approximately four times and two times a week, respectively. The statistical inference from Table 4.1 suggests that coliform values were quite high with the maximum value around 20,000 (CFU/100 ml) observed at Lärjeholm and 16,000 (CFU/100 ml) observed at Garn. The features were log_{10}(x+1) transformed to reduce the impact of high values (outliers) on machine learning models. Three lag observations were created and labelled as 'coliforms_LAE_lag1O', 'coliforms_LAE_lag2O', 'coliforms_LAE_lag3O' for Lärjeholm. Similarly, lag observations were created for coliforms at Garn. Once lag values were constructed, 'coliforms_LAE' and 'coliforms_GA' were removed from the dataset.

Turbidity at Lärjeholm

Turbidity observations were available approximately three times a week. The column was labelled as 'turb_LAE'. Turbidity had 1246 data points for a time period of 7 years (2012 to 2019). Preprocessing steps used for turbidity were similar to *E. coli* data. Three lag observations were extracted similar to *E. coli* observations and were labelled as 'turb_LAE_lag1D', 'turb_LAE_lag2D', 'turb_LAE_lag3D', respectively. Once lag values were constructed, 'turb_LAE' was removed from the dataset.

Water temperature at Lärjeholm

Water temperature at Lärjeholm had 1400 data points for a time period of 7 years (2012 to 2019). Preprocessing techniques were similar to *E. coli* data except for log_{10}(x+1) transformation. A reason to avoid log_{10}(x+1) transformed was that, most of the data points were found to be very low, i.e. 75% of the values were found to be less than 9 °C and taking logarithmic transformation might reduce the values more which makes it difficult for machine learning algorithms to learn. 1-day lag observation was extracted and labelled as 'waterTemp_LAE_lag1D' and the original data 'waterTemp_LAE' was removed from the dataset.

Precipitation at Vänersborg and Komperöd

Precipitation at Komperöd and Vänersborg were processed in similar ways as other indicators. Precipitation at Komperöd and Vänersborg, had 2890 data points each for a time period of 7 years (2012 to 2019). The features were labelled as 'precipitation_KR' and 'precipitation_VB'. Similar to water temperature, precipitation values were not log_{10}(x+1) transformed. Four lag observations were extracted and labelled as precipitation_VB_lag1D, precipitation_VB_lag2D, precipitation_VB_lag3D, precipitation_VB_lag4D and a similar procedure was done for precipitation_KR. Once the lag observations were extracted, the original data namely precipitation_VB, precipitation_KR were removed from the dataset.

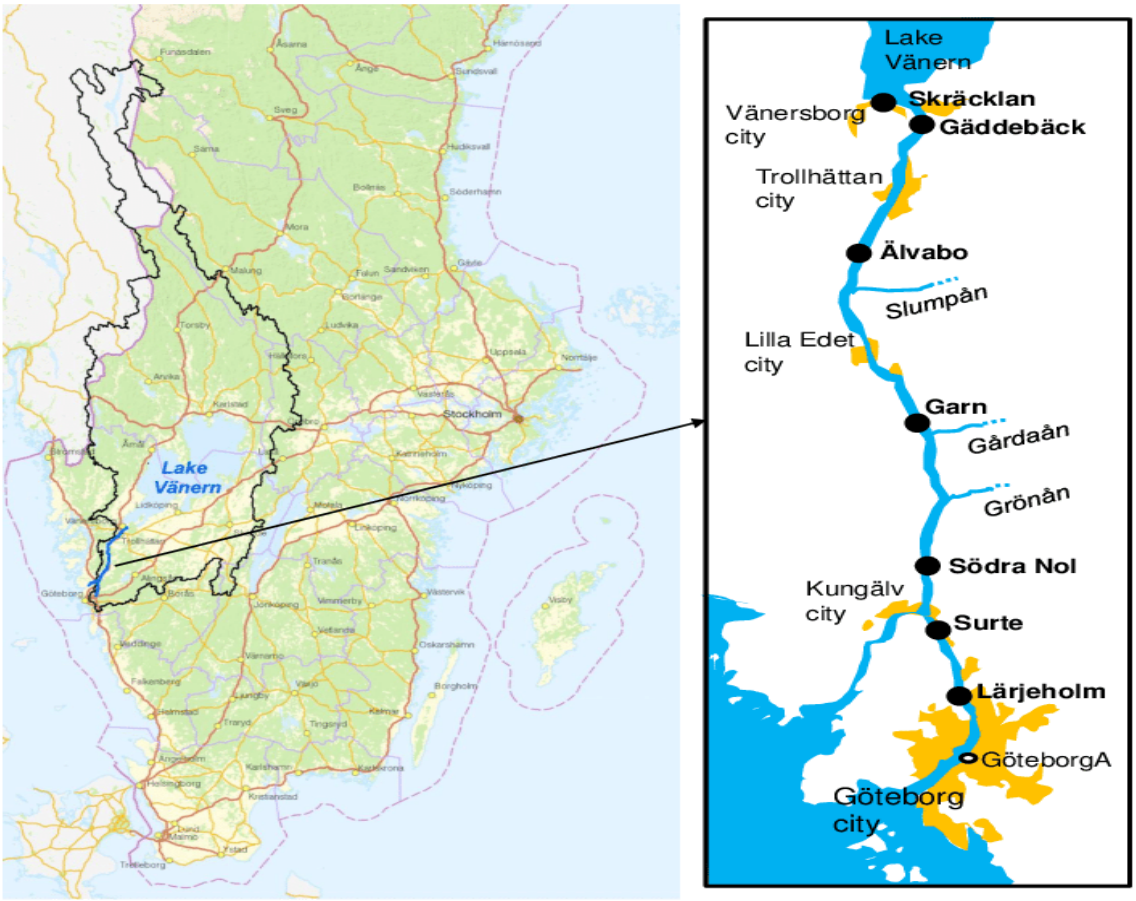


Figure 4.1: *The seven water treatment facilities of Göta älv (right panel) taken from a study by Göransson et al. (2013), licensed under CC BY 3.0. Background map (left panel) ©Lantmateriet, taken from SMHI shows the section of Göta älv catchment area from Vänern lake to Kattegatt sea.*

4.3 Exploratory data analysis

Exploratory data analysis (EDA) was used to find insights about the data. The features evaluated in EDA were *E. coli*, water temperature and turbidity at Lärjeholm and precipitation at Komperöd. Other features namely coliform at Lärjeholm and Garn, *E. coli* at Garn, precipitation at Vänersborg, are provided in Appendix A (Figures A.1 to A.6).

4.3.1 Scatter plot

Many studies suggest that precipitation with prior 2 day observations had greater effect on *E. coli* levels than other physio-chemical indicators (Avila et al., 2018; Tornevi et al., 2014). To understand the effect of precipitation and its lag values on features, scatter plots were constructed using the matplotlib library available in python. The x-axis represents the precipitation at Komperöd and its 2-day lag observations and the y-axis represents different features mentioned above.

4.3.2 Descriptive statistics

Descriptive statistics provides a quantitative summary of information about the observations under consideration. The descriptive statistics were constructed using the *'describe'* method available in the pandas library (Pandas, 2021). This helps to analyse the data about statistical aspects such as central tendency, variance (standard deviation). Measurement of central tendency includes mean, median, and mode, while standard deviation was used to measure the variability in the feature. To have a balanced dataset, the number of data points in all the features were made equal to *E. coli* levels. Tables 4.1 and 4.2 show the descriptive statistics of initial dataset (without processing) and processed dataset. The tables do not inform about lag values since these exhibits similar information as the initial dataset.

Box plot analysis was used to visually analyse descriptive statistics and detect outliers in the dataset (Ahmed et al., 2019). Boxplot was constructed using the *seaborn* library available in python.

4.3.3 Correlation matrix

Correlation is the measure of association between any two variables. Spearman's correlation matrix was used in this study to analyse the degree of association between two variables. The entire dataset was used to construct the correlation matrix. Variables can have a positive correlation meaning one can positively influence another and a negative correlation means vice versa. A neutral correlation represents no association between the variables. Spearman's correlation matrix was constructed using the *'corr_'* function available in the scikit library in python (Correlation, 2021).

4.3.4 Missing number matrix

The missing number matrix was used to visually analyse the number of missing values in each feature. In this way, before starting our filling method, it was useful to know if it is necessary to fill the data, drop the values or if the number of missing values were negligible. Missing observations were visualized using the *'missingno'* method available in python (Scikit, 2021b). Figure 4.2 below shows the missing values in the dataset.

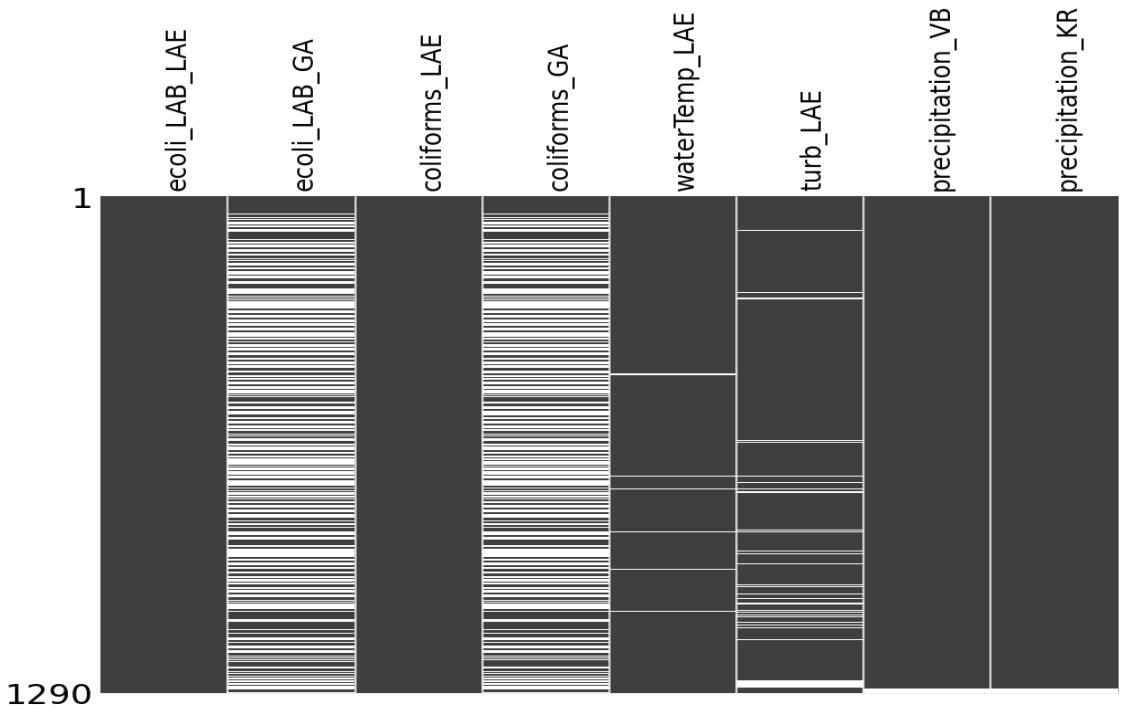


Figure 4.2: *Missing number matrix. The white lines indicate the number of missing observations.*

From Figure 4.2, it was evident that *E. coli* and coliforms at Garn had a lot of missing values, with few missing observations of the water temperature and turbidity at Lärjeholm. The missing values for faecal bacteria, water temperature and turbidity were filled using the *'fillna'* method available in python that helps to fill the missing values with preceding observations (Avila et al., 2018; Joslyn, 2018). This way it was realistic to have a dataset that does not change in its distribution when the missing values were filled. The missing values in precipitation were filled by the mean of the feature because most of the precipitation values were zero and hence filling it with the mean will not interfere in the statistical distribution of the data (Muharemi et al., 2019). Table 4.2 shows the dataset after applying data preprocessing techniques.

Table 4.1: *Descriptive statistics on the initial data collected from Göteborg Kretslopp och Vatten and SMHI.*

Feature	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
E. coli LAE (/100 ml)	1290	145	167	0	41	86	170	1300
E. coli GA (/100 ml)	924	163	204	0	41	86	190	1600
Coliforms LAE (/100 ml)	1331	935	1500	12	260	440	960	20000
Coliforms GA (/100 ml)	937	1024	1745	4	250	470	960	16000
Water Temperature (°C)	1400	9.45	6.33	-0.1	3.5	8.9	15	23.7
Turbidity (FNU)	1246	7.59	4.92	1.9	4.7	6.3	8.6	49
Precipitation_VB (mm)	2890	2.23	4.51	0	0	0	2.40	53.3
Precipitation_KR (mm)	2890	3.04	5.77	0	0	0	3.68	62.1

Table 4.2: *Descriptive statistics of the initial dataset after applying data preprocessing techniques (i.e., $log_{10}(x+1)$ transformation, removing duplicates, and filling missing values).*

Feature	**count**	**mean**	**std**	**min**	**25%**	**50%**	**75%**	**max**
E. coli LAE (log_{10}(x+1) /100 ml)	1290	1.93	0.48	0	1.62	1.94	2.23	3.11
E. coli GA (log_{10}(x+1)/100 ml)	1290	1.96	0.45	0.3	1.67	1.97	2.27	3.2
Coliforms LAE (log_{10}(x+1)/100 ml))	1290	2.73	0.44	1.11	2.42	2.66	3	4.3
Coliforms GA (log_{10}(x+1)/100 ml)	1290	2.73	0.44	1.32	2.43	2.70	3	4.2
Water Temperature (°C)	1290	9.19	6.33	-0.1	3.40	8.30	15.5	23.7
Turbidity (log_{10}(x+1) FNU)	1290	0.89	0.19	0.46	0.76	0.86	0.99	1.70
Precipitation_VB (mm)	1290	2.31	4.51	0	0	0	2.60	53.3
Precipitation_KR (mm)	1290	3.17	6.09	0	0	0	3.8	62.1

4.4 Baseline and complex datasets

Statistical significance tests are tools that can be used to ascertain the effect of independent variables on the target variable (*E. coli*_LAB_LAE). One such tool used in this study is called the A/B testing method. A/B testing is a randomized control experiment, where two versions of a dataset are compared to find out which performs better in a controlled environment (Saxena, 2020).

In the context of above scenario, the dataset after preprocessing (Table 4.2) was divided into two; baseline dataset and complex dataset. The baseline dataset had lag observations of two physio-chemical indicators namely precipitation (at Komperöd) and water temperature (at Lärjeholm), along with the target variable *E. coli* at Lärjeholm. The complex dataset was an upgraded version of the baseline dataset with lag values of additional indicators: faecal indicators - *E. coli* and coliforms (at Garn and Lärjeholm) and also a physio-chemical indicator - turbidity at Lärjeholm. Supervised machine learning models were run on both the dataset and the performance of the algorithms were evaluated using error metrics. Based on the results, conclusions were drawn on the efficiency of adding features to the baseline dataset.

- **Baseline Dataset**: Upto 4-lag observations of precipitation (at Vänersberg and Komperöd) and 1-lag observation of water temperature (at Lärjeholm) as features and *E. coli* values at Lärjeholm as the target variable.
- **Complex Dataset**: Upto 3-lag observations of *E. coli* (at Garn and Lärjeholm), and coliforms (at Garn and Lärjeholm). 1-lag observations of water temperature (at Lärjeholm), 1-3 lag observations of turbidity (at Lärjeholm), 1-4 lag observations of precipitation (at Vänersberg and Komperöd) as features and *E. coli* values at Lärjeholm as the target variable.

4.5 Train, test split

The final step before feeding the datasets to the machine learning algorithms was to split them into train and test data. Algorithms use training data to learn the general characteristics of the features and the test data to evaluate the model on how well it has trained. Both baseline and complex datasets were split into training and test data. The spit was performed using the sklearn library's *train_test_split* method available in python (Sklearn, 2021b). Majority of the data was taken to train the machine learning algorithm (75%) and the rest of the data was kept for testing the models (25%). This corresponds to 967 data points in the training data and 323 data points in the test set.

4.5.1 Cross validation

Cross-validation was used to reduce overfitting, by making the model learn the generalized property of the training data rather than to learn all the traits (outliers, noise) of the training data. Training data was further split into two, the training and evaluation data, where the model was trained on k split and evaluated on k-1 split. The evaluation split was replaced from the training data for n iterations so that the model learns the generalized property of the data instead of overfitting. *'cross validation'* function, available in scikit library was used to split the training data into 6 splits (Scikit, 2021a). The training data was taken to be 5 splits and the last split was the evaluation split for 10 iterations (Ahmed et al., 2019).

Additionally, the hyperparameters of the algorithms were tuned using *GridSearchCV*. *GridSearchCV* is available from scikit library of python. Table 4.3 shows the summary of input to *GridSearchCV* and the best hyperparameters that were selected.

4.6 Supervised machine learning algorithms

Laureano-Rosario et al. (2019) state that non-linear models were more accurate in predicting faecal bacteria levels than linear models. In context of the above statement, two linear models (MARS and Elasticnet regression) and a non-linear

tree-based model (XGBoost regression) were used in this study to predict faecal bacteria levels at Göta älv (Lärjeholm water intake facility).

4.6.1 Multivariate adaptive regression splines (MARS)

MARS is a linear model available from the *'PyEarth'* library (Brownlee, 2021). Splines refer to the hinge functions that are similar to rectified linear activation functions in neural networks (Pyearth, 2021). MARS fits a number of these functions to the data points and the predictions are given as output of these functions. The MARS model was fit to the training data without any hyperparameters since the model was capable of selecting the best hyperparameter setting by itself (Brownlee, 2021). Overfitting was reduced by cross-validation.

4.6.2 Elasticnet regression

The Elasticnet regressor was constructed using the *'scikit.linear_model'* library available in python (Sklearn, 2021a). The Elasticnet regression combines the advantages of both *l1* and *l2* regularization to penalize coefficients of highly correlated features as mentioned in the theory chapter (Brownlee, 2020a). The final loss function incorporates both *l1* and *l2* penalty functions. Two hyperparameters, i.e. *'alpha'* and *'l1_ratio'* were used to control the magnitude and the contribution of *l1* and *l2* penalty functions to the final loss function. A range of values for *'alpha'* and *'l1_ratio'* was provided based on the official document provided by sklearn (Sklearn, 2021a). Using *GridSearchCV*, the best value for the hyperparameter was picked up from the range of values (by iterating through all the values, and selecting ones that result in the lowest error possible). The hyperparameters with the chosen values, were fit to the training data. Table 4.3 provides the range of values and the best value for the hyperparameters used in the model. The feature coefficients plot examines the absolute value of coefficients for each feature. The feature coefficient plot was constructed using *'coef_'* attribute of Elasticnet regressor.

The hyperparameters used in Elasticnet (*GridSearchCV*) are as follows:

- ***alpha*** - Regulates the effect of the total regularization factor (*l1* and *l2* penalty function) to the loss function.
- ***l1-ratio*** - Provides the weights to *l1* and *l2* penalty functions.

4.6.3 Extreme gradient boosting regression (XGBoost)

Extreme gradient boosting regression, also known as XGBoost regression belongs to the ensemble method, built with a culmination of decision trees to make the predictions (Joslyn, 2018). XGBoost regressor is constructed using the *'ensemble_methods'* library of python. The hyperparameters *'n_threads'*, *'learning _rate'*, *'max_depth'* and *'max_feature'* were used as input to *GridSearchCV* (Chen & Guestrin, 2016). A range of values were used for the hyperparameters and the best values for each hyperparameter was chosen using *GridSearchCV* (*XGBoost Parameters*, 2021). Table 4.3 shows the hyperparameters used and the best values for the hyperparameters by *GridSearchCV*. Using this result, a final XGBoost model

was built and trained on the data. XGBoost provides the importance of each feature in predicting the target variable using the attribute called *'_feature_importances_'* available in python (Brownlee, 2020b).

The hyperparameters used in XGBoost (*GridSearchCV*) are as follows:

- ***n_estimators*** - Number of decision trees that XGBoost to construct
- ***n_threads*** - Parallel processing and the number of cores needed to be activated
- ***objective*** - The loss function that needs to be minimized
- ***max_depth*** - The maximum depth of the tree
- ***max_features***- The maximum number of features that needs to be included in each split

Table 4.3: *Hyperparameter tuning for Elasticnet and XGBoost regression using GridSearchCV.*

Algorithms	**Hyperparameters evaluated**	**Final values**
Elasticnet	*alpha* = [1e-5,1e-3, 1e-1,1e+1], *l1-ratio* = (0.1, 1, 0.01)	*alpha* = 10.0, *l1-ratio* = 0.1
XGBoost	*n_estimators* = 200, *n_threads* = 4, *objective* = ['reg:squarederror'], *max_depth* = [5,6,7], max_features = [1,2]	n_estimators = 200, n_threads = 4, *objective* = 'reg:squarederror', *max_depth* = 5, *max_features* = 1

4.7 Performance metrics

The accuracy of the algorithms was evaluated using error metrics. The error metrics used were mean absolute error (MAE), root mean squared error (RMSE) and R^2 score (coefficient of determination). The error metrics were available in the *'scikit.error metrics'* library of python. Once the error metrics were calculated, a simple variation analysis was conducted to examine the relative change in error metrics, when the models were trained on the complex dataset. Relative change in error metrics is the improvement rate in errors when the models were trained on complex dataset with respect to the error values obtained in baseline dataset. Relative change in error metrics was calculated for all three error metrics mentioned above.

5
Results

This section deals with the results obtained in the study. The results were summarized into sections that cover aspects of exploratory data analysis, and the performance of supervised machine learning models.

5.1 Exploratory data analysis

5.1.1 Correlation matrix

Correlation is the measure of association between any two variables. Spearman's correlation matrix was constructed using the initial dataset (Figure 5.1). The correlation matrix represents the relationship between features in the form of a heat map by colour grading and numerical values. The colour-grading provides a visual representation of the relationship between different features and the target variable. The colour-grading goes from dark brown to dark green which shows the negative and positive relationship between variables, respectively. The numbers inside the cell shows the absolute correlation coefficients between the variables, ranging between -1 to +1 which provides the same information.

From Figure 5.1, it can be seen that turbidity at Lärjeholm, coliforms at Läjerholm and Garn, *E. coli* at Garn are positively correlated with *E. coli* at Lärjeholm, while water temperature at Läjerholm are negatively correlated with *E. coli* at Lärjeholm. It is known that *E. coli* decays slower at lower temperature than at higher temperature and hence, inversely proportional to higher water temperature (Avila et al., 2018). Instantaneous values of precipitation at Vänersborg and Komperöd are negatively correlated with *E. coli* at Läjerholm.

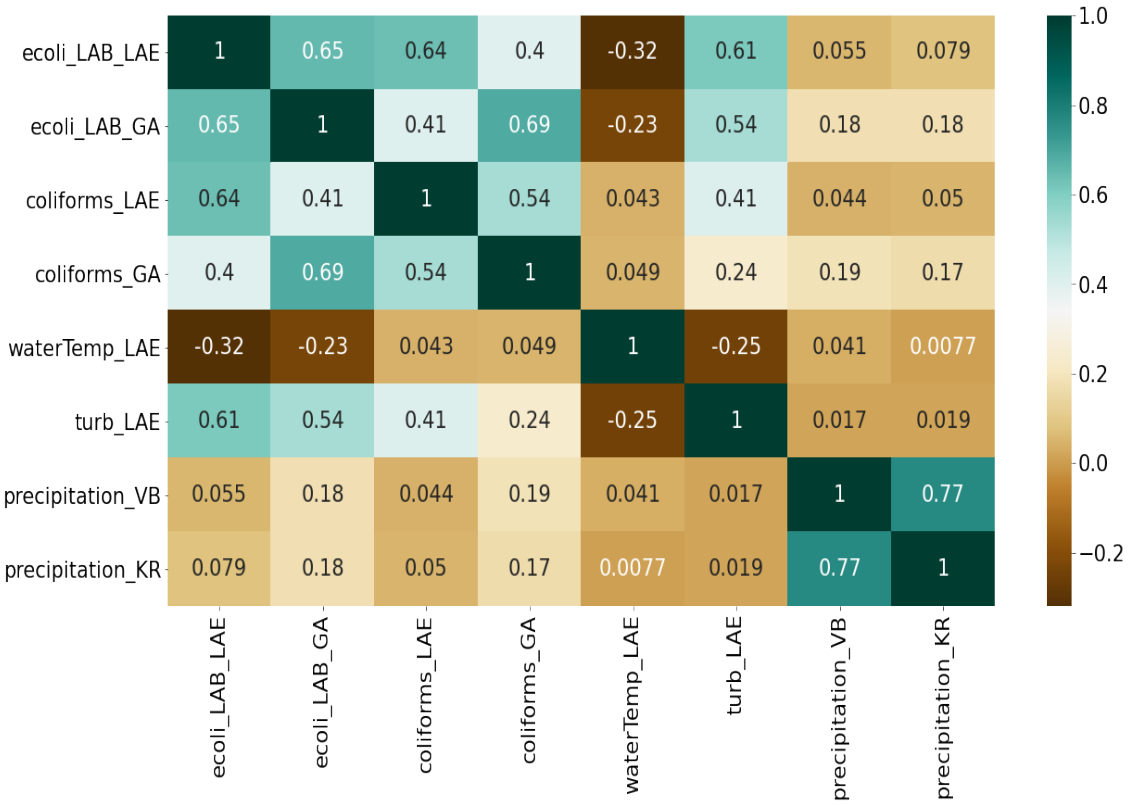


Figure 5.1: *Spearman's correlation matrix. The matrix was built with the features from the initial dataset. The color grading and the absolute coefficient values in the matrix represent the relationship between variables.*

5.1.2 Visualization plots

A visual analysis of the dataset was helpful to examine and check for variability, trends, distribution and correlation between different variables. Two different visualization methods were used i.e. scatter plots and box plots, to examine the data on its statistical aspects. However, only the following features were used in this section:

- *E. coli* laboratory data at Läjerholm
- Water temperature at Läjerholm
- Turbidity at Läjerholm
- Precipitation at Komperöd

Other features namely *E.coli* at Garn, coliforms at Garn and Lärjeholm, precipitation at Vänersborg, were similar in their statistical inference and hence are provided in the Appendix A (Figures A.1 to A.6).

Scatter plot analysis

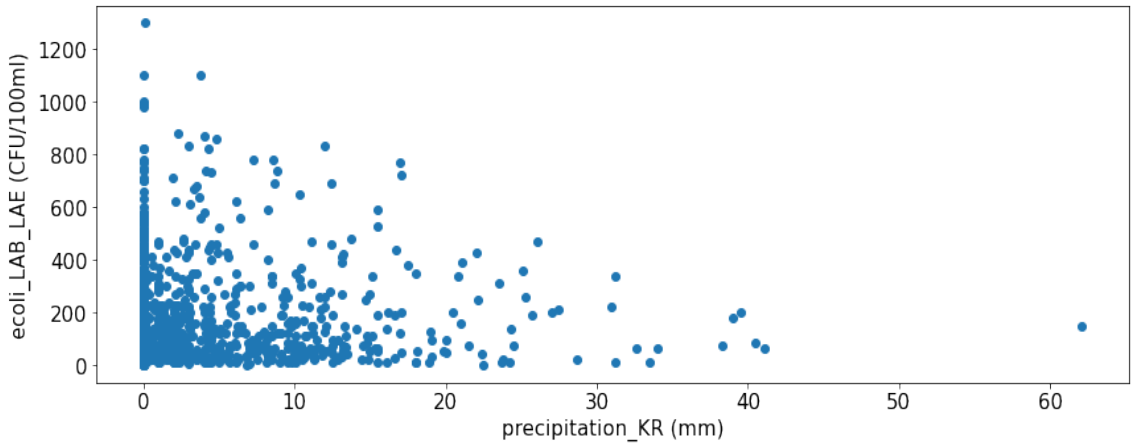
Tornevi et al. (2014) conclude that precipitation with 2-day lag values affect faecal bacteria levels the most, when compared to any other physio-chemical indicator in Göta älv. Scatter plots were constructed to analyse the effect of precipitation and

its 2-day prior values on faecal indicator bacteria and also on physio-chemical indicators such as turbidity and water temperature.

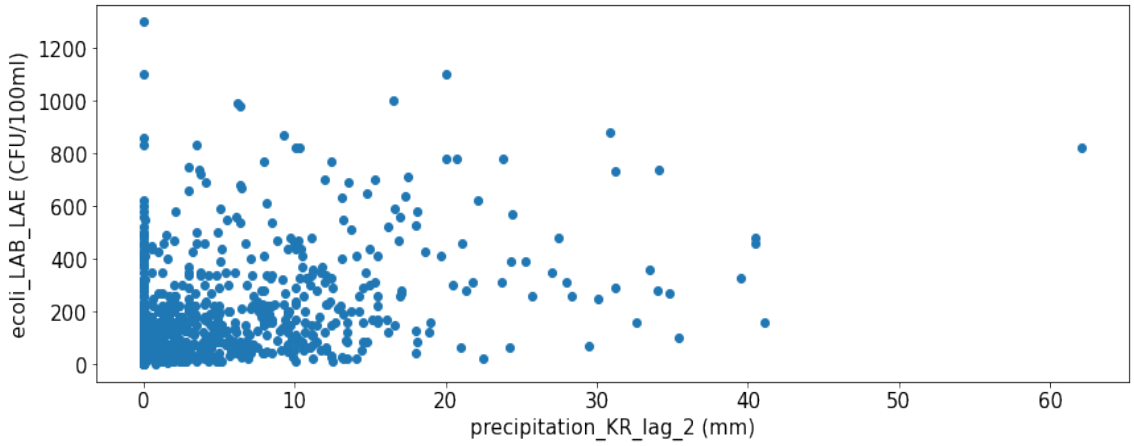
The scatter plots in Figure 5.2 ((a) and (b)) show the relation between observations of *E. coli* levels at Lärjeholm and precipitation and 2-day lag observations of precipitation at Komperöd. Note that the trend of *E. coli* tends to gradually decrease with an increase in precipitation at Komperöd (Figure 5.2 (a)). *E. coli* levels tend to be more scattered in the upward direction, when 2-day lag values of precipitation is used compared to precipitation on the same day (Figure 5.2 (b)). The reason for elevated *E. coli* levels could be due to the fact that stormwater runoff can carry animal/human sewage which would cause an increase in *E. coli* concentration at Lärjeholm. Additionally, rain can cause soil erosion and higher turbidity (poor water clarity) that can increase *E. coli* levels (Haramoto et al., 2006). Similar relationship was observed between other faecal bacteria indicators (i.e., *E. coli* at Garn, coliforms at Lärjeholm and Garn) and 2-day lag values of precipitation at Komperöd (appendix A, Figures A.1 to A.3).

Lag of precipitation (at Komperöd) do not affect water temperature (at Lärjeholm). Figure A.4 in the Appendix A, shows the relationship between water temperature and precipitation. The trend was fairly constant with a gradual decrease in water temperature levels at precipitation greater than 40 mm (Appendix A, Figure A.4 (a)). However, lag values of precipitation at Komperöd (Appendix A, Figure A.4 (b)) do not seem to significantly affect water temperature at Lärjeholm.

Lag values of precipitation greatly affect turbidity (Figure 5.3 (a) and (b)) (Göransson et al., 2013). Turbidity at Lärjeholm, on a given day, gradually decreased with elevated precipitation on that day (Figure 5.3 (a)) while, turbidity values were scattered upwards with 2-day lag values of precipitation (Figure 5.3 (b)). The reason is that, with heavy rainfall, particles from the soil washes into the river. This would cause the river bed sediment to re-suspend along the water surface causing an increase in turbidity levels (Sinton et al., 2002).

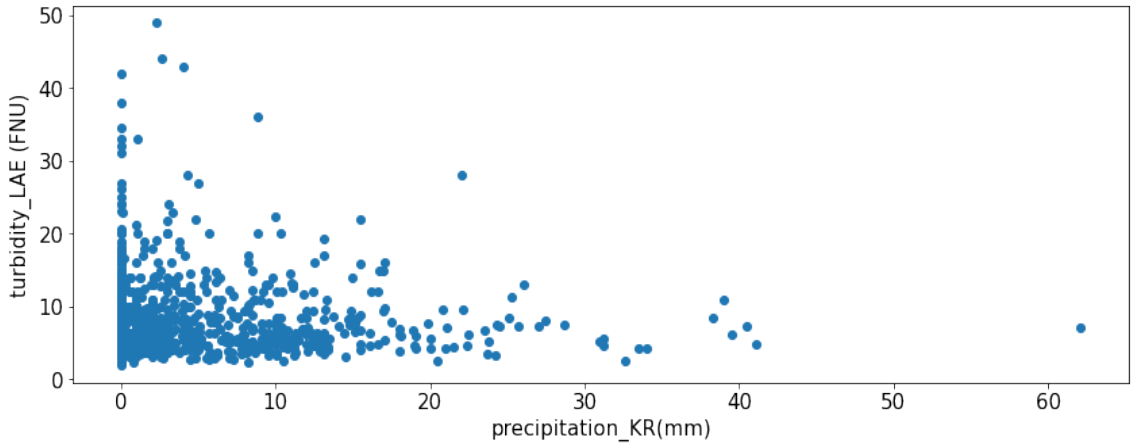


(a)

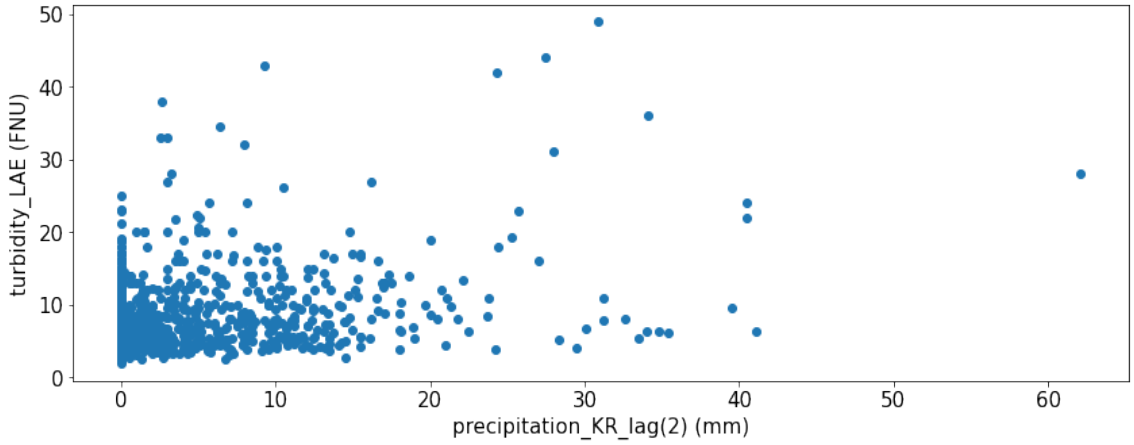


(b)

Figure 5.2: *Relationship of precipitation and its 2-day lag values with different features. Figure (a) represents precipitation at Komperöd with E. coli levels at Lärjeholm, Figure (b) represents 2-day lag observations of precipitation at Komperöd with E. coli at Lärjeholm.*



(a)



(b)

Figure 5.3: *Figure (a) represents relationship between precipitation at Komperöd with turbidity at Lärjeholm and Figure (b) represents the relationship between 2-day lag observations of precipitation at komperöd with turbidity at Lärjeholm.*

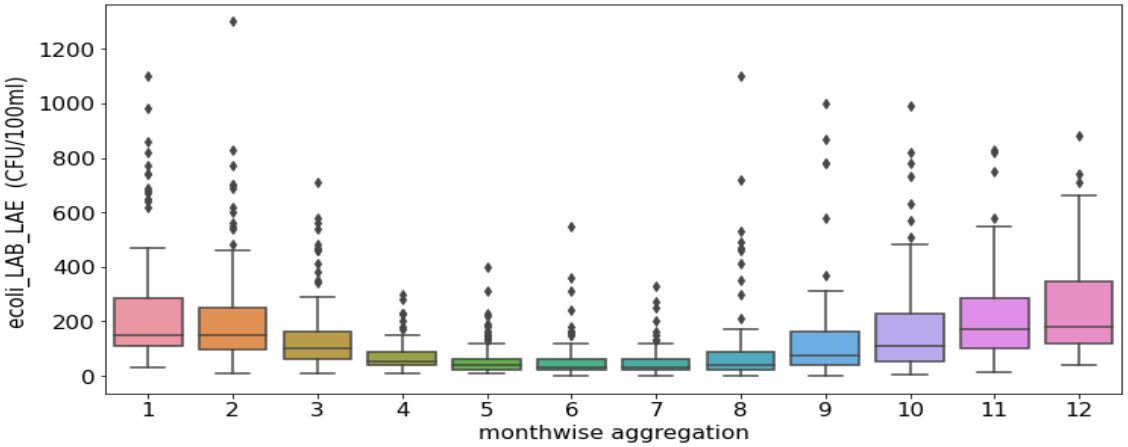
Box plot analysis

A box plot visualization was helpful to examine the distribution of variables. Figure 5.4 represents the box plot analysis of features such as *E. coli*, water temperature, turbidity at Lärjeholm, and precipitation at Komperöd. The lower horizontal line represents the minimum value, the lower line of the box represents the 25% percentile, the middle line represents the mean value, the upper line of the box presents the 75% percentile and the upper horizontal line represents the maximum value. The x-axis represents month-wise aggregation of 7 years of data (2012-2019) used in this study. The y-axis represents the data points observed.

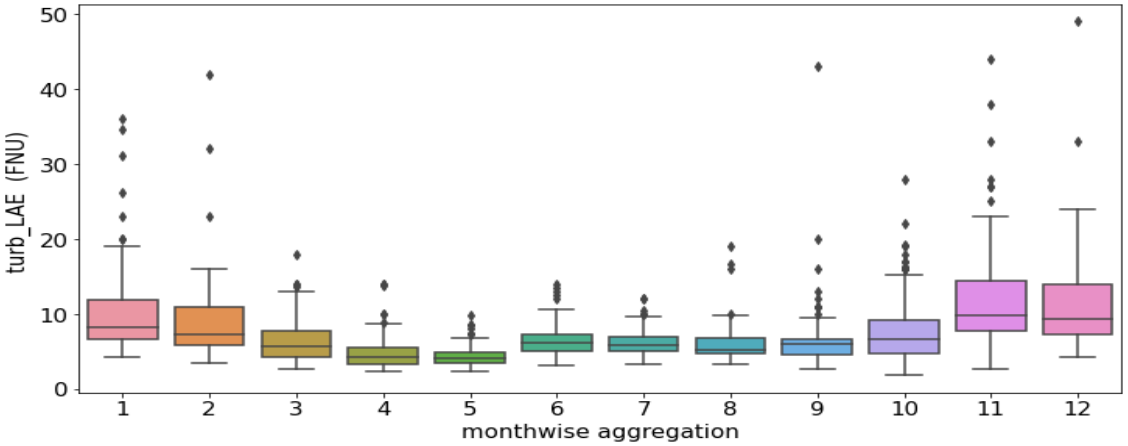
From Figure 5.4 (a), *E. coli* levels follow a parabolic trend with higher values observed at the start (January to March) and end of a year (October to December).

The maximum *E. coli* concentration during April to August was less than the average value observed during the winter period (January to March and October to December). This is due to the fact that higher rainfall event and lower water temperature were observed in the start and end of a year, which is associated with an increase in *E. coli* concentration (Figure 5.4 (c) and (d)). Figure 5.4 (b) shows that turbidity follows similar pattern as that of *E. coli*, i.e. with higher values at the start (January to March) and end of the year (October to December) and lower values in the middle of the year (April to August). This is because, with fewer rainfall events during summer, there is less amount of soil that runs off to the river, thus the amount of particles that suspends in the river surface is far less when compared to a rainy season (Sinton et al., 2002). Similar statistical inference was found for *E. coli* at Garn and coliforms at Lärjeholm and Garn (Appendix A, Figures A.5 and A.6 (a)).

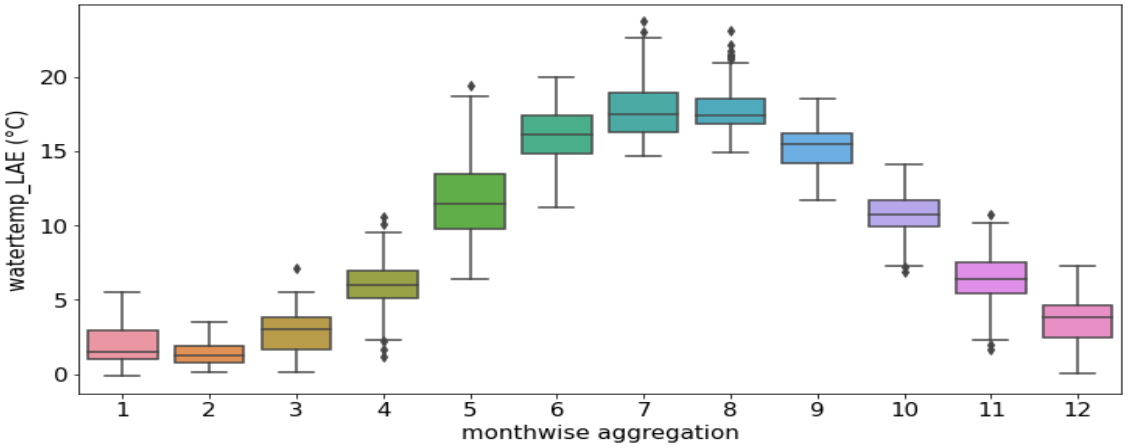
The box plot of water temperature followed an inverted U-shaped trend. It increased from April-July and declined from August to December (Figure 5.4 (c)). The boxplot of precipitation (at Komperöd) in Figure 5.4 (d) shows that precipitation did not have a clear trend, with values mostly in the range of 0-5 mm, and outliers ranging between 10-25 mm for 7 years. However, maximum values (top horizontal line) were lower during April, May and July, when compared to the maximum values observed in other months. Constant rainfall was observed from August to January with higher outliers during October. Boxplot analysis of precipitation at Vänersborg is given in Appendix A, Figure A.6 (b).


ecoli_LAB_LAE (CFU/100ml)
0
200
400
600
800
1000
1200
1
2
3
4
5
6
7
8
9
10
11
12
monthwise aggregation

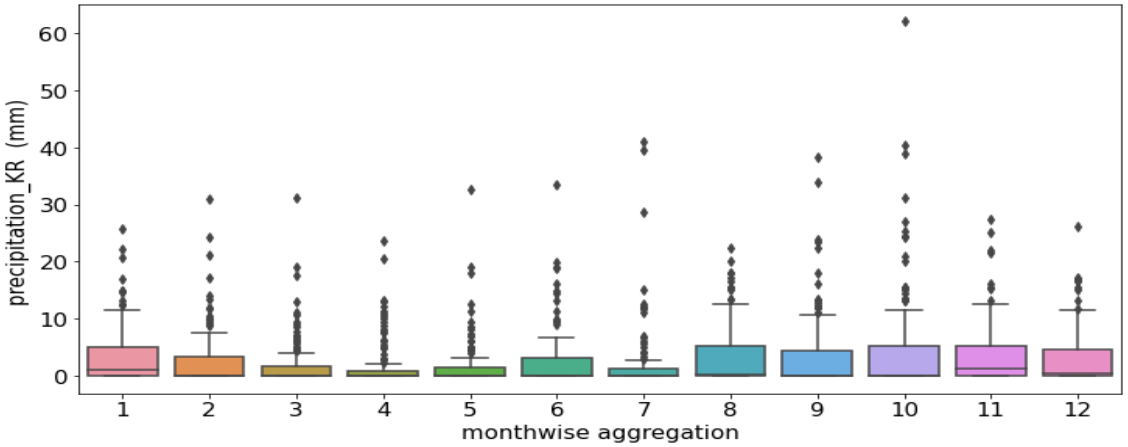
(a)


turb_LAE (FNU)
0
10
20
30
40
50
1
2
3
4
5
6
7
8
9
10
11
12
monthwise aggregation

(b)


watertemp_LAE (°C)
0
5
10
15
20
1
2
3
4
5
6
7
8
9
10
11
12
monthwise aggregation

(c)



(d)

Figure 5.4: *Box plot analysis of features observed at Lärjeholm: E. coli (Figure a), turbidity (Figure b), and water temperature (Figure c). Figure (d) shows the boxplot of precipitation at Komperöd.*

5.2 Performance of supervised machine learning algorithms

The values predicted by supervised machine learning models were validated against actual data points observed. Three types of metrics namely MAE, RMSE and R^2 score were used in the study. Tables 5.1 and 5.2 provide the error metrics of all three algorithms for baseline and complex datasets.

The performance plot visually examines how well the model fit on training and test data. The x-axis and y-axis in the plot represent the years of observations of *E. coli* levels and the observations, respectively. The blue and orange lines represent the actual observations and the predictions, respectively.

Table 5.1: *Performance metrics on baseline dataset - (CFU/100 ml)*

Test Performance	**RMSE**	**MAE**	R^2
MARS	149	100	0.27
Elasticnet	145	98	0.31
XGBoost	151	97	0.26
Train Performance	**RMSE**	**MAE**	R^2
MARS	133	84	0.35
Elasticnet	128	82	0.41
XGBoost	132	84	0.36

Table 5.2: *Performance metrics on complex dataset - (CFU/100 ml)*

Test Performance	**RMSE**	**MAE**	R^2
MARS	154	86	0.22
Elasticnet	125	77	0.48
XGBoost	128	79	0.46
Train Performance	**RMSE**	**MAE**	R^2
MARS	120	72	0.46
Elasticnet	111	67	0.54
XGBoost	110	65	0.55

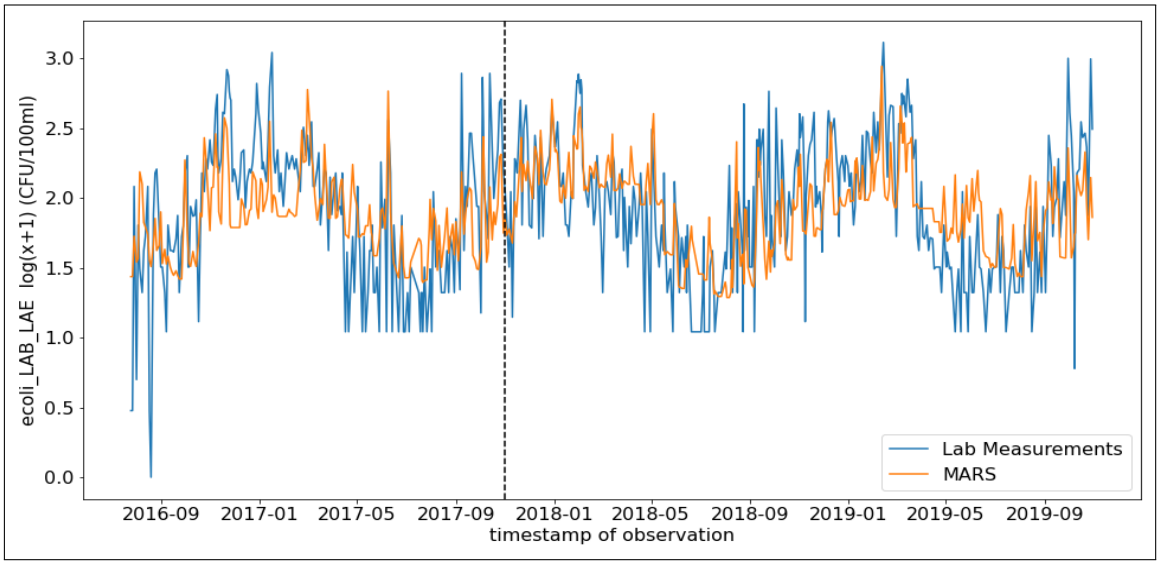
5.2.1 Linear models -MARS and Elasticnet regression

Error metrics - MARS

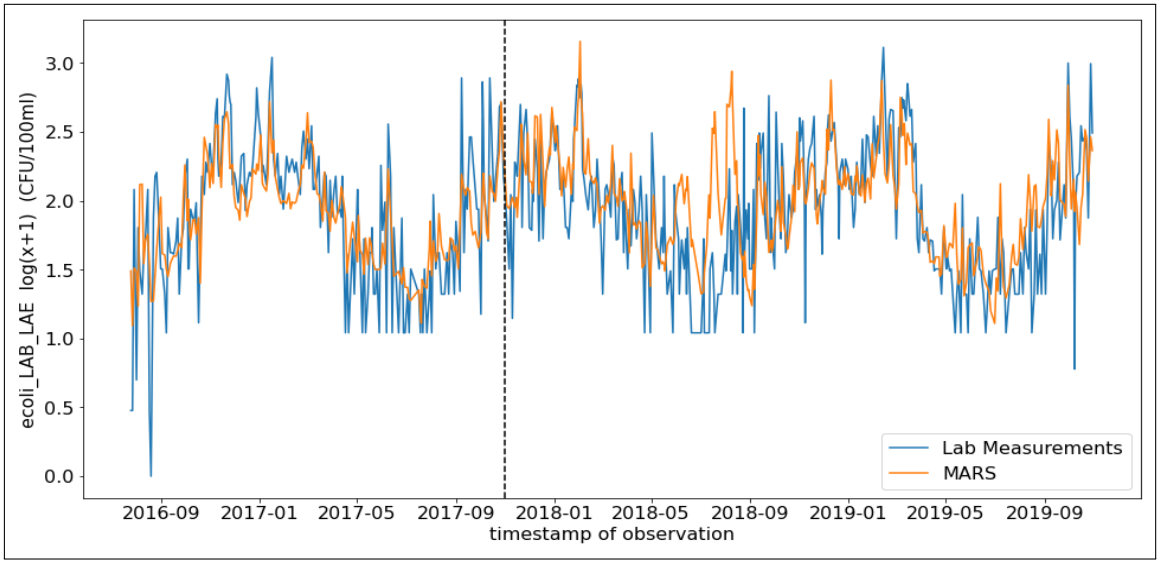
The error metrics of MARS on baseline dataset is shown in Table 5.1. The error metrics show that the MARS model performed poorly when applied on the baseline dataset. RMSE increased from 133 CFU/100 ml on training data to 149 CFU/100 ml on test data (1.12 times), while MAE increased from 84/100 ml on training data to 100 CFU/100 ml on test data (1.19 times). The R^2 score reduced from 0.35 on training data to 0.27 on testing data indicating how poor the observed outcomes are replicated by the model. The model suffered from overfitting, i.e. the model's performance was reasonable on the training data but does poorly on the test data.

Similar results were observed when the MARS model was fit on the complex dataset (Table 5.2). From Table 5.2, RMSE increased from 120 CFU/100 ml on the training data to 154 CFU/ml on the test data (1.28 times). MAE increased from 72 CFU/100 ml on the training data to 86 CFU/100 ml on test data (1.19 times). R^2 score had reduced similar to the baseline dataset i.e. from 0.46 on training data to 0.22 on the test data. This shows that MARS algorithm had poor prediction capability even with the addition of new features indicating further improvement needed as it had the highest error metric change from training to test data when compared to other algorithms used in the study.

Performance plots - MARS



(a)



(b)

Figure 5.5: *Figure (a) and (b) shows the performance of MARS on baseline and complex dataset, respectively. The left side of the dotted lines represent the last 200 points of train dataset, while the right side represents the entire test data.*

From Figure 5.5 (a), it can be seen that, the MARS model struggled to predict *E. coli* levels when limited features were available. Poor performance on both MAE and RMSE indicate that the model is incapable of predicting higher and lower values of *E. coli* levels with baseline dataset.

Compared to baseline dataset, the predictions when using complex dataset were smooth and it follows the trend of *E. coli* levels. The model improved by capturing lower values of *E. coli*, but struggles when the values were high. This was evident from Table 5.2 with high RMSE score and an improvement in the MAE score compared to baseline dataset. Performance plots of MARS on baseline and complex datasets (without $log_{10}(x + 1)$ transformation) is given in Appendix B Figures B.1 (a) and B.2 (a), respectively.

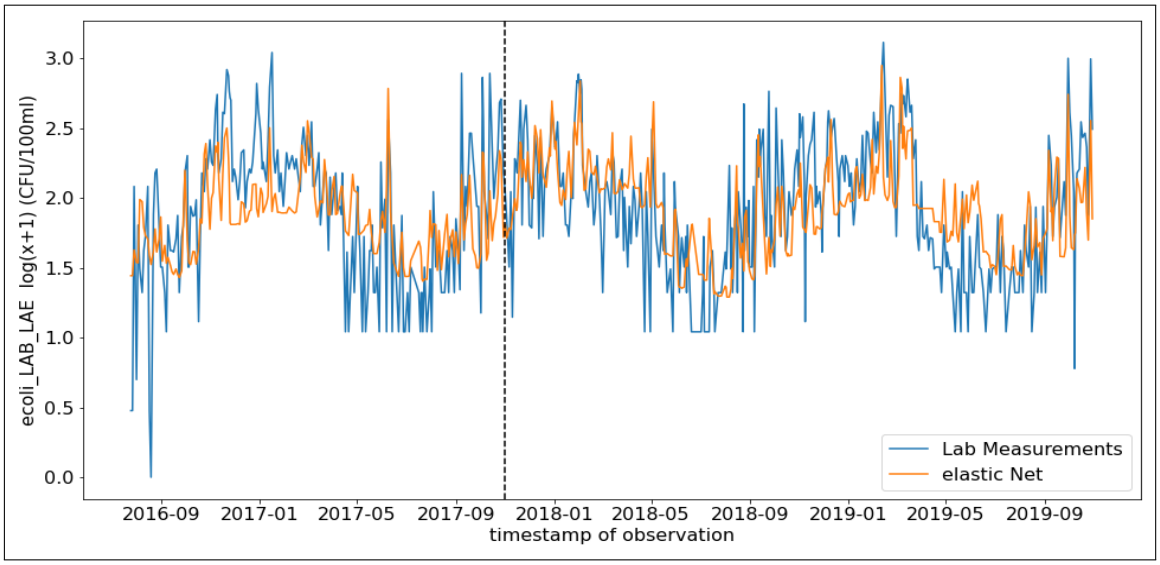
Error metrics - Elasticnet regression

Similar to MARS, Elasticnet regression performed poorly on the baseline dataset. Comparing Elasticnet performance on training and test data, Table 5.1 shows that RMSE increased from 128 CFU/100 ml on training data to 145 CFU/100 ml on the test data (1.13 times). The MAE score increased from 82 CFU/100 ml on training data to 98 CFU/100 ml on test data (1.19 times) and a decrease in R^2 score from 0.41 to 0.31 indicates poor fit on the unknown test data.

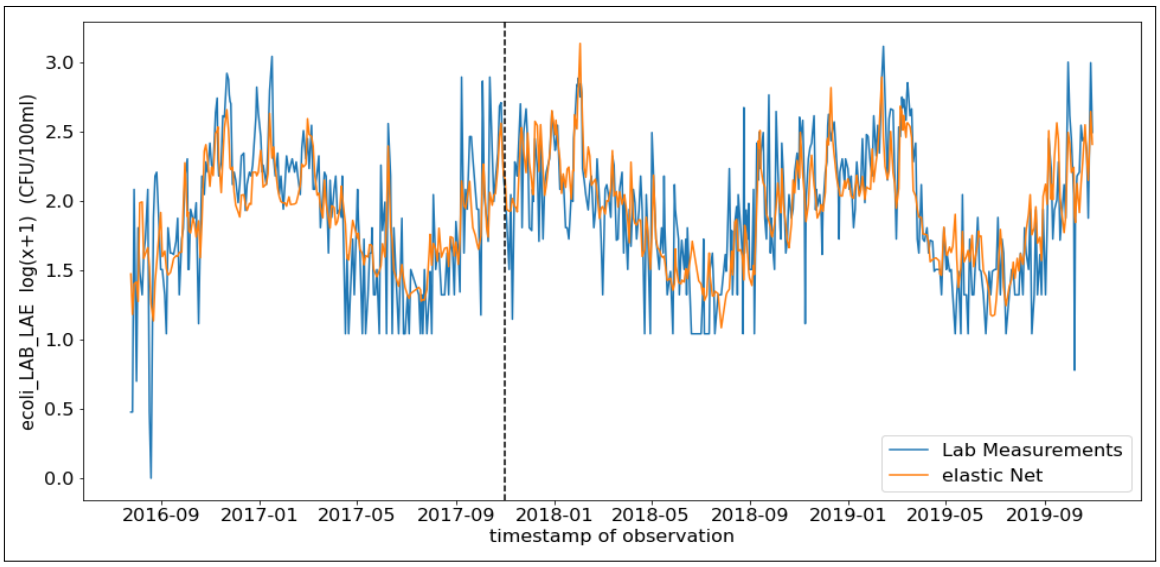
The model improved considerably on adding new features to the dataset (complex dataset). The RMSE on test data went down from 145 CFU/100 ml from baseline dataset to 125 CFU/100 ml on the complex dataset (16% reduction). MAE on test data reduced from 98 CFU/100 ml on the baseline dataset to 77 CFU/100 ml on the complex dataset (22% reduction). The R^2 score increased on test data from 0.31 (baseline dataset) to 0.48 (complex dataset) which indicates an increase in accuracy obtained when the model trained with additional features in the dataset.

Performance plots -Elasticnet regression

The Figures 5.6 ((a) and (b)) show the performance plots of Elasticnet regression on baseline and complex dataset.



(a)



(b)

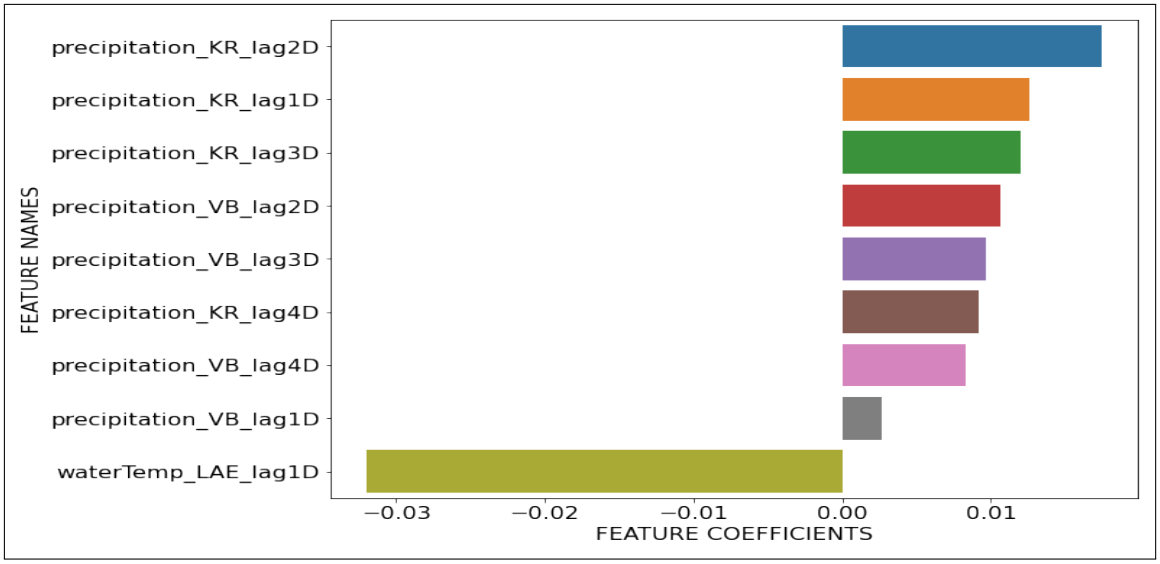
Figure 5.6: *Figure (a) and (b) shows the performance of Elasticnet on baseline and complex dataset, respectively.*

From Figures (5.5 (a) and 5.6 (a)), it is seen that Elasticnet regression performed better when compared to the MARS model on the baseline dataset. Additionally, the model also improved by capturing lower and higher *E. coli* levels when new features were added to the dataset (complex dataset) (Figure 5.6 (b)). Performance

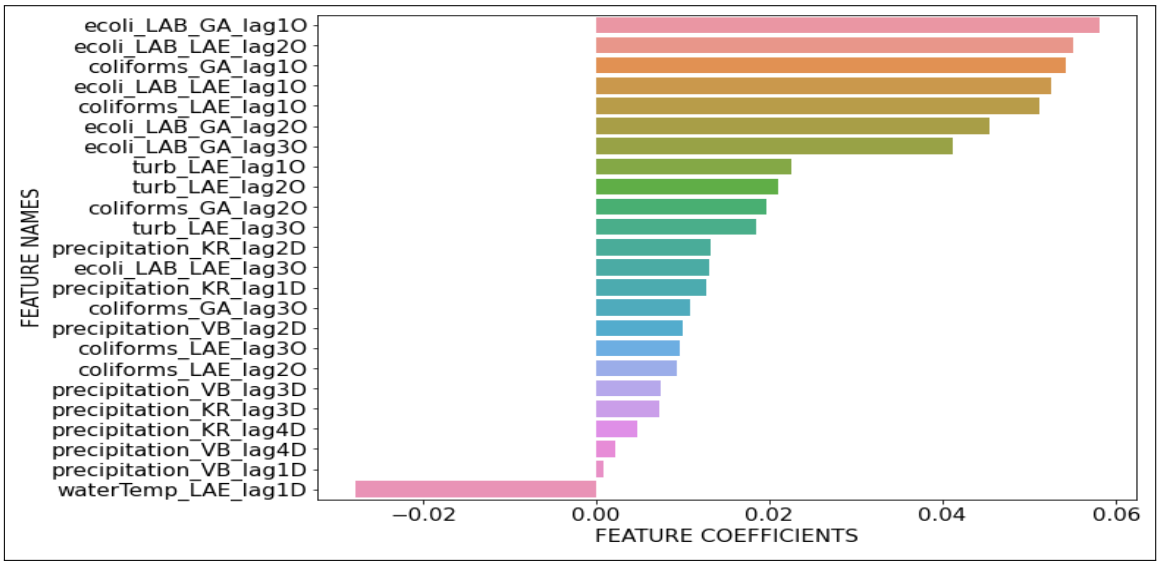
plots of Elasticnet on baseline and complex datasets (without $log_{10}(x+1)$ transformation) is given in Appendix B Figure B.1 (b) and B.2 (b).

Feature coefficients - Elasticnet regression

Figure 5.7 provides the absolute coefficient values assigned to each feature by Elasticnet regression on baseline and complex datasets. Elasticnet regression works by fitting a linear function (y= $\sum mx$ + b, where x is the input variables (features), m is the coefficients of the input variables and b is a constant). The predicted values were the output of the linear function used by Elasticnet. As mentioned in theory, section 2.2.2, the Elasticnet regression implements regularization, i.e. penalize higher coefficients of features using *l1* and *l2* penalty function. The contribution to the net loss function from *l1* and *l2* penalty is controlled by *alpha* and *lambda* and the resulting coefficient values for each feature were minimized. Figure 5.7 shows the final coefficient values of features after regularization.



(a)



(b)

Figure 5.7: *Feature coefficient plots constructed using the 'coef_' attribute of Elasticnet regressor. Figure (a) shows the feature coefficient plot constructed on baseline dataset. Figure (b) shows feature coefficient plot constructed on complex dataset.*

From Figure 5.7 (a), 2-day lag value of precipitation at Komperöd had the highest (positive) coefficient while 1-day lag of water temperature at Lärjeholm had lowest (negative) coefficient value. This was evident from the correlation matrix (section 5.1.1). Figure 5.7 (b) shows that with addition of new features, lag-observations of faecal bacteria had the highest coefficient followed by 2-day and 1-day lag observations of precipitation at Komperöd. The coefficients of certain features (e.g., 'precipitation_VB_lag4D', 'precipitation_VB_lag1D'), were pushed to the bottom

of the plot. The reason might be due to the fact that features were observed in different scales (faecal bacteria is measures in CFU/100 ml while precipitation is measured in mm), which produced biased decision from Elasticnet regression when assigning the coefficient values for each feature.

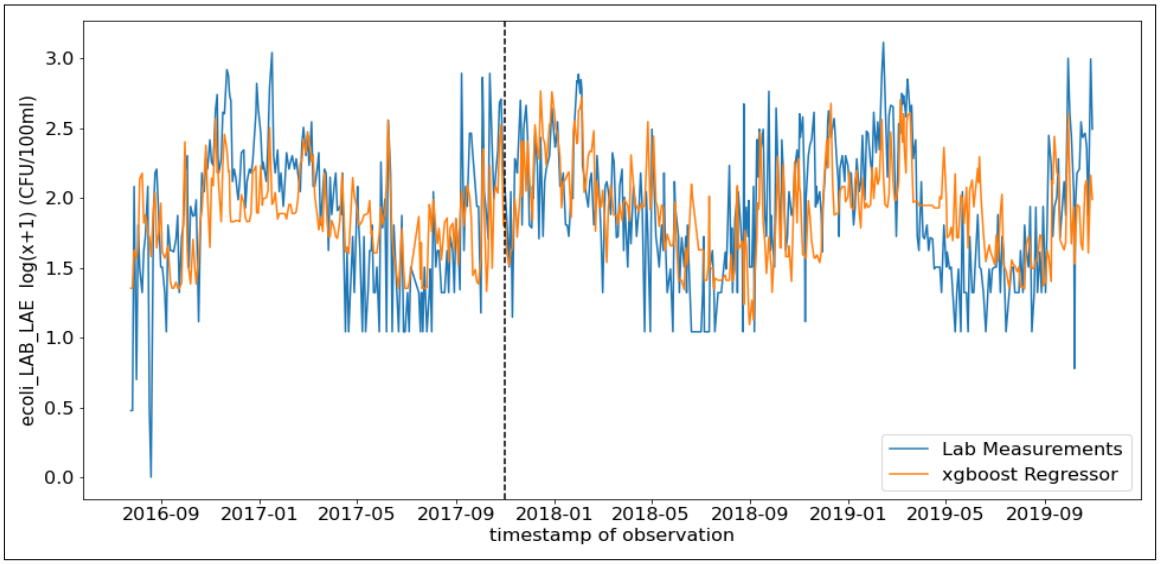
5.2.2 Non-linear model - XGBoost regression

Error metrics -XGBoost regression

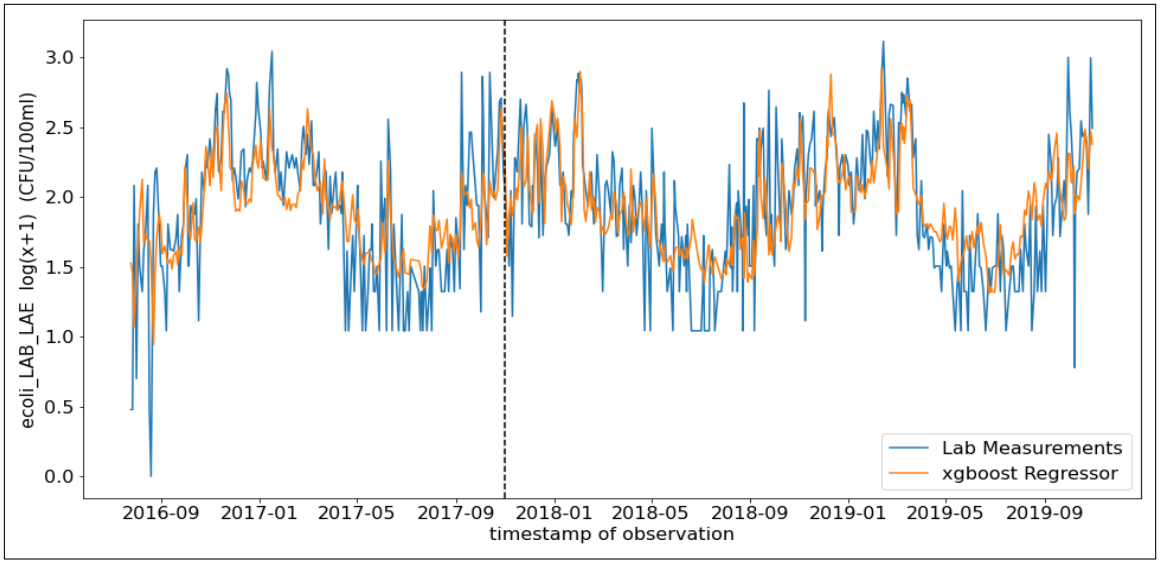
Non-linear model (XGBoost regression) performed poorly on baseline dataset, similar to linear models such as MARS and Elasticnet. The baseline error metrics Table 5.1 indicate that XGBoost suffered from overfitting. RMSE values increased from 132 CFU/100 ml on training data to 151 CFU/100 ml on the test data (1.14 times). The MAE scores increased from 84 CFU/100 ml on training data to 97 CFU/100 ml on test data (1.15 times) while there was a drop in R^2 score from 0.36 to 0.26.

When compared to the baseline dataset, XGBoost improved significantly on the complex dataset. From Tables 5.1 and 5.2, the RMSE decreased from 151 CFU/100 ml on baseline (test) data to 128 CFU/100 ml on complex (test) data. MAE decreased from 97 CFU/100 ml on baseline (test) data to 79 CFU/100 ml complex (test) dataset and R^2 score increased by 43% from 0.26 on baseline dataset to 0.46 on the complex dataset, the highest improvement rate among all three models.

Performance plots -XGBoost regression



(a)



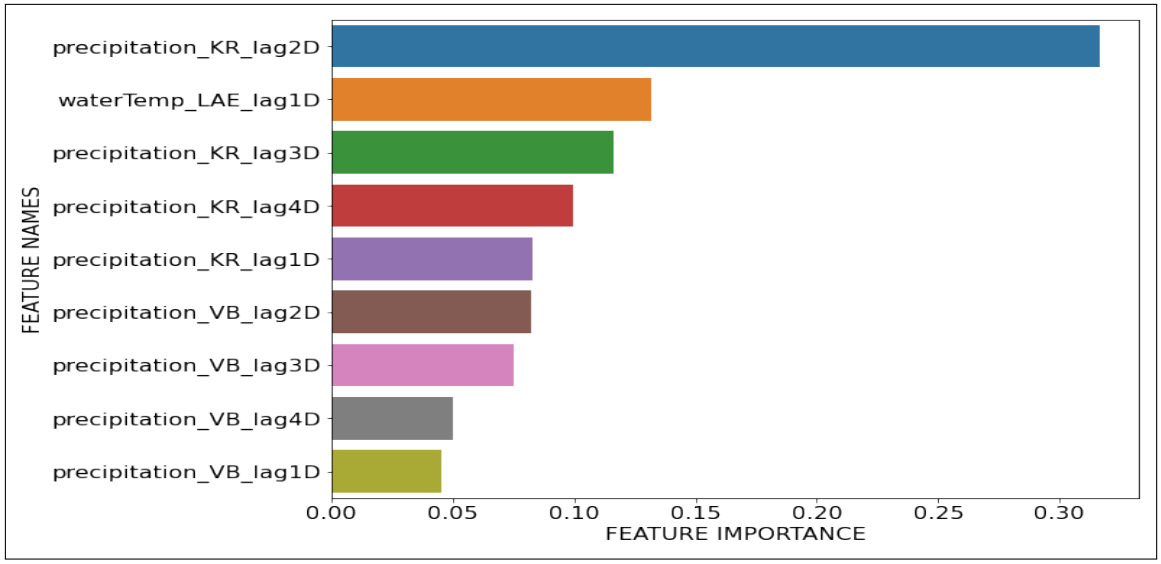
(b)

Figure 5.8: *The Figure (a) and (b) shows the performance plot of XGBoost regressor on baseline and complex dataset, respectively.*

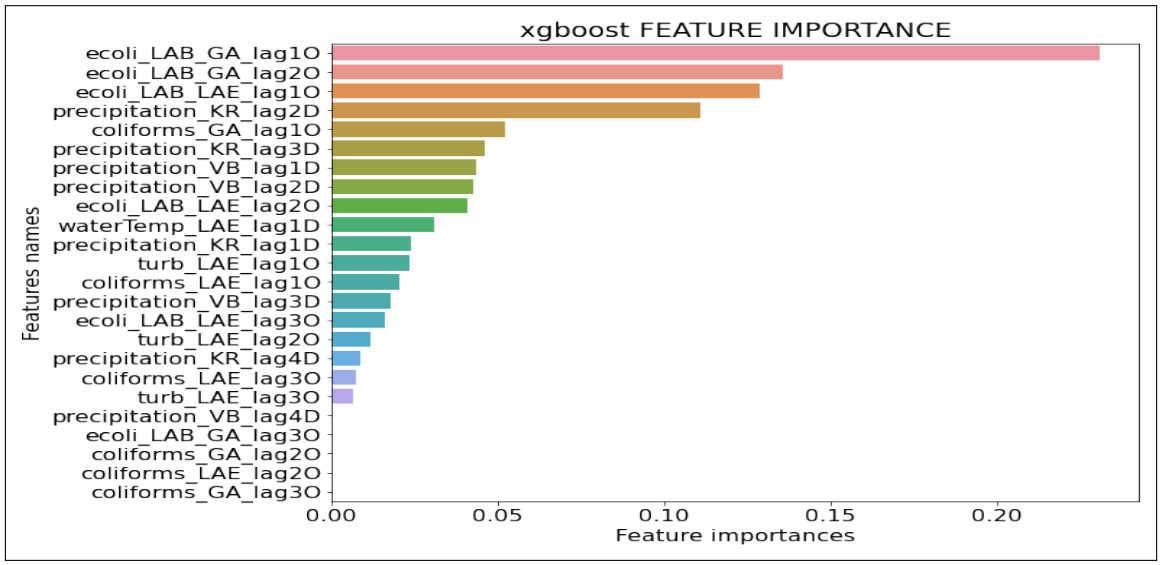
From Figure 5.8 (a), XGBoost suffered to capture peak values of *E. coli* levels but does reasonably well on lower values. Figure 5.8 (b) shows that adding new features (complex dataset) had improved the model considerably, backed up by the drop in error values. Performance plot of XGBoost on baseline and complex datasets (without $log_{10}(x+1)$ transformation) is given in Appendix B Figures B.1 (c) and B.2 (c).

Feature Importances

Unlike the feature coefficient plot, the feature importance plot of XGBoost provides the value or usefulness of each feature in predicting *E. coli* levels. The feature importance plot ranks the features according to the use of features in splitting the decision tree with the lowest resulting loss function.



(a)



(b)

Figure 5.9: *Feature importance plots constructed using the 'feature_importances_' attribute of XGBoost Regressor. Figure (a) shows the feature importance constructed on baseline dataset. Figure (b) shows feature importance plot constructed on complex dataset.*

From Figure 5.9 (a), 2-day lag values of precipitation at Komperöd was considered to be the most important feature to be used in the split, while 4-day lag values of precipitation at Vänersborg was the least important. This is realistic as Vänersborg is around 90 km from Lärjeholm and with 4 days of lag value, it might have the least effect on *E. coli* levels.

Adding new features resulted in changing the feature importance plot as shown in Figure 5.9 (b). The first three important features were 1-lag and 2-lag observations of *E. coli* at Garn and Lärjeholm, followed by 2-day lag values of precipitation at Komperöd. This was realistic as lag values of *E. coli* had more importance in predicting its future levels. 1-lag observation of turbidity was one of the important features as an increase in turbidity reduced the amount of sunlight to enter the river and thereby providing room for *E. coli* levels to decay at a slower rate (Göransson et al., 2013). From 5.9 (b), certain features (e.g., 'ecoli_LAB_GA_3O', 'coliforms_GA_lag2O', 'coliforms_GA_lag3O', 'coliforms_LAE_lag2O') had zero importance in predicting *E. coli* levels. The reason might be that faecal bacteria decay and reproduce rapidly (on average doubles every 20 minutes, and at this rate would increase to 1 million in 7 hours (Chan et al., 2015). Average time taken to reduce 90% of *E. coli* concentration is 10.8 to 42.3 hours (Chan et al., 2015). Hence 3-day and 2-day lag values of coliforms (a group of bacteria, where *E. coli* is a subset) were not importance in predicting *E. coli* levels.

5.2.3 Variation analysis

One of the hypotheses in this study was to check if the supervised machine learning algorithms performed better when additional features were added to the baseline dataset. A simple variation analysis was used to find the improvement in error metrics, when additional features were added to the baseline dataset. Using statistical testing tools such as A/B testing, the initial dataset after being processed (after completing data preprocessing steps - Table 4.2) was divided into two: 1. baseline dataset with limited features and 2. complex dataset with additional features. Relative change (theory, section 2.4.4) in error metric was helpful to examine and compare models on baseline and complex datasets.

Table 5.3: *Relative change in error metrics from baseline dataset to complex dataset for the three algorithms.*

Metrics	**Effect on residuals**	**Ranking**	**Δ *error***
MAE	higher on smaller residuals	• Elasticnet • XGBoost • MARS	• 21% • 18% • 14%
RMSE	higher on larger residuals	• XGBoost • Elasticnet • MARS	• 15% • 13% • -3%
R^2	higher on larger residuals	• XGBoost • Elasticnet • MARS	• 43% • 35% • -18%

Table 5.3 shows the error metrics used in this study, effect of error metrics on the residuals and ranking of algorithms based on the relative change in error when it is trained on complex dataset. MAE score for Elasticnet regression reduced around 21% from baseline dataset followed by XGBoost with the reduction of 18% from the baseline dataset. Since MAE has larger effect on small residuals values (|actual-predicted|), it was seen that Elasticnet performed better in predicting lower *E. coli* values (summer) with additional features in the dataset.

An improvement in R^2 score and RMSE indicate that the model predicted higher values of residuals more accurately. RMSE error improved around 15% for XGBoost and 13% for Elasticnet regression from baseline dataset when trained on complex dataset. XGBoost had the highest improvement rate in error metrics when it was trained with additional features. When the number of features were low, XGBoost suffered in predicting larger values of *E. coli* (lowest R^2 - 0.26), whereas after adding new features, the model improved around 43% (R^2 - 0.46) from baseline dataset, the highest improvement in R^2 score among the three models. MARS turned out to be the least performing among all three models, with negative error change (performance of the model has downgraded) in RMSE, and R^2 error values, when additional features were added to the model (complex dataset). This indicate that MARS was not suitable for further analysis in this study.

6
Discussion

The discussion section deals with exploring the results that would reflect on the hypothesis framed in this study. It also highlights and compares the study with the past research on faecal indicator bacteria (FIB) prediction.

Importance of water sanitation and management

Pure drinking water has always been a prime sustainability goal for all countries. The United Nation Environment Programme listed 'Clean water and Sanitation' as their 6^{th} goal in their 17 sustainable development goals to be achieved within 2030 (UNSDG, 2021). Proper water quality monitoring and treatment are important steps since lack of these can cause contamination of water leading to degradation in water quality. Previous research stresses the association of elevated *E. coli* concentration and past rainfall events, that causes illness in human beings (Drayna et al., 2010; Tornevi et al., 2013). Dryna et al. (2010) report an increase in emergency visits at Milwaukee (WI), when there are elevated *E. coli* levels, after 4 days following heavy rainfall. Increased nurse calls were linked to an increase in concentration of *E. coli* with 4-days earlier rainfall event in Gothenburg (Tornevi et al., 2013). It is necessary to take action on proper water treatment and sanitation procedures when heavy rainfall event occurs. AI has been on a revolution for a long time and it has created many interesting improvements in the field of sustainability and water sanitation management (Nishant et al., 2020). To reduce the time complexity and expensive lab analysis, it is important to utilize AI for the prediction of faecal bacteria levels, so that proper sanitary action can take place in advance.

Comparison between linear and non-linear models

Tables 5.1 and 5.2 provide the error metrics for all three algorithms. The study used two linear models, i.e. MARS and Elasticnet for predicting *E. coli* levels. MARS, could not effectively predict *E. coli* levels because it relies on hinge functions (rectified linear functions), that approximate values to zero in certain situations (theory chapter, section 2.1.1). Due to this attribute, the MARS model is suitable when there are more complex, non-linear interactions between features. However, the underlying relationship between faecal and physio-chemical indicators used in this study tends to be linear in nature. This might be the reason for the poor performance of the MARS in predicting *E. coli* levels. From Tables 5.1 and 5.2, Elasticnet proved to be the most effective in predicting *E. coli* levels among all three algorithms. One reason would be that Elasticnet uses regularization that can penalise

higher coefficient values of the features. By this way, Elasticnet can reduce the effect of higher coefficients that may not be necessary in predicting the target variable.

The study conducted by Laureano-Rosario et al. (2019) support non-linear algorithms for faecal bacteria prediction at Escambron beach, Puerto Rico. The authors report that the relationship between faecal bacteria and various other indicators such as environmental factors, physio-chemical, and human interaction to be non-linear in nature. Contradicting to the research conducted by Laureano-Rosario et al. (2019), this study is based on the dataset obtained from the river Göta älv, and hence the non-linear model (XGBoost) used in this study did not prove to be the most effective in predicting *E. coli* levels. Another reason could be the choice of input variables used for predicting *E. coli* levels. Since the relationship between *E. coli* and physio-chemical features used in this study tend to be linear in nature, XGBoost regression was not able to perform effectively. If more complex and non-linear indicators (e.g., human interactions, environmental factors, etc,. that increases the variance in the dataset) were included, then it would increase the accuracy of non-linear model (XGBoost).

Comparison with previous studies

Various studies employed machine learning techniques to predict faecal bacteria levels using physio-chemical and faecal indicators (Jayalakshmi & Santhakumaran, 2011; Edberg et al., 2000; Zhang et al., 2018). Specifically, studies were focused on using only lag of precipitation/rainfall to predict FIB levels (Avila et al., 2018; Vijayashanthar et al., 2018; Tornevi et al., 2014). Often, data normalization, filling in missing values, and dealing with outliers were found to be widely used as preprocessing steps for data cleaning and transformation (Mohammed et al., 2018; Dheda & Cheng, 2020; Joslyn, 2018). From the literature review, it was evident that supervised-learning techniques held a prominent place in prediction of FIB (Avila et al., 2018; Eleria & Vogel, 2005). Algorithms like ANN were widely used by researchers due to their data-driven adaptive technology i.e. to tap information automatically and learn from non-linear relationships (Choi & Bae, 2018; He & He, 2008; Laureano-Rosario et al., 2019).

Compared to the literature review, this study incorporated three supervised machine learning algorithms namely MARS and Elasticnet regression (linear-models), and XGBoost (non-linear tree-based model). Results from Eleria and Vogel (2005) conclude that an ordinary least square linear regression for predicting coliform levels resulted in R^2 score between 0.46 to 0.56 which approximately coincides with the score obtained in this study by linear models. The conclusion from Tornevi et al. (2014) report that 2-day lag values of precipitation was the most influential in predicting the *E. coli* levels. Contradicting to the study by Tornevi et al. (2014), lag values of faecal bacteria proved to be more effective in predicting future *E. coli* levels in this study. Another conclusion by Tornevi et al. (2014) specify that faecal bacteria levels were higher during the winter and then gradually decreased as the season changed. This was evident from the insights obtained from box plot analysis

performed in this study. Although Tornevi et al. (2014) and this study have both evaluated Göta älv, it is important to remember that two different data sets have been used. The data set used by Tornevi et al. (2014) span from 2004 to 2010 while the data set used in this study covered from 2012 to 2019. One conclusion drawn by Laureano-Rosario et al. (2019) mention that non-linear algorithms perform better on predicting *E. coli* levels. Contrasting to the study, the non-linear model (XGBoost) did not prove to be the best among the three algorithms used in this study.

7
Conclusion

The objective of this study was to predict *E. coli* levels at Lärjeholm using supervised machine learning algorithms. Exploratory data analysis was performed to obtain statistical inference on the data. Error metrics were used to evaluate the performance of supervised machine learning models. The following are the conclusions derived from the study:

- From exploratory data analysis, it was evident that an increase in faecal bacteria was associated with an increase in precipitation events. More precisely, prior precipitation events at Komperöd tend to increase *E. coli* levels at Lärjeholm due to stormwater runoff and excess sewage, that is carried along with the water to Lärjeholm.
- Lag values affects the concentration of *E. coli* levels at Lärjeholm. When the features were limited, lag values of precipitation were found to be the most important input variable in predicting *E. coli* levels. When additional features were added, lag values of faecal indicators were considered to be the most important features in predicting future *E. coli* levels.
- Elasticnet and XGBoost performed significantly better on complex dataset, while the performance of MARS deteriorated on the complex dataset when compared to the baseline dataset.
- Elasticnet regression (linear model) had the best performance metrics among all three model on both the datasets.
- XGBoost (non-linear model) was expected to perform better than linear models but failed to do so. However, by adding features the relative change in R^2 score was around 43% from baseline dataset to complex dataset, the highest improvement rate among the three models with the addition of new features.

7.1 Recommendations

The purpose of this study was to predict *E. coli* levels using a set of indicators as input variables to supervised machine learning algorithms. Though the results were found to be reasonable, there is potential to upgrade the methods and techniques used in this study. It is worth checking if the results can be improved by adding more observations from other water quality monitoring plants namely Södre Nol and Surte. The number of independent variables (features) can be increased by adding physio-chemical indicators such as pH, conductivity, colour, dissolved oxygen, flow rate and sunlight intensity. The number of data points can also be increased, to

check if there is an improvement in the error metrics. The insights obtained from the literature suggest that it is worth checking the performance of deep learning algorithms namely RNN, ANFIS, etc., since these algorithms were widely known for high processing speed and accurate results on predicting faecal bacteria levels (Wijaya et al., 2020; Won et al., 2016).

References

Abimbola, O. P., Mittelstet, A. R., Messer, T. L., Berry, E. D., Bartelt-Hunt, S. L., & Hansen, S. P. (2020). Predicting escherichia coli loads in cascading dams with machine learning: An integration of hydrometeorology, animal density and grazing pattern. *Science of The Total Environment*, *722*, 137894.

Ahmed, U., Mumtaz, R., Anwar, H., Shah, A. A., Irfan, R., & García-Nieto, J. (2019). Efficient water quality prediction using supervised machine learning. *Water*, *11*(11), 2210.

Avila, R., Horn, B., Moriarty, E., Hodson, R., & Moltchanova, E. (2018). Evaluating statistical model performance in water quality prediction. *Journal of environmental management*, *206*, 910–919.

Bardenet, R., Brendel, M., Kégl, B., & Sebag, M. (2013). Collaborative hyperparameter tuning. In *International conference on machine learning* (pp. 199–207).

Behrens, J. T., & Yu, C.-h. (2003). Exploratory data analysis. *Handbook of psychology*, *2*, 33–64.

Brownlee, J. (2020a, Jun). Elasticnet regressor. *Machine Learning Mastery*. Retrieved from `https://machinelearningmastery.com/elastic-net-regression-in-python/`

Brownlee, J. (2020b, Aug). Feature importance and feature selection with xgboost in python. *Machine Learning Mastery*. Retrieved from `https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/`

Brownlee, J. (2021, Apr). Multivariate adaptive regression splines (mars) in python. *Machine Learning Mastery*. Retrieved from `https://machinelearningmastery.com/multivariate-adaptive-regression-splines-mars-in-python/`

Chan, Y., Thoe, W., & Lee, J. H. (2015). Field and laboratory studies of escherichia coli decay rate in subtropical coastal water. *Journal of Hydro-Environment Research*, *9*(1), 1–14.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794).

Choi, S.-W., & Bae, H.-K. (2018). Daily prediction of total coliform concentrations using artificial neural networks. *KSCE Journal of Civil Engineering*, *22*(2), 467–474.

Clark, L. A., & Pregibon, D. (2017). Tree-based models. In *Statistical models in s* (pp. 377–419). Routledge.

Correlation. (2021). pandas corr. *pandas.DataFrame.corr - pandas 1.2.5 documentation*. Retrieved from https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html

Cox, C., Moscardini, E. H., Cohen, A. S., & Tucker, R. P. (2020). Machine learning for suicidology: A practical review of exploratory and hypothesis-driven approaches. *Clinical Psychology Review*, 101940.

Dheda, D., & Cheng, L. (2020). A multivariate water quality parameter prediction model using recurrent neural network. *arXiv preprint arXiv:2003.11492*.

Drayna, P., McLellan, S. L., Simpson, P., Li, S.-H., & Gorelick, M. H. (2010). Association between rainfall and pediatric emergency department visits for acute gastrointestinal illness. *Environmental health perspectives*, *118*(10), 1439–1443.

E. coli. (2021). World Health Organization. Retrieved from https://www.who.int/news-room/fact-sheets/detail/e-coli

Edberg, S., Rice, E., Karlin, R., & Allen, M. (2000). Escherichia coli: the best biological drinking water indicator for public health protection. *Journal of applied microbiology*, *88*(S1), 106S–116S.

Eleria, A., & Vogel, R. M. (2005). Predicting fecal coliform bacteria levels in the charles river, massachusetts, usa 1. *JAWRA Journal of the American Water Resources Association*, *41*(5), 1195–1209.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, 1–67.

Göransson, G., Larson, M., & Bendz, D. (2013). Variation in turbidity with precipitation and flow in a regulated river system–river göta älv, sw sweden. *Hydrology and Earth System Sciences*, *17*(7), 2529–2542.

Gross, K. (2021). Machine learning and linear models: How they work (in plain english). *Blog*. Retrieved from https://blog.dataiku.com/top-machine-learning-algorithms-how-they-work-in-plain-english-1#:~:text=Machinelearningisreallyall,alineofbestfit.

The göta älv estuary. (2021). *, Interreg VB North Sea Region Programme*. Retrieved from https://northsearegion.eu/immerse/project-estuaries/the-goeta-aelv-estuary/

Haramoto, E., Katayama, H., Oguma, K., Koibuchi, Y., Furumai, H., & Ohgaki, S. (2006). Effects of rainfall on the occurrence of human adenoviruses, total coliforms, and escherichia coli in seawater. *Water science and technology*, *54*(3), 225–230.

He, L.-M. L., & He, Z.-L. (2008). Water quality prediction of marine recreational beaches receiving watershed baseflow and stormwater runoff in southern california, usa. *Water research*, *42*(10-11), 2563–2573.

Jayalakshmi, T., & Santhakumaran, A. (2011). Statistical normalization and back propagation for classification. *International Journal of Computer Theory and Engineering*, *3*(1), 1793–8201.

Jiang, Y., Nan, Z., & Yang, S. (2013). Risk assessment of water quality using monte carlo simulation and artificial neural network method. *Journal of environmental management*, *122*, 130–136.

Joslyn, K. (2018). Water quality factor prediction using supervised machine learning.

Katie, G. . (2021). Tree-based models: How they work (in plain english!). *Blog*. Retrieved from https://blog.dataiku.com/tree-based-models-how-they-work-in-plain-english

KDnuggets. (2021). Unveiling mathematics behind xgboost. Retrieved from https://www.kdnuggets.com/2018/08/unveiling-mathematics-behind-xgboost.html

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, *160*(1), 3–24.

Laureano-Rosario, A. E., Duncan, A. P., Symonds, E. M., Savic, D. A., & Muller-Karger, F. E. (2019). Predicting culturable enterococci exceedances at escambron beach, san juan, puerto rico using satellite remote sensing and artificial neural networks. *Journal of water and health*, *17*(1), 137–148.

Mohammed, H., Longva, A., & Seidu, R. (2018). Predictive analysis of microbial water quality using machine-learning algorithms. *Environmental Research, Engineering and Management*, *74*(1), 7–20.

Motamarri, S., & Boccelli, D. L. (2012). Development of a neural-based forecasting tool to classify recreational water quality using fecal indicator organisms. *Water research*, *46*(14), 4508–4520.

Muharemi, F., Logofătu, D., & Leon, F. (2019). Machine learning approaches for anomaly detection of water quality on a real-world data set. *Journal of Information and Telecommunication*, *3*(3), 294–307.

Naser, M., & Alavi, A. (2020). Insights into performance fitness and error metrics for machine learning. *arXiv preprint arXiv:2006.00887*.

Nishant, R., Kennedy, M., & Corbett, J. (2020). Artificial intelligence for sustainability: Challenges, opportunities, and a research agenda. *International Journal of Information Management*, *53*, 102104.

Odonkor, S. T., & Ampofo, J. K. (2013). Escherichia coli as an indicator of bacteriological quality of water: an overview. *Microbiology research*, *4*(1), 5–11.

Pandas. (2021). Pandas. *pandas.DataFrame.describe - pandas 1.2.5 documentation*. Retrieved from https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html?highlight=describe#pandas.DataFrame.describe

Price, R. G., & Wildeboer, D. (2017). E. coli as an indicator of contamination and health risk in environmental waters. *Escherichia coli-Recent Advances on Physiology, Pathogenesis and Biotechnological Applications*.

Pyearth. (2021). Introduction to mars. *py-earth 0.1.0 documentation*. Retrieved from https://contrib.scikit-learn.org/py-earth/content.html

Reis, J., Santo, P. E., & Melão, N. (2019). Impacts of artificial intelligence on public administration: A systematic literature review. In *2019 14th iberian conference on information systems and technologies (cisti)* (pp. 1–7).

Saxena, S. (2020, Dec). What is a/b testing: How data scientist leverage a/b testing. *Analytics Vidhya*. Retrieved from https://www.analyticsvidhya.com/blog/2020/10/ab-testing-data-science/

Scikit. (2021a). *3.1. cross-validation: evaluating estimator performance.* Retrieved from https://scikit-learn.org/stable/modules/cross

_validation.html

Scikit. (2021b). missingno. *PyPI*. Retrieved from https://pypi.org/project/missingno/

Sinton, W, L., Hall, H, C., Lynch, P. A., & Davies-Colley, R. J. (2002). Sunlight inactivation of fecal indicator bacteria and bacteriophages from waste stabilization pond effluent in fresh and saline waters. *Applied and environmental microbiology*, *68*(3), 1122–1131.

Sklearn. (2021a). sklearn-elasticnet. *scikit*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html

Sklearn. (2021b). sklearn-traintestsplit. *scikit*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

Solanki, A., Agrawal, H., & Khare, K. (2015). Predictive analysis of water quality parameters using deep learning. *International Journal of Computer Applications*, *125*(9), 0975–8887.

Strobl, R. O., & Robillard, P. D. (2006). Artificial intelligence technologies in surface water quality monitoring. *Water international*, *31*(2), 198–209.

Thoe, W., Gold, M., Griesbach, A., Grimmer, M., Taggart, M., & Boehm, A. (2014). Predicting water quality at santa monica beach: evaluation of five different models for public notification of unsafe swimming conditions. *Water research*, *67*, 105–117.

Tornevi, Axelsson, G., & Forsberg, B. (2013). Association between precipitation upstream of a drinking water utility and nurse advice calls relating to acute gastrointestinal illnesses. *PloS one*, *8*(7), e69918.

Tornevi, Bergstedt, O., & Forsberg, B. (2014). Precipitation effects on microbial pollution in a river: lag structures and seasonal effect modification. *PloS one*, *9*(5), e98546.

UNSDG. (2019). Goal 6: ensure availability and sustainable management of water and sanitation for all. *target*, 100.

UNSDG. (2021). Unsdg. *United Nations*. Retrieved from https://www.un.org/sustainabledevelopment/water-and-sanitation/

Vijayashanthar, Vasikan, Qiao, J., Zhu, Z., Entwistle, P., Yu, & Guan. (2018). Modeling fecal indicator bacteria in urban waterways using artificial neural networks. *Journal of Environmental Engineering*, *144*(6), 05018003.

WHO. (2020). Achieving quality health services for all, through better water, sanitation and hygiene: lessons from three african countries.

Wijaya, S., Saumnuari, M., Nasution, A., Ramadhan, D., & Hasibuan, L. (2020). Deep learning approach for predicting the therapeutic usage of jamu. , *1566*(1), 012052.

Winfield, M. D., & Groisman, E. A. (2003). Role of nonhost environments in the lifestyles of salmonella and escherichia coli. *Applied and environmental microbiology*, *69*(7), 3687–3694.

Won, I., Seo, Yun, S. H., & Choi, S. Y. (2016). Forecasting water quality parameters by ann model using pre-processing technique at the downstream of cheongpyeong dam. *Procedia Engineering*, *154*, 1110–1115.

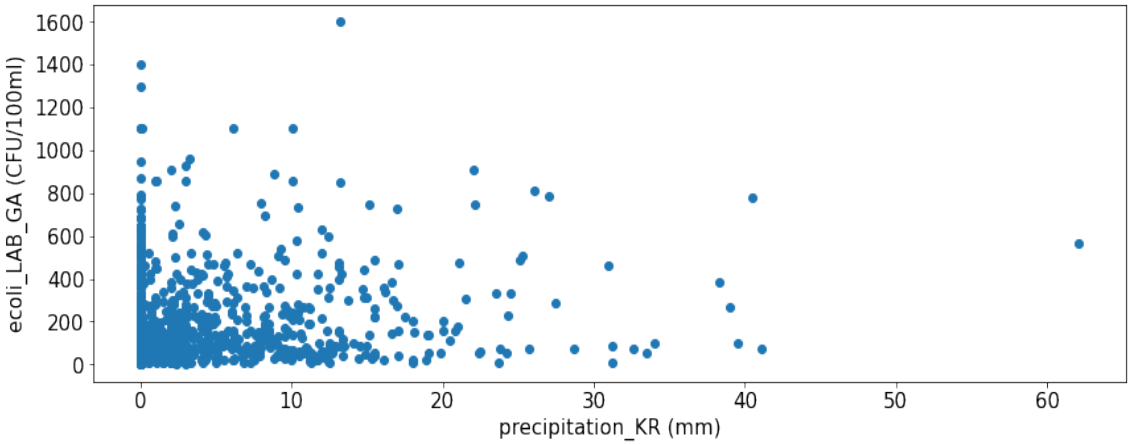
Xgboost parameters. (2021). Retrieved from `https://xgboost.readthedocs.io/en/latest/parameter.html`

Zhang, Juan, Qiu, Han, Li, Xiaoyu, ... S, M. (2018). Real-time nowcasting of microbiological water quality at recreational beaches: A wavelet and artificial neural network-based hybrid modeling approach. *Environmental science & technology*, *52*(15), 8446–8455.

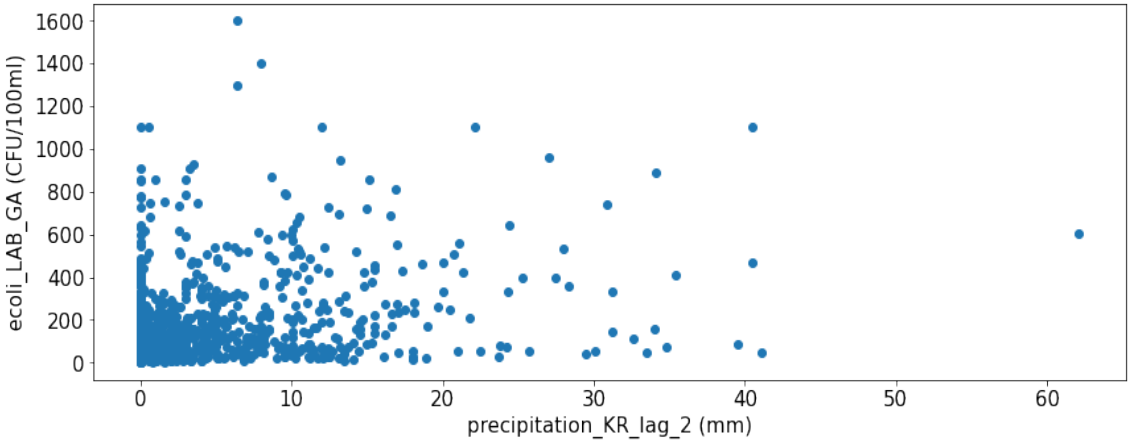
Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67*(2), 301–320.

A

Appendix

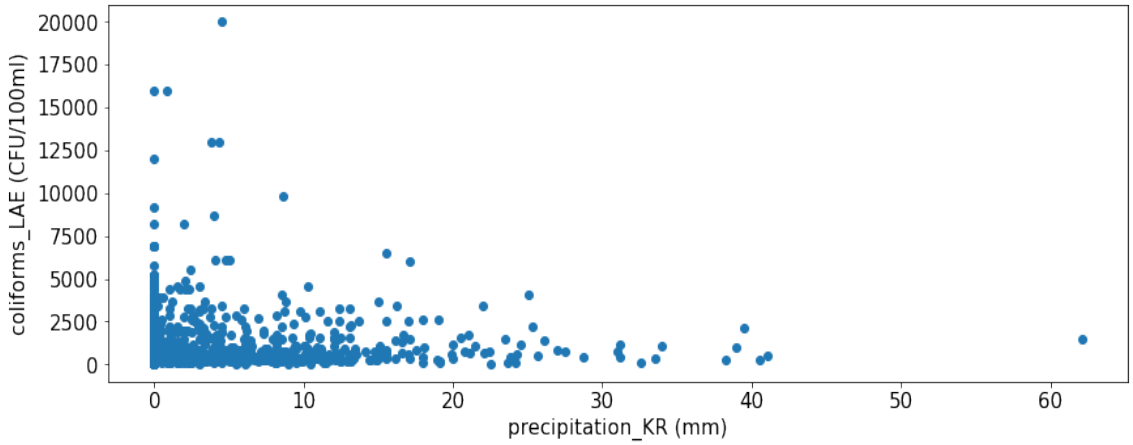


(a)

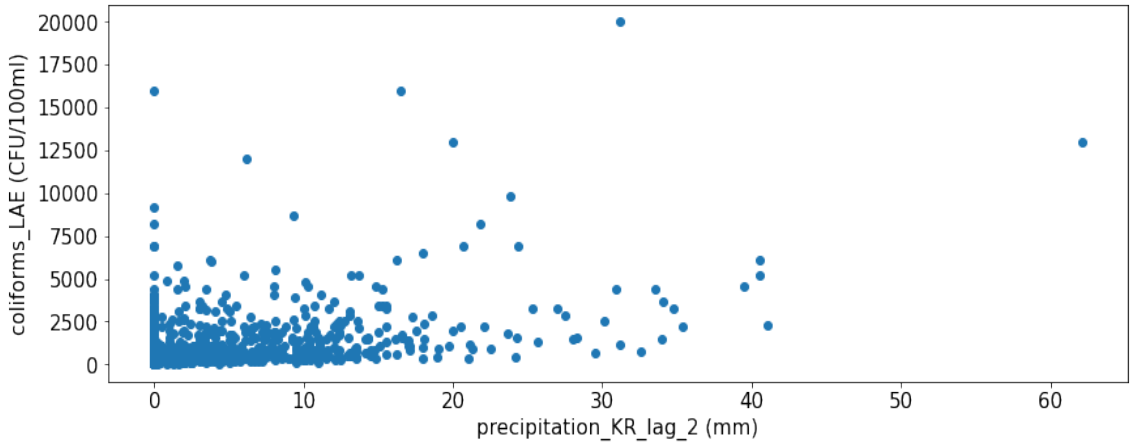


(b)

Figure A.1: *Relationship of precipitation and its 2-day lag values with different features. Figure (a) represents precipitation at Komperöd with E. coli levels at Garn, Figure (b) represents 2-lag observations of precipitation at Komperöd with E. coli at Garn.*

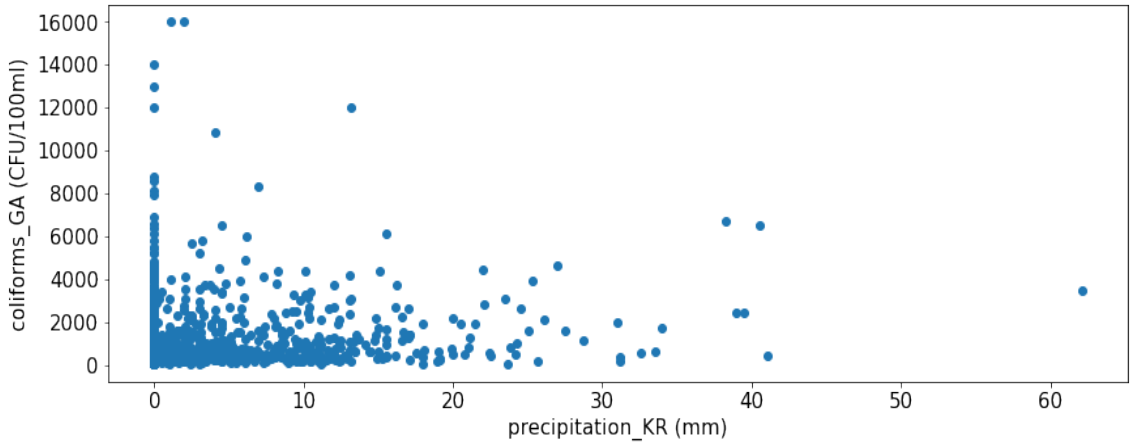


(a)

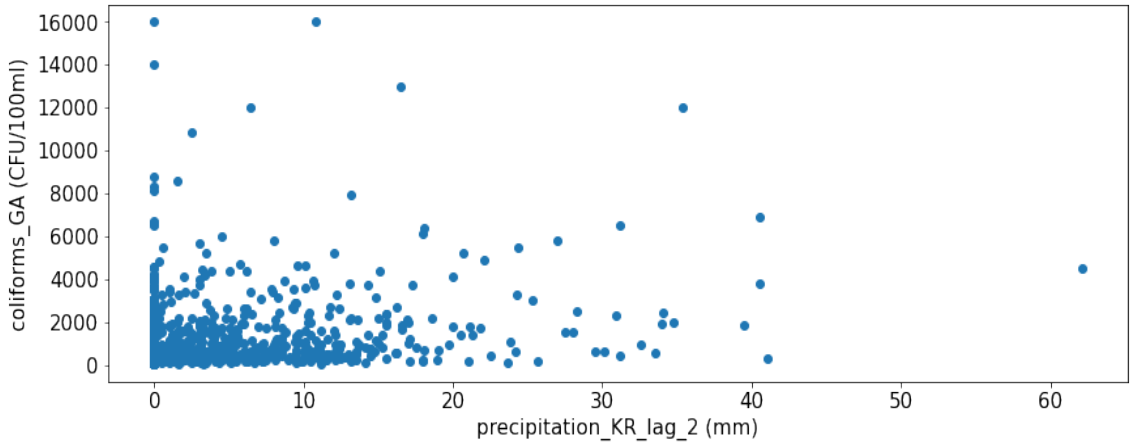


(b)

Figure A.2: *Relationship of precipitation and its 2-day lag values with different features. Figure (a) represents precipitation at Komperöd with coliforms levels at Lärjeholm, Figure (b) represents 2-lag observations of precipitation at Komperöd with coliforms at Lärjeholm.*

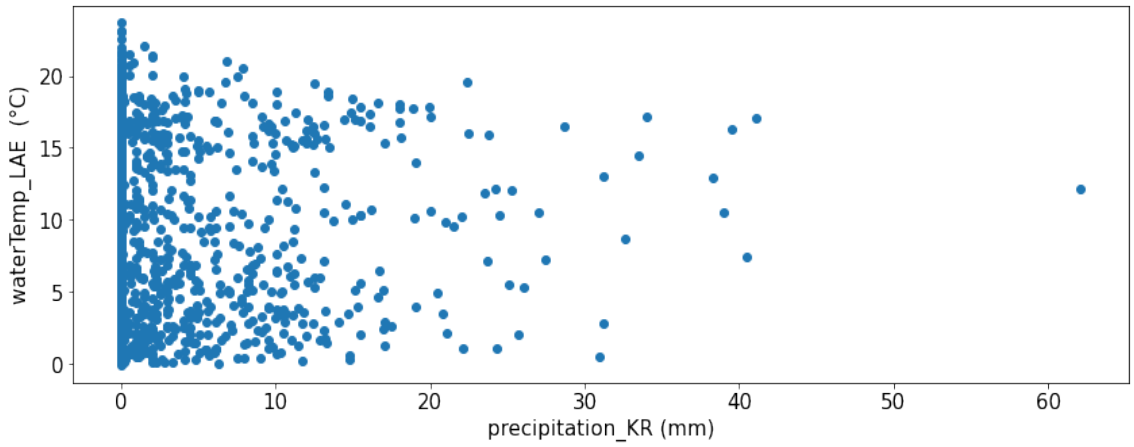


(a)

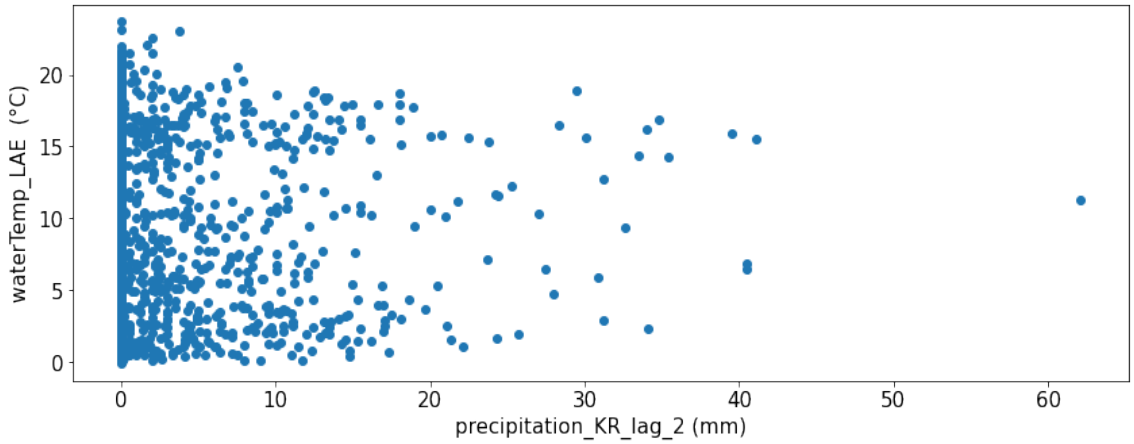


(b)

Figure A.3: *Relationship of precipitation and its 2-day lag values with different features. Figure (a) represents precipitation at Komperöd with coliforms levels at Garn, Figure (b) represents 2-lag observations of precipitation at Komperöd with coliforms at Garn.*

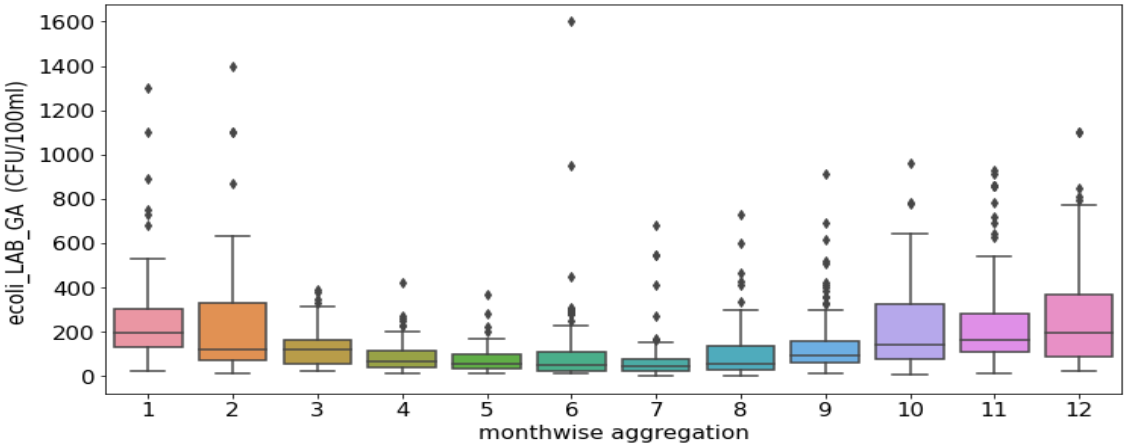


(a)

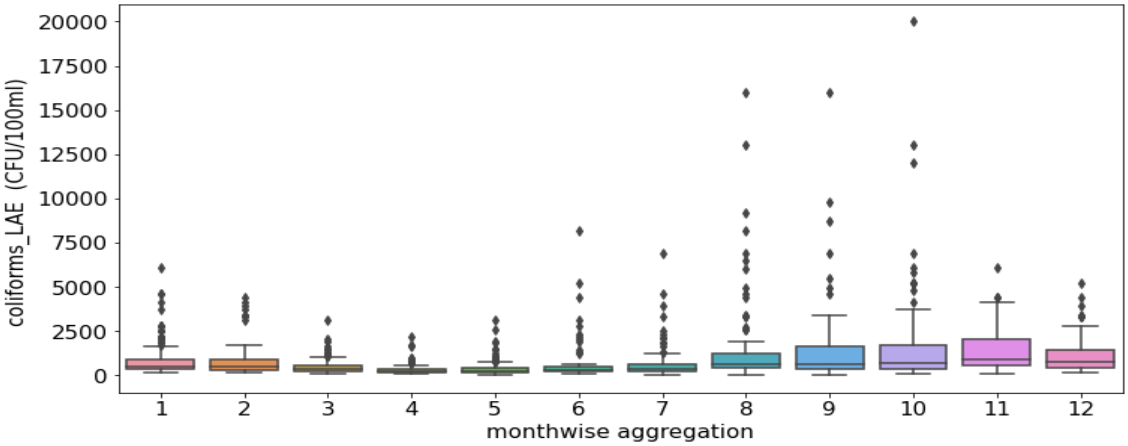


(b)

Figure A.4: *Relationship of precipitation and its 2-day lag values with different features. Figure (a) represents precipitation at Komperöd with water temperature levels at Lärjeholm, Figure (b) represents 2-lag observations of precipitation at Komperöd with water temperature at Lärjeholm.*

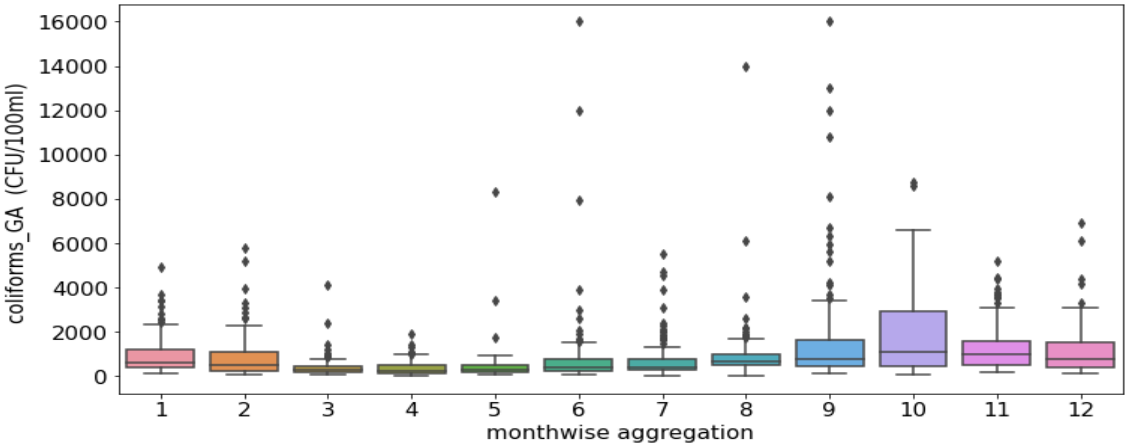


(a)

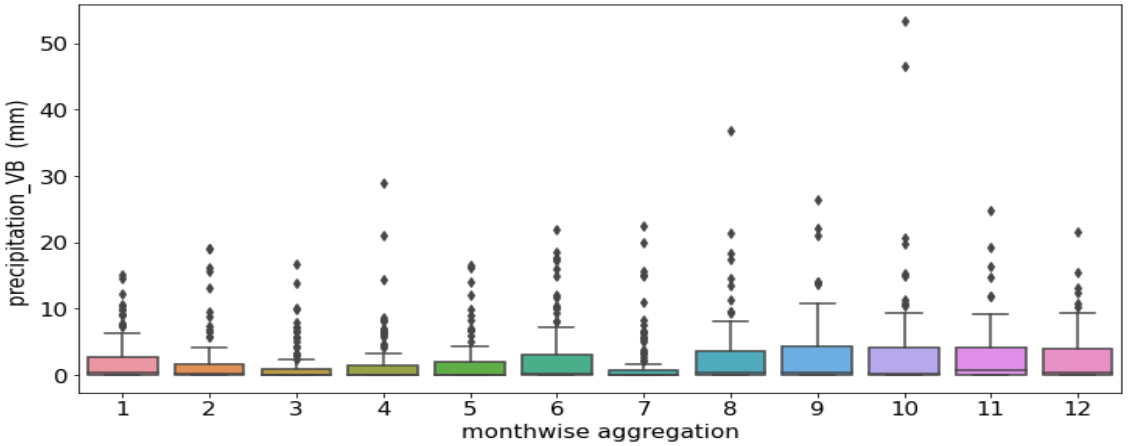


(b)

Figure A.5: *Box plot analysis of E. coli at Garn (Figure a) and coliforms at Lärjeholm (Figure b).*



(a)

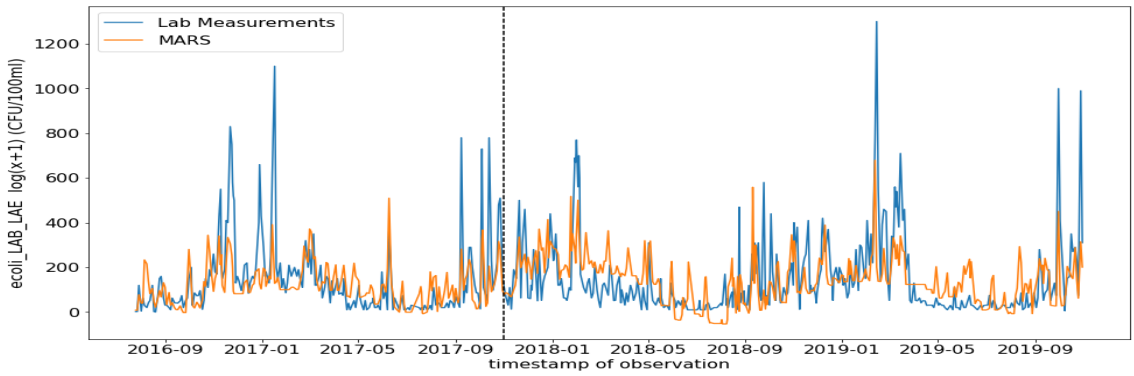


(b)

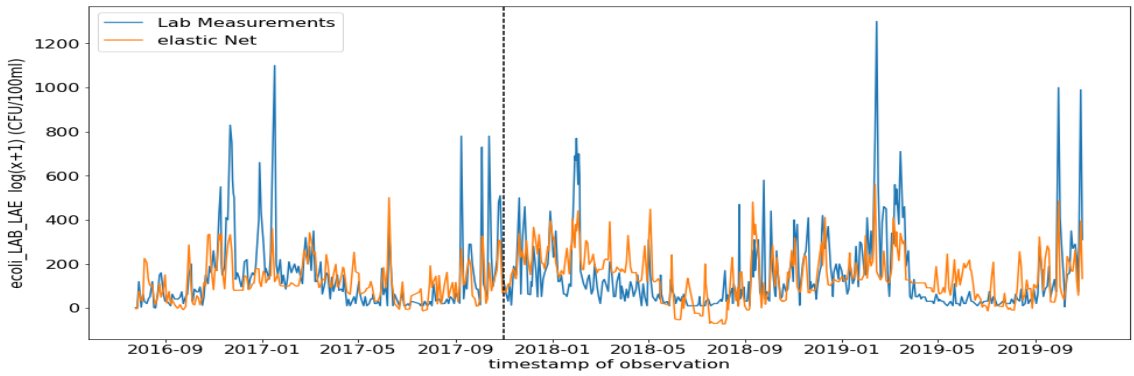
Figure A.6: *Box plot analysis of coliforms at Garn (Figure a) and precipitation at Vänersborg (Figure b).*

B

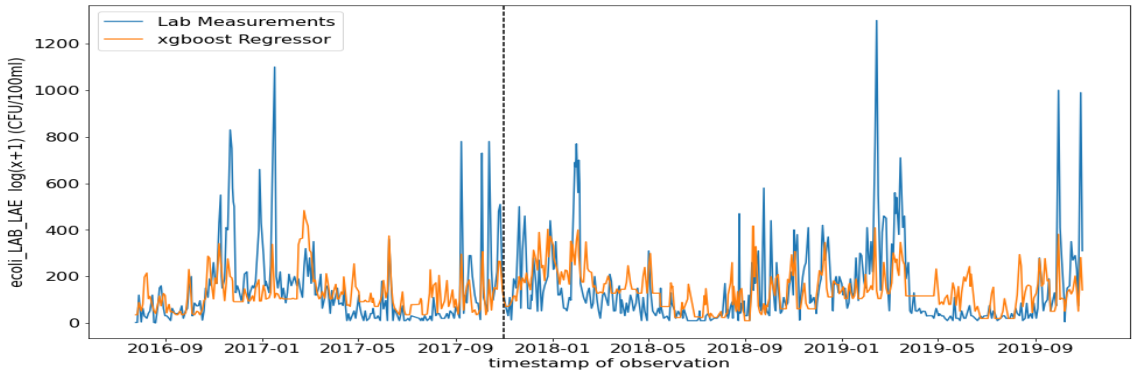
Appendix



(a)

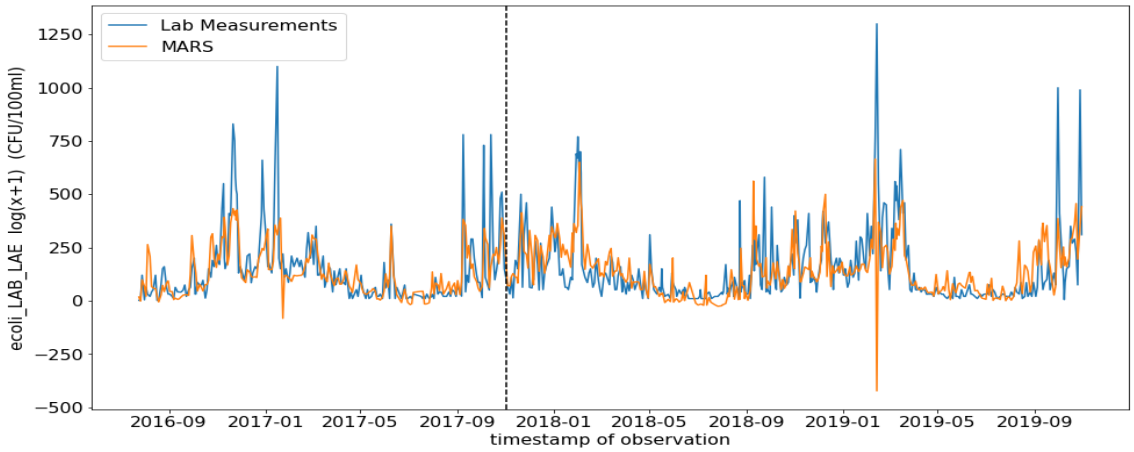


(b)

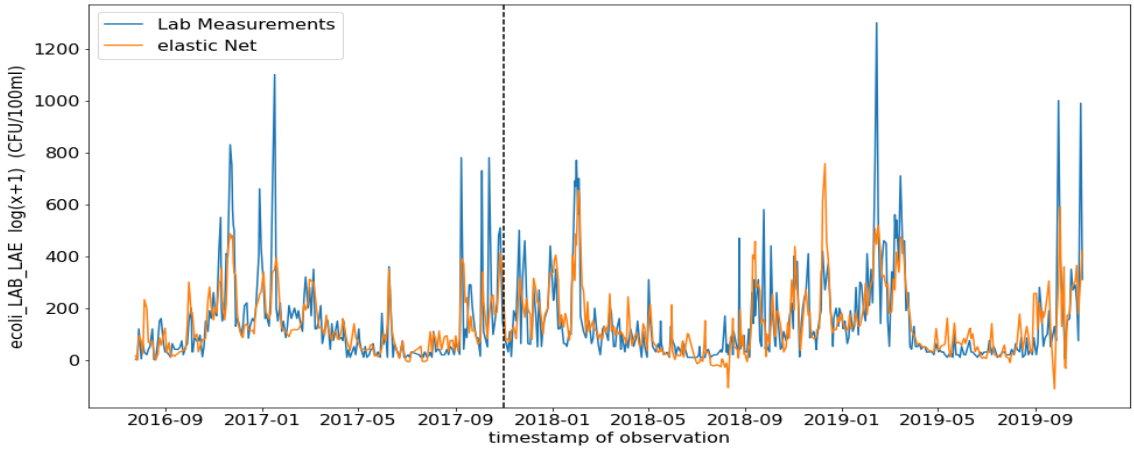


(c)

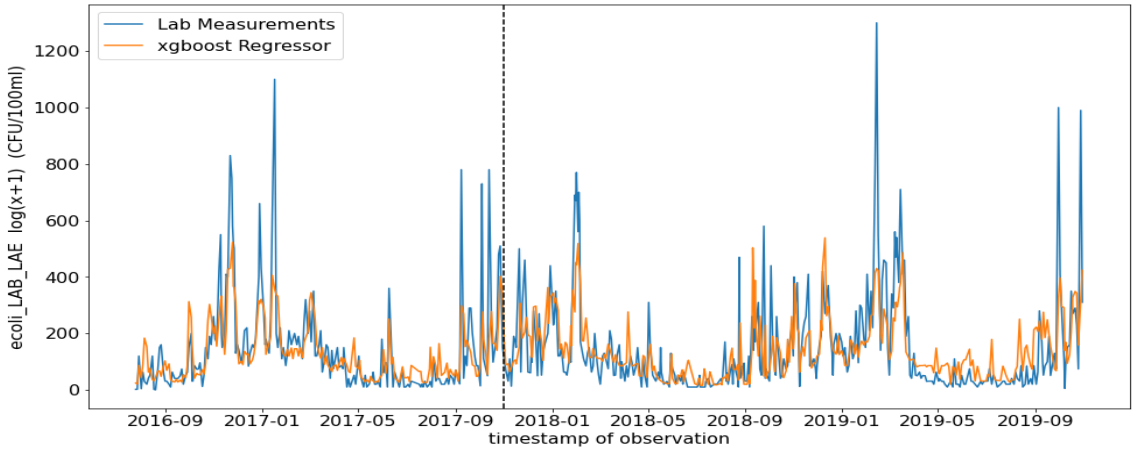
Figure B.1: *Performance plots of MARS (Figure a), Elasticnet regression (Figure b), XGBoost regression (Figure c) on baseline dataset (without $log_{10}(x+1)$ transformation).*



(a)



(b)



(c)

Figure B.2: *Performance plots of MARS (Figure a), Elasticnet regression (Figure b), XGBoost regression (Figure c) on complex dataset (without* $log_{10}(x+1)$ *transformation).*

Department of Architecture and Civil Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY