Soft Sensor Modelling with Multivariate Adaptive Regression Splines and M5P tree in Fermentation Process of Methanotrophs

Student Name: Peian Lu

Supervisor Name: Dr.Rui Carvalho

Submitted as part of the degree of M.Sc. MISCADA to the

Board of Examiners in the Department of Computer Sciences, Durham University

Sep 2th

Abstract —

The fermentation process of methanotrophs plays an important role in alleviating the problem of global food resources. Soft sensor modelling is garnering more attention in dealing with time-varying, nonlinearity, strong interaction, and complex mechanisms of the fermentation process under the data-driven context. This paper proposes an approach based on integrated learners to calculate feature importance and remove predictors with high collinearity through forward selection, using data from one batch of fermentation in Calysta. Multivariate Adaptive Regression Splines (MARS) and the M5P tree are further implemented to create interpretable models with exceptional precision. The models identified a multi-stage influence of predictors on the dependent variable acetic acid, and oxygen flow at fermentor gas station 2 and NH4.Productivity were considered essential.

Keywords —

Soft Sensor Techniques, Fermentation Process Modeling, Calysta, Methanotrophs, Feature Selection, Severe Multicollinearity, MARS and M5P

I INTRODUCTION

Population and consumer growth have placed agriculture and natural resources under unprecedented pressure while available land for food production is running out. Over the past few decades, the feed industry has prioritised the production of alternative protein sources (Biswas et al. 2020). Single-cell proteins (SCP) generated by microbes have been demonstrated to be safe and nontoxic in studies. It's high in protein, amino acids, and vitamins. It can be fed to cattle and poultry to boost output, enhance feed utilisation, and substitute supplementary protein feed like fish meal, meat, and skimmed milk powder. The feed industry has focused on decreasing and removing fish meal protein over the last few decades (Liu et al. 2013) and has successfully produced alternative protein sources (Biswas et al. 2020). Methanotrophs employ inexpensive carbon sources such as methane and methanol to synthesis their own biomass via cellular action; hence, the fermentation process created by methanotrophic bacteria has becoming source of alternative source of alternative protein and gained increasing interest. Over the last few decades, soft sensors have established as a useful alternative to conventional techniques for acquiring important process variables, process monitoring, and other process control-related tasks. This paper examines fermentation data from a period of 900 hours provided by Calsta.

Calysta created FeedKind Protein to resolve these food source challenges and give protein-rich alternatives to several conventional fish feed additives. FeedKind protein requires the least amount of land and water compared to other land-based fish feed ingredients. As a result, FeedKind protein has a significant advantage since land and water will become increasingly valuable resources in the future as the global population increases. Calysta partnered with the Centre for Process Innovation (CPI) to construct the Teesside Pilot Plant (TPP) in 2016. The TPP is a 6,000-litre loop fermenter that primarily performs methanotrophic fermentation, with gas and nutrients provided through additional ports around the loop to ensure equitable resource distribution to the culture. This paper is motivated by the observation that the production of organic acids can have a significant impact on the final SCP output. Hence, it is critical to forecast the creation of organic acids and identify and control the related variables to reduce risk costs and speed up production. Time is money when it comes to the large-scale industry. Furthermore, this paper presents some tentative solutions based on the multicollinearity of variables in fermentation modelling that have not been thoroughly investigated in the academic field, as well as interpretable models to characterise the potential influencing factors and hence improve the control of various parameters in the fermentation process.

II RELATED WORK

The establishment of a reasonable mathematical model is the basis for the optimization of the fermentation process. In general, the mathematical model captures the link between variables (strain concentration, medium composition and concentration, ambient conditions, pH, temperature, dissolved oxygen, etc.) and outputs (biomass, product concentration, etc.) in the fermentation process. The model can be utilised to enhance the process's control in order to increase production and efficiency. Currently, mathematical models can be divided into two categories based on their construction: mechanism models (known as ”white boxes”) and data-driven models (the ”black boxes”)

A Mechanism Modelling

Beginning with the mechanism of the process, the model is developed using the principles of enzyme kinetics, fermentation kinetics, biochemical reaction engineering, and material balance based on the metabolic network. This model incorporates all of the biochemical reaction networks, as well as the transfer, diffusion, and absorption of dissolved oxygen and substrates in the fermentation broth and cells, allowing it to correctly and exactly depict the fermentation process. Yuan conducted research on the problem of online estimation of the maximum growth rate of online autotrophic biomass (Yuan et al. 1999), using a simple ammonia synthesis process model as the basis for creating an aluminium foil estimation algorithm. In this research, a reaction rate equation of the Monod type was used to predict biomass concentration and ammonium salt concentration. In creating a lysine fermentation process using Corynebacterium glutamicum, a method for parameter estimation (Takiguchi et al. 1997) was suggested using two relaxation variables and Extended Kalman Filter (EKF). The model was also constructed by analysing the ATP/ADP metabolic pathway, and its rate vectors were estimated using least-squares regression (LS).

B Data-Driven Modelling

The data-driven model is a model built by utilising mathematical statistical regression and other approaches to determine the fermentation law without knowing or taking into account the process mechanism, based on field experience and a large number of fermentation batch data. This model depicts an apparent dynamic property of pure data between state variables and modified variables. Due to the fact that this mathematical model does not account for the fermentation process's actual reaction mechanism and reaction mechanism. Consequently, this model lacks apparent biological and chemical significance.

Cimander utilised ANN to estimate lactose, galactose, lactate, and pH during yoghurt fermentation (Cimander et al. 2002). For better feature combination, the ten output signal from gas sensors was reduced to 6 using principal component analysis (PCA), and the Quasi-Newton method was employed to accelerate network learning convergence. A study of support vector machines (SVM) and radial basis neural networks (Desai et al. 2006) showed that SVM perform comparably to neural networks when dealing with extremely nonlinear fermentation processes. An approach that combines multi-criteria and Gaussian process regression to optimise the erythromycin fermentation process was proposed in (**?**). The strategy incorporated multiple models with definite order requirements, and the appropriate application of a Gaussian process allowed the model to predict biomass concentration with high precision.

C Discussion and limitation

The fermentation process is characterised by its high dimensionality, complex nonlinearity, and hysteretic behaviour. The traditional mechanistic model can provide an intuitive explanation; however, as it is founded on a large number of assumptions, it results in a poor model fit and low generalisation capacity. Improvements in manufacturing machinery have paved the way for the spread of data-driven methods. Equally challenging, though, is figuring out how to better regulate manufacturing conditions and making sense of the constructed model. Prior research, such as those in (Thibault et al. 2000), have utilised a series-type mixed model to represent the dynamics of biological reaction processes by integrating mechanistic differential equations and a neural network, a hybrid modelling technique is a viable option. For large numbers of predictors and cyclic fermentation processes, however, hybrid models may not adequately capture the entire reaction. Consequently,

the purpose of this paper is to deliver credible and meaningful conclusions while developing a predictive model. Three more aspects are rarely discussed in academic literature. 1: How to perform feature selection in high dimensions with existence of severe multicollinearity? This paper discusses complex situations with 65 predictors, while earlier studies have been focused on fermentation processes with only 5–15 predictors. 2: Existing research typically investigates a limited sample size in which predictors and target variables match. However, in the industrial process, data collection is primarily separated into offline and online sampling. The mismatch in sample distribution renders the conventional imputation methods unworkable, hence future study should investigate ways to generate data that can be used for supervised learning. 3: Based on the timestamp mismatch between the dependent variable X and the independent variable Y, how should the error be measured? Jian rong used interaction to match samples in the modelling of glutamate fermentation process (Zheng 2020). This is the approach referenced in this study, however future research is required to determine how to measure the error of the simulated target variable based on interpolation estimation.
This paper next presents some tentative approaches for the previously unaddressed difficulties stated above.

III SOLUTIONS

Data was collected from TPP-19 that primarily performs methanotrophic fermentation in loop, with additional ports throughout the loop supplying gas and nutrients to ensure equitable resource distribution to the culture. Predictors have local variables scattered around a fermenter and global variables that capture global environmental values.

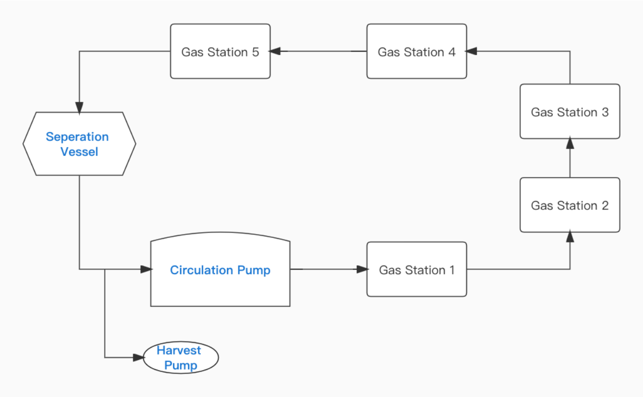


Figure 1: Schematic diagram of the fermentation process of TPP

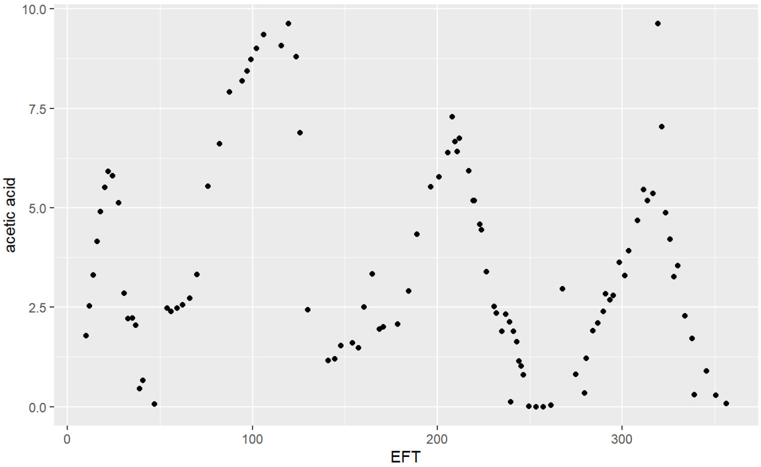


Figure 2: Target variable within 360 hours

A Data Description

The fermentation process operates in a cyclic serial mode, as depicted in Figure 1. Soft sensors are installed in various locations to gather data on various biochemical parameters. The collected data consists primarily of online data and offline data with a high dimension of 65. Online data collection is used for metrics that are easily measured and do not require high precision. The online data includes environmental parameters that measure the overall fermentation reaction, such as harvest flow, percentage of oxygen and carbon dioxide in offgas, etc.; it also includes the measurement of local variables that have a significant impact on the fermentation of a single fermenter and must be monitored separately, such as pH value, temperature, continuous added flow at each station of oxygen, methane and amount of dissolved oxygen along the loop, etc.. The offline data are sampled manually in the laboratory, including the optical density and dry cell weight reflecting the number of cells, the productivity of the cells and the content of acetic acid that is of interest.

B Predictors Definition

Predictors measure primarily the nutrients necessary for cell growth. Methane is the energy source for increasing Metantrophs; oxygen and dissolved oxygen are also required and must be carefully regulated to prevent cell damage. Methane and oxygen are flammable and explosive gases, hence a LEL variable is used to assess the level of explosive gases in the system. The flow rate of phosphoric acid is used to regulate the pH of the system; the flow rate of trace elements gives the trace quantity required for optimal cell growth and protein production; and the flow rate of ferrous sulphate records iron, which and stimulates methane uptake by cells. For simplicity, all global variables are abbreviated to their first words (such as LEL and $CO_2.Offgas$), and local variables are added "stn.X" after the variable name to indicate the information of the X^{th} station, as $pH.stn1$ and $oxygen.stn2$. In addition, the oxygen-to-methane ratio is regarded an important growth-controlling parameter and is therefore recalculated and included in the data under the symbol $Ratio.stnX"$.
Online data is automatically collected every half hour, while offline data (as shown in figure 2) can only be collected irregularly every two to three hours due to labor costs. During a 900-hour fermentation process, there are approximately 2100 rows of predictors X and 110 values of target Y (because the production of acetic acid is only concentrated in the first 360 hours). Therefore, we may confront a dilemma that: suppose we only have a row of data X at time t=10 hours, while the most recently measured target variable Y is at time t=10.5 hours. Therefore, this paper seeks to employ interpolation to compensate for the imbalance and mismatch in the data distribution on an industrial scale.

C Data Merging

C.1 Challenges

Given that the ratio of sample sizes between the two datasets is approximately 20 to 1, the conventional imputation method employing KNN, random forest, etc., would result in significant inaccuracy due to an excessive number of missing values. In addition, the time series methodology cannot be applied due to the different sample times of the two datasets. Smoothing and interpolation are common non techniques that require only the sequence of target variables to produce an estimated curve through time. To aggregate the two datasets, we can utilise the estimated curve from the offline target acid to derive the value at a specific time point (i.e., the timestamp of the online data). The smoothing method attempts to capture the overall trend, but information about specific inflection points is lost. Changes in biochemical parameters during the fermentation process are more indicative of the special events occurring at this time and should cause us concern. Consequently, interpolation is applied in this paper.

D Interpolation and Cubic Spline Interpolation

Interpolation is the estimation of constructing new data points based on the range of a discrete set of known data points; however, unlike smoothing, it is guaranteed to pass all provided points. Information that accurately reflects the trend of acetic acid at different time points is retained as much as feasible from this standpoint.

This has the advantage that no small variations in the fermentation parameters are lost. It is often the unusual variations that are more worthy of our attention. Meanwhile, the readings from soft sensors are accompanied by noise and are retained in the same manner during interpolation. To offset this effect and preserve as much of the variable's original trend as possible, sliding window-based noise reduction technique is employed and will be discussed in later section. Common interpolations are linear interpolation, polynomial interpolation and spline interpolation.
1. Linear Interpolation: Linear interpolation is the straight line fit between two known points (x_0, y_0) and (x_1, y_1). For any interested point $x \in (x_0, x_1)$, the estimated value y is calculated by:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0}$$

It is proven that the error of linear interpolation of a function **f** has a continuous second derivative, denoted by R is bounded by:

$$|R| \leq \frac{(x_1 - x_0)^2}{8} \times max_{x_0 \leq x \leq x_1} |f^{''}(x)|$$

As acid production has nonlinearity and abrupt variation, the rate of change of acid can be very large and unstable, making linear interpolation unsuitable.
2. Polynomial Interpolation: The mathematical forum of Polynomial interpolation is given by:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_2 x^2 + a_1 x + a_0$$

for all given (x_i, y_i) where $i \in 0, 1, \cdots, n$. Solving **n** equations yields the exact expression of polynomial interpolation whereas the unknown values y of unknown timestamp $x_t \in (x_0, x_n)$ can be obtained directly. Although intuitively it is well understood and can approximate complex curves, the interpolation curve will not be smooth at the nodes. As the power of the polynomial **n** increases, the interpolation function oscillates extremely at the between the data points (known as the Runge's phenomenon).
3. Cubic Spline Interpolation Cubic spline interpolation is a method constructing spline where each sub-interval is a third-degree polynomial in form of:

$$f(x) = \begin{cases} a_1 x^3 + b_1 x^2 + c_1 x + d_1, & x \in [x_1, x_2] \\ a_2 x^3 + b_2 x^2 + c_2 x + d_2, & x \in [x_2, x_3] \\ \cdots & \\ a_n x^3 + b_n x^2 + c_n x + d_1, & x \in [x_n, x_{n+1} \end{cases}$$

Cubic spline interpolation uses the third degree of polynomials in each interval to make them fit smoothly. Compared to previous two methods, spline interpolation, like polynomial interpolation, has less error than linear interpolation and is smoother due to the property of first-order and second-order derivables; it is easier to evaluate than higher-order polynomials and is not affected by the Runge's phenomenon.

E Data Pre-processing

E.1 Near Zero Variance

The near-zero-variance predictor has a large ratio of samples of the most frequent value over the second most frequent value. In other words, this variable contains a large number of repeated or similar values, which is considered to be less informative. In ideal conditions, the ratio should be close to 1 to ensure a well-balanced distribution of samples. For highly unbalanced samples, this can lead to model collapse or unstable fits. Using this method, the predictor variable ”pump outlet for the tempered water system” was removed.

E.2 Data Cleaning

Some negative numbers are accidentally recorded due to sensor errors. However, as none of the parameters should be negative, we only truncate negative values at 0 rather than discarding the entire row of data at this timestamp. According to the recommendations of the Calysta engineers, when the values of all local sensors are below 0.05, it means that this fermentor station is in the off state. At this point the values recorded by the sensors are also just noise, so data less than 0.05 will be adjusted to 0. In addition, the oxygen-to-methane ratio is crucial for the growth of methanogenic bacteria, hence the oxygen-to-methane ratios from stations 1,2,3,4, and 5 are added as new predictors.

E.3 Elimination of flow totaliser

The online data totaliser is accountable for recording the accumulation of the material flow, such as the total amount of nitric acid, the flow of trace elements, etc. If these variables are included in the regression equation for acid, strong significance of their coefficients are presented. Initially, this was a highly rare phenomenon, as the accumulated amounts of different chemicals are steadily increasing overall. However, the significance of this totaliser diminishes when only one of the variables is preserved and the rest are omitted. Therefore, it can be concluded that they exist solely to cancel each other out. Since they do not add to the model's creation, retaining them would weaken the model's interpretability. The seven variables of fermentation duration, phosphoric acid totaliser, sodium hydroxide totaliser, etc., are removed.

E.4 Sliding Windows denoise

In the fermentation process industry there is a lot of noise in the data collected by the sensors, which can have a negative impact on the subsequent modelling. Due to the complexity of the chemical parameters of fermentation in real production and the fact that operators manually control the flow of individual substances, not all extreme values are outliers. For example, the addition of antifoam at some point has a cascading effect. If such data is smoothed directly, some useful information may be lost. On the other hand, short-term abnormal fluctuations caused by signal interference during transmission should also be excluded. However, it would be difficult to detect outliers with large fluctuations over short periods of time if we relied solely on the common outlier test. We, therefore, use absolute difference of two sequential data point as a measurement, and if the differences between the variables are too large, meaning that they change very dramatically and it is likely to be noise. We use the average weighted value of the sliding window to replace this outlier. This means that a variable can be preserved so long as its value does not change drastically, regardless of how large or little its value is. In the situation of data that fluctuates sporadically across a brief time interval, its information is replaced by samples from the surrounding time interval.

Algorithm 1 Sliding windows denosing by weighted average

```
Require: a multiplier: c;    input data: $X$
Initialise number of predictors: p;    size of data: n;    window size: k;
for $j \in 1, \cdots, p$ do
    Calculate the standard deviation $\sigma_j$ of $X_j$
    for $i \in k+1, \cdots, n-k-1$ do
        diff $\leftarrow x_{(i+1)j} - x_j$
        if diff $\geq \mathbf{c}\sigma_j$ then                          ▷ sign of noise
            Set weights vector W:
                $W \leftarrow (1, 2, \cdots, k, k-1, \cdots, 1)$
            Normalise the weights vector:
                $W \leftarrow \frac{W}{sum(W)}$
            Set sliding windows vector $X_{windows}$:
                $X_{windows} \leftarrow (x_{(i-k)j}, \cdots x_{(i-1)j}, x_{(i+2)j}, \cdots, x_{(i+k+1)j})$
                ▷ since $x_{ij}$ and $x_{(i+1)j}$ are considered as abnormal changes, they should be neglected
            Replace the value of $x_{ij}$ by:
                $x_{ij} = W^T X_{windows}$
        end if
    end for
end for
```

F Feature Selection

Common feature filtering methods include best subset selection, shrinkage methods with an application in (Xing et al. 2001) and dimension reduction methods with full details in (Fodor 2002), which are popular in statistics, as well as wrapper (Liu & Motoda 2012) and filter methods (Kim et al. 2002) in machine learning.

F.1 Best Subset

This approach involves identifying a subset of **p** predictor variables that we consider to be relevant to the response. We then construct a model(e.g. using least squares or building a decision tree) on the reduced set of variables . This is an exhaustive method of trying different combinations of variables and filtering out the best ones based on certain criteria such as adjusted R^2 and AIC. As the complexity of the method is 2^p, it is not applicable to our high-dimensional problem in terms of computational overhead. The large search space can lead to over-fitting and high variance in the coefficient estimates.

F.2 Stepwise Selection

Stepwise selection consists of forward selection, backward elimination and bidirectional elimination selection. The first approach is to add variables to the model step by step from an empty set, filtering each time for the variables that will most improve the performance of the model. The second is the opposite, considering all variables first and removing the ones that contribute least to the model's performance at each iteration. The third is a combination of the first two, adding variables sequentially as in forward selection, but allowing one variable to be removed after each iteration under certain conditions. Nevertheless, stepwise selection is ineffective against variables with multicollinearity. The greater the number of variables considered, the higher the chance of encountering coincidences in statistical tests in which some significant factors become insignificant while others become significant (Calude & Longo 2017).

F.3 Shrinkage Methods

Regularised coefficient estimation can also be used to fit a model with all **p** predictor variables or, equivalently, to reduce the coefficient estimates by an equal fraction or to zero. Ridge regression and Lasso regression are two common choices.

Ridge regression can be effective because it trades off a slight increase in bias for a large decrease in variance, which includes the following adjustments to the objective function in simple linear regression (SLR):

For SLR:

$$\min_{\beta} : \sum_{i=1}^{n}(y_0 - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 = RSS$$

For Ridge regression:

$$\min_{\beta,\lambda} : RSS + \lambda\sum_{j=1}^{p}\beta_j^2$$

It also has considerable computing advantages over optimal subset selection, which needs a 2^p model search. This search is computationally infeasible even for tiny values of p. Unlike best subset, forward stepwise selection, and backward stepwise selection (which involve only a subset of variables in the final model), ridge regression includes all p predictor variables in the final model. When the number of variables p is enormous, model interpretation becomes difficult. As a result, strict variable selection is not achievable, although it is possible to limit the impact of inconsequential factors. Ridge regression coefficients can be used as weights for each variable and then coupled with other methods for variable selection.

To compensate for the inadequacies of ridge regression, **Lasso** regression applies an l_1 norm of penalty to truncate the unimportant coefficients at 0, which has the form of:

$$\min_{\beta,\lambda} : RSS + \lambda \sum_{j=1}^{p} |\beta_j|$$

Cross-validation is a straightforward approach to find the regularisation term. By setting a grid of λ values, cross-validation errors for each λ can be computed, and then the tuning parameter λ is selected for which provides the smallest CV errors. Finally, the model is re-fit using all of the available observations and the selected value of the tuning parameter.

However, neither strategy helps in determining an appropriate penalty term. In CV, the best λ converges to zero (depending on the smallest unit of the search grid selected in advance), implying that the degree of contribution of any variable should not be weakened or underestimated. In addition, this is irrational because the data contains several variables of equivalent relevance. The presence of highly severe multicollinearity may result in a non-smooth performance of the shrinkage technique. Alternatively, due to the non-linear, non-time-sensitive nature of the fermentation process, linear regression models are insufficient to capture variations in the target variable acetic acid.

F.4 Dimension Reduction Methods

A dimensionality reduction method is one that investigates a set of transformations on the predictor variables and then fits least squares to the transformed features.

Let $Z_1, Z_2, \cdots, Z_M$ represent lineaer combinations of original p predictors where $M < p$.

$$Z_m = \sum_{j=1}^{p} \phi jmXj$$

for some combination constants$\phi_{1m}, \phi_{2m}, \cdots, \phi_{pm}$. Then the linear regression model is fitted by least squares method:

$$y_i = \beta_0 + \sum_{m=1}^{M} \beta_m Z_{im}, \qquad i = 1, 2, \cdots, n$$

For **Principal Component Regression**, the predictors are replaced by the first M principal components, $Z_1, \cdots, Z_M$. PCA ensures that the different individual dimensions of the data are linearly uncorrelated, which overcomes the problem of multicollinearity. Simultaneously, PCA maximises the variance of the new predictors, keeping more information from the original data. Another popular dimensionality reduction method is **Partial Least Squares Regression**, which is a supervised variant of principal component analysis. The difference is that instead of finding the hyperplane of maximum variance among the independent variables, it constructs a linear regression model by projecting the predictors and target variable into a new space. The PLS model attempts to discover the multidimensional direction in X-space that explains the most variance in Y-space. That is, the components generated from covariance-based PLS regressions are designed to explain as much of Y as possible. Since PLSR considers the response variable, it enables the model to fit response variables with fewer components in most circumstances.

Because of the project's strategic aims, interpretable models are critical. The large dimension of the variables makes intuitive interpretation and transformation of the reduced predictors difficult. Although the dimensionality reduction approach is not the final model, it is employed in the subsequent variable selection section because of its outstanding property of eliminating multicollinearity.

F.5 Filter and Wrapper Approaches

Filter selection is one of the most basic approaches, and its main benefit is that it does not rely on the model and merely investigates the worth of the features from the standpoint of feature ranking. It examines all predictors using simple univariate statistical procedures (such as pairwise correlation, Euclidean distance, and maximum information coefficients), then uses only those that pass some criterion in subsequent model stages. The **wrapper** approach measures the importance of a subset of features by training the model. Using the model's own features, it calculates the degree of contribution of the variables through the CV. The absolute value of the t-statistic of the estimated coefficients is the criteria of importance in linear regression. The random forest criteria is the accuracy of the out-of-bag prediction. The weighted sum of the absolute regression coefficients is the criterion in PLS. More criteria can be obtained from the **VarImp** function in the **caret** package.

The filter method has computational advantages, but it has a tendency to over-select features when highly correlated predictor variables are available. Furthermore, it evaluates each predictor separately and does not support extracting and retaining combinations of several variables. The wrapper technique finds a better collection of variables, but the model expression is constrained by its own applicability. Different wrapper learners could produce different feature selections, resulting in inconsistency.

F.6 Ensembled wrappers through by forward selection

To overcome the limits imposed by individual models, this paper proposes an ensembled greedy-wrapper strategy that functions as a feature selection by leveraging multicollinearity as determined by the variance inflation factor (VIF). VIF is calculated as follows:

$$VIF(X_j) = \frac{1}{1 - R^2_{-j}}, \quad j \in (1, \cdots, p)$$

where R^2_{-j} is the unadjusted R squares determined by regressing the j^{th} independent variable on the remaining predictors.

The VIF method investigates not only the pairwise correlation between variables, but also the relationship between numerous variables. This method is more likely to collect information on factors that are not directly associated, but when combined with other variables, it can establish a significant relationship. Many sensors are installed at different fermentation stations in the TPP-19 fermentation process to record the same parameters, such as dissolved oxygen flow, methane flow, and so forth. Cycling the fermenters promotes nutrient utilisation while also causing extreme multicollinearity. The multicollinearity may be related to delays in the delivery of substances or to the comparable properties of variables themselves. This additional information may be utilised to create segmented models using mechanistic models. In this research, however, data-driven modelling is utilised; therefore it is only essential to preserve as many non-repetitive and representative parameters as possible. Variables having a VIF greater than a given threshold are targeted, similar to filter approaches. However, the challenge is which variables to delete in order to lose the least amount of information. As a result, this work presents a variable importance measurement based on several learners weighted by their performance in the training set to choose which variables to keep and which to reject. To reiterate, the importance of the variables in the different models is calculated according to the **VarImp** function of the 'caret' package.

Algorithm 2 Variables Selection

```
Require: Data: $X$ ; Learners list: $F$; Tolerance of deletion: Tol; Acceptance of addition: $Acc$
Initialise number of predictors: p;   size of data: n;   number of learners: M;
training set and test set
for $m \in 1, \cdots, M$ do
    Initialise Leaner $F_m$ (train the model)
    Obtain the variables importance vector $W_m$ of $m^{th}$ leaner $F_m$:
        $W_m \leftarrow (w_{m1}, \cdots, w_{mp})$
    Calculate the performance $\epsilon_m$ of leaner $F_m$ on the test set  ▷ Root mean square error is adapted as the performance criterion in this paper
end for
Calculate the weighted importance vector of p variables:
    $W = \epsilon W_m$
        ▷ The $\epsilon$ here can be treated as the weight and the $W_m$ should be coefficients
```

```
Sort the **weighted importance** vector W in decreasing order
Generate a variable list $Pnames$ containing p predictors' names, and sort it by the same order of the weighted importance vector W
Initialise a candidate list $Cnames$:
    $$Cnames \leftarrow Pnames[1:2]$$
                                        ▷ For VIF calculation, at least three variables are needed
**for** $j \in (3, \cdots, p)$ **do**
    Add variable $Pnames[j]$ into $Cnames$
        $$Cnames \leftarrow (Cnames, Pnames[j])$$
    Construct linear regression by regressing one predictor against the remaining predictors of a sub-data subtracting columns of data $X$ corresponding to the $Cnames$
    Calculate the VIF values of each variable in the $Cnames$
                                        ▷ The VIF value can be computed by **vif** function in 'car' package
    Denote the VIF value of a predictor $a$ by $V_{\mathbf{a}}$
    **for** $c \in Cnames$ **do**
        **if** $V_c \leq Acc$ **then**
                                        ▷ adding this variable doesn't bring the multicollinearity
            **Continue**
        **else if** $V_c \geq$ Tol
            Remove the added variable $Pnames[j]$ from $Cnames$
                                        ▷ adding this variable causes the multicollinearity
            **Break**
        **else**
            Make decisions based on other criterion and background knowledge
        **end if**
    **end for**
**end for**
**Output**: the selected predictors list $Cnames$
```

This algorithm implements a forward selection. In essence, the variables introduced at each iteration are less significant than those already present on the candidate list. Therefore, the only decision to be made is whether or not to retain this newly introduced variable. At each iteration, the variable of lower importance is included, and its effect on multicollinearity is evaluated. In general, the outcomes of such proposed combinations cannot be predicted in advance. For instance, when the $Ratio.stn2$, $oxygen.stn2$, and $oxygen.stn3$ are included in the candidate list, the addition of $methane.stn3$ significantly increases the VIF value of $oxygen$. Given the information from the previous stations, it is suggested that two distinct types of nutrient flow at station 3 can represent each other. This conclusion, which would have been difficult to anticipate in advance, indicates the viability of the proposed strategy. Another scenario that is difficult to foresee in advance is discovered during the forward process. During the forward process, it may be observed that a newly introduced variable with a low VIF value raises the VIF of another variable significantly. This is a scenario that is hard to anticipate in advance. When the $dissolved.stn1$ was added, the VIF of the $Ratio.stn2$ was increased by 8; however, $dissolved.stn1$ itself only had a VIF of 3. This gives more evidence that the changes to the candidate list resulting from the selection of various variables at each iteration have a considerable impact on subsequent additions.

Eventually, the number of predictors is dropped from 65 to 45.

IV METHODOLOGY

A *Multivariate Adaptive Regression Splines (MARS)*

MARS is an adaptive regression method well-suited for high-dimensional applications (i.e., with a large number of input predictors). It can be viewed from two angles: first, as a generalisation of stepwise linear regression,

and second, as an improvement to the performance of CART in regression tasks. MARS differs from previous approaches in that it divides the space of each explanatory variable into distinct sub-regions and constructs a hinge function for each sub-region.
MARS performs the following linear basis expansion for each variable for piece-wise division:

$$h(x-t) = (x-t)_+ = \begin{cases} x-t, & if \quad x > t \\ 0, & if \quad x \leq t \end{cases}$$

$$h(t-x) = (t-x)_+ = \begin{cases} t-x, & if \quad x < t \\ 0, & if \quad x \geq t \end{cases}$$

The MARS algorithm consists of three stages. The first phase is forward fitting the model followed by the gradual insertion of new spline functions, in which the input data is divided into knots, and each division interval is fitted with a hinge function to produce a new basis function. Typically, the forward process generates several basis functions, resulting in an overfitting model.Therefore, the second step is a backward pruning procedure using generalised cross-validations (GCV), where the cross-validation parameters are the modified sum of squared residuals, which includes a penalty for model complexity. GCV is given as followed:

$$GCV(M) = \frac{1}{n} \frac{\sum_{b=1}^{M}(y_i - \hat{y}_i)^2}{(1 - \frac{C(M)}{n})^2}$$

$C(M)$ is the penalty term for complexity defined as:

$$C(M) = trace(B(B^T B)^{-}1B^T) + 1 + dM$$

where M is the number of basis function, B is a $M \times N$ matrix, $trace(B(B^T B)^{-}1B^T) + 1$ is the number of effective coefficient in the model, which is identical to M generally, d is the model's penalty factor, which is chosen to be 3 in this study.
This criterion restricts the number of spline functions included in the final model. The last step is choosing a model according to the GCV performance. The backward pruning procedure is used to eliminate duplicate basis functions and to assure the correctness of the model. Ultimately, the model with the highest test accuracy or other criterion is chosen as the final output model. The final expression of the MARS model can be:

$$Y = \beta_0 + \begin{cases} \beta_{10}h(X_1 - t_1) \\ \beta_{11}h(t_1 - X_1) \end{cases} + \begin{matrix} \beta_{20}h(X_2 - t_2) \\ \beta_{21}h(t_2 - X_2) \end{matrix} + \cdots + \begin{matrix} \beta_{p0}h(X_p - t_p) \\ \beta_{p1}h(t_p - X_p) \end{matrix} + \begin{matrix} \beta'_{p0}h(X_p - t_p) \times h(X_k - t_k) \\ \beta'_{p1}h(t_p - X_p) \times h(t_k - X_k) \end{matrix} + \cdots$$

where βs are not 0 if the terms are included in the model, and the higher order interaction are possible.

B M5P Regression Tree

The M5P algorithm was developed by Yong and Ian based on the M5 method found by Quinlan. Similar to a decision tree, the expansion of the tree is determined at each tree node based on the attribute values of the samples until it divides into a leaf node depending on some criterion. The distinction is that the terminal leaves of the tree contain linear regression models in M5P. In each iteration, the attribute with the largest Standard Deviation Reduction (SDR) is elected as the node's splitting attribute.

$$SDR = sd(S) - \sum_i \frac{|S_i|}{|S|} sd(S_i)$$

where S is the subset of data arriving at the node, S_i is the subset of nodes partitioned based on the potential attribute, and sd is the standard deviation of target value. After constructing the linear model, pruning methods were used to prevent overfitting. In order to compensate for the underestimating of the predicted error in the unseen dataset, a modified estimator is employed during pruning:

$$\delta_{error} = \frac{n+v}{n-v} \frac{\sum_i^n |V_i - \hat{V}_i|}{n}$$

where n is training samples of current node, v is number of parameters in the regression model. In a pruned tree, sharp discontinuities inevitably arise between the linear models of two adjacent leaf nodes, and to compensate for this, smoothing is performed in the final step. Smoothing is more effective for prediction than directly calculating predicted values, as given below.:

$$p' = \frac{np + kp}{n + k}$$

where p' is the predicted value passed to the upper node, p is the predicted value passed from the lower node, q is the prediction at this current node, n is the sample size of lower node and k is a smoothing constant.
MARS and M5P trees are called through the existing **earth** package and **cubist** package in R language. The cubist package improves the performance of the M5P tree by combining predictions from multiple-committee and K-nearest neighbours. However, as this is only a correction to the prediction results and has no effect on the relationship between the predictor variables and the target variables, the new technique is not utilised when constructing interpretable models. The final prediction models will be built by ensemble version of both MARS and M5P tree.

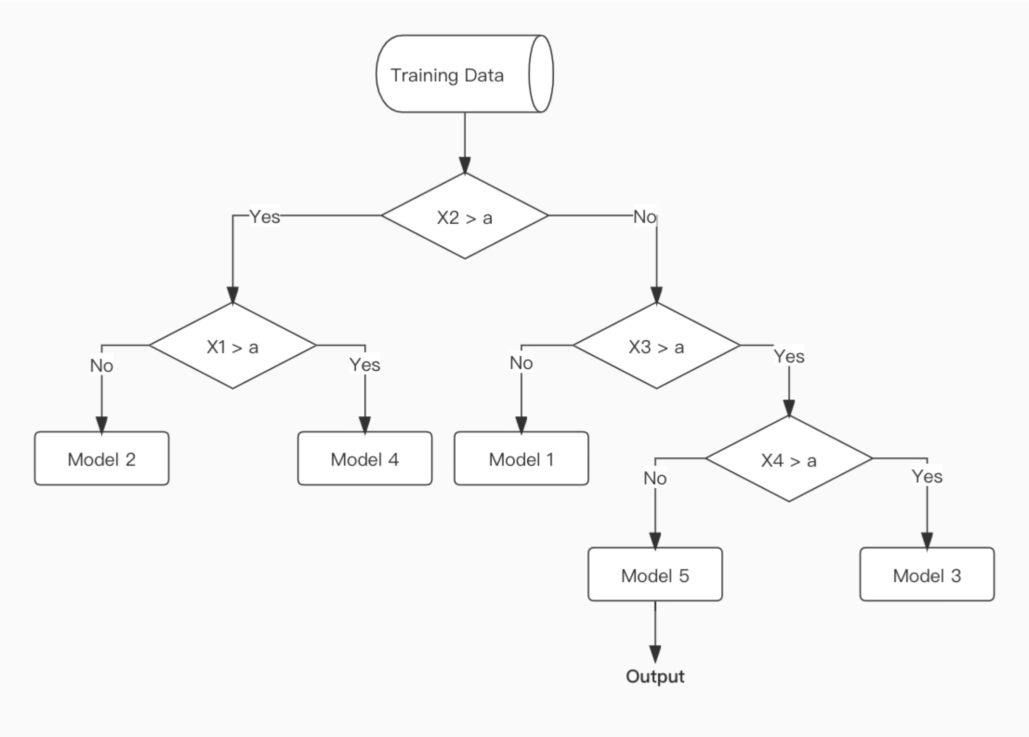


Figure 3: M5P prediction model

C Learners Settings

In principle, all models that predict continuous variables and can be utilised in regression problems can be accepted, as the importance of variables supplied back by different models is dependent on the model's fit to the TPP-19 dataset. For inadequate models, the significance of variables decreases. Some learners have a propensity for capturing linear correlations, some for building tree models, and others for transforming predictors into different dimensions. In the presence of strong multicollinearity, the algorithm presented in this paper is essentially a weighted technique. We hope that combining many learners will reduce the randomness of the estimations of variable importance, e.g., to avoid the statistical coincidences discussed previously.
All hyperparameters are obtained via CV tuning using **train** function in the caret package'. In order to boost computational efficiency, a search for random hyperparameter combinations (Bergstra & Bengio 2012) is launched and pre-processed with the same random seeds to assure reproducibility.
The importance of the variables is evaluated by training multiple learners provided by the "caret" package using a 10-fold CV, and the performance of the model is measured by the accuracy of the test set.

Linear Regression	Gaussian Process Kernel Regression	M5P tree
Regression Tree	xgb Linear Tree	xgb Tree
k-Nearest Neighbors	Bagged adBoost Tree	MARS
Partial Least Squares Regression	Principal Component Regression	Random Forest
SVM with Radial Kernel	Lasso	Elastic Net Regression

Table 1: Learners list

D Errors Evaluation

Even though the offline and online data are merged by cubic interpolation, the imbalance in sample size between offline and online data (about 20:1) makes it challenging to design a suitable split of the training set and test set. We do not want the estimations from the interpolation fit to be treated equally as the real data, so the method presented in this paper is to match each timestamp of the offline data to the online data that is closest to it in time, and treat this combined sample as a test set.

Algorithm 3 Creating the test set

```
Require:
vector of timestamps in the online data: $T_{online}$
vector of timestamps in the offline data: $T_{offline}$
dataframe of online data with the interpolate target values: $D = (X_{online}, Y_{interpolated})$
Initialise vector of index recording the test set: $I$
for $t \in T_{offline}$ do
    Find the timestamp $t$ in the online data such that:
        $\min_{\hat{t} \in T_{online}} : |\hat{t}_{online} - t|$
    Record this timestamp $\hat{t}$ in the index vector $I$:
        $I \leftarrow (I, \hat{t})$
end for
Set the test set as:
        $D_{test} = D[I, :]$
                        ▷ the rows of data where timestamps match $I$ are extracted
```

The average sample interval for offline data collection was 3.36 hours, compared to 0.5 hours for online data. The average time offset is 0.04 hours, and the maximum time deviation is 0.08 hours, according to the procedure stated above. Thus, on average, we have advanced or delayed the value of a true target value at time T by around 0.04 hours; in the worst case, value at time T is allocated to the data at time $T \pm 0.08$.

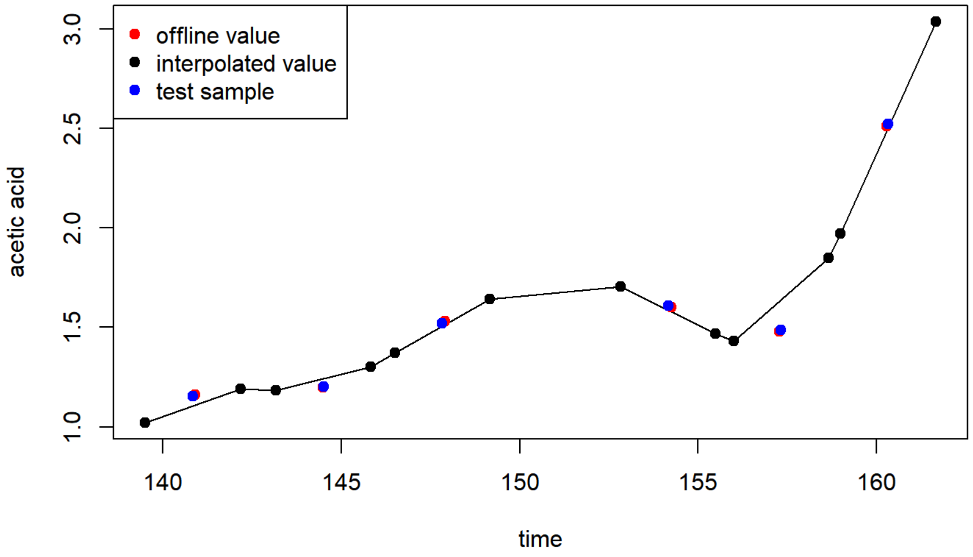


Figure 4: Extraction of test set. Red samples are the reliable target values, and the rest samples are estimated values corresponding to the online timestamp. Blue samples are extracted as test set due to the negligible difference between offline values and blue samples

On the basis of the acid value in the offline dataset, linear interpolation allows us to estimate that the fluctuation per hour is around 0.0049. Therefore, the average variation per 0.04 hours is fairly modest. Similarly, the maximum error is restricted to 0.08×0.0049. According to offline data, the mean value of acid is 3.5893, hence, the errors of the partitioning the test set is roughly bounded by:

$$Error_{test} \leq \frac{0.08 \times 0.0049}{3.5893} \approx 0.0109\%$$

Error analysis on the missing target values remains problematic and is anticipated to be studied in greater detail in the future. Practically, the same interpolation procedure may be used to match offline timestamps with online predictors. However, this increases uncertainty, as the effects of interpolation mistakes for each predictor add up and make the final estimate less reliable and estimation error more difficult to evaluate. In addition, the full dataset is considered the training set for building the model, as the reliable test dataset was only roughly one-twentieth of the full dataset. We do not have a larger sample of comparable data due to the uniqueness of the fermentation procedure for this batch of TPP-19, in which numerous devices were updated. This paper examines the single-batch scenario and seeks to grasp the relationship between the predictor variables and acid production. As interpretability is of the utmost priority, the generalizability of the model is compromised in order to obtain more accurate association patterns.

V RESULT

According to Algorithm 2, the final weighted variable's importance is standardised by min-max scaling which projects the importance into range the $(0, 1)$ for a more intuitive visualisation:

$$x_i = \frac{x_i - x_{min}}{x_{max} - x_{min}}$$

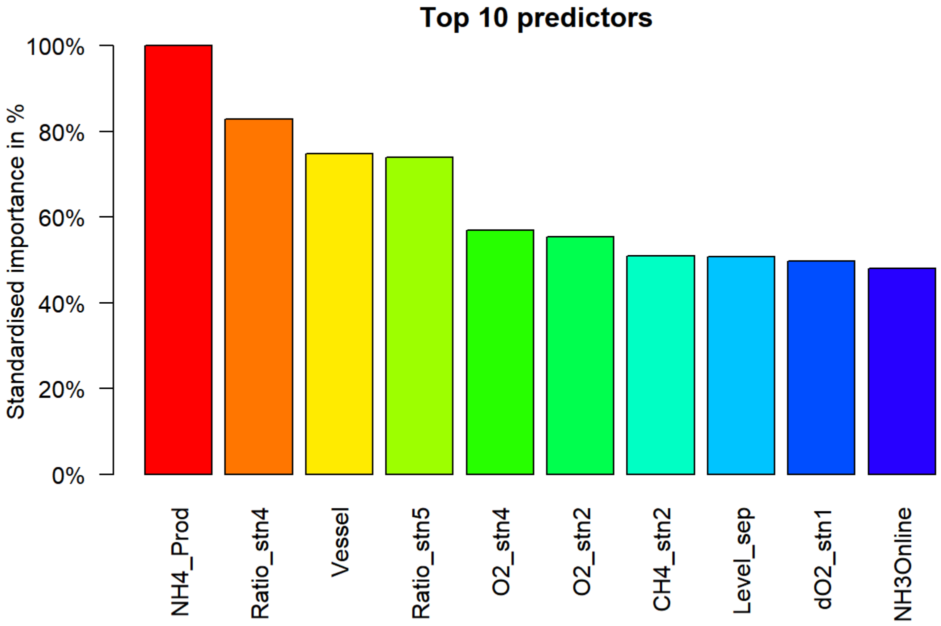


Figure 5: Most important predictors after normalisation

On the basis of importance ranking, a forward selection of the VIF was undertaken, resulting in a reduction from 65 to 45 predictors Noticeably, the average VIF values for all variables reduced significantly from 143 to 8.7. The majority of variables with extremely high VIF values were deleted, resulting in a decrease in VIF values for some variables. In general, multicollinearity is considered to be strong if the VIF is larger than 10. Although there was still considerable multicollinearity among the filtered variables, this was considered satisfactory for the reason that a reduced number of features drastically impacted the model's precision.

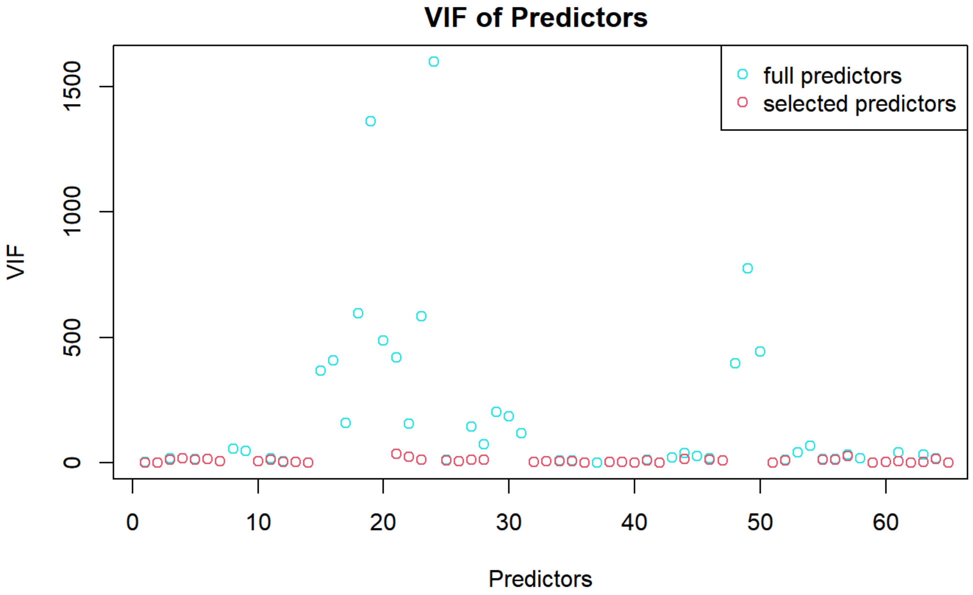


Figure 6: VIF of entire dataset and sub-dataset. Removed predictors are not shown in the figure.

Figure 3 illustrates the mean absolute percentage error of each learner used to calculate the final importance of the variables. The mean absolute percentage error of a leaner is given by:

$$\delta_{leaner} = \sum_{i=1}^{n} \frac{|\hat{y_i} - y_i|}{y_i}$$

Compared to mean squared error (MSE) and mean absolute error (MAE), it is more intuitive since it scales the prediction errors like R^2 so that the magnitude of the target value is negligible. Even further, it carries simpler interpretation in practice.

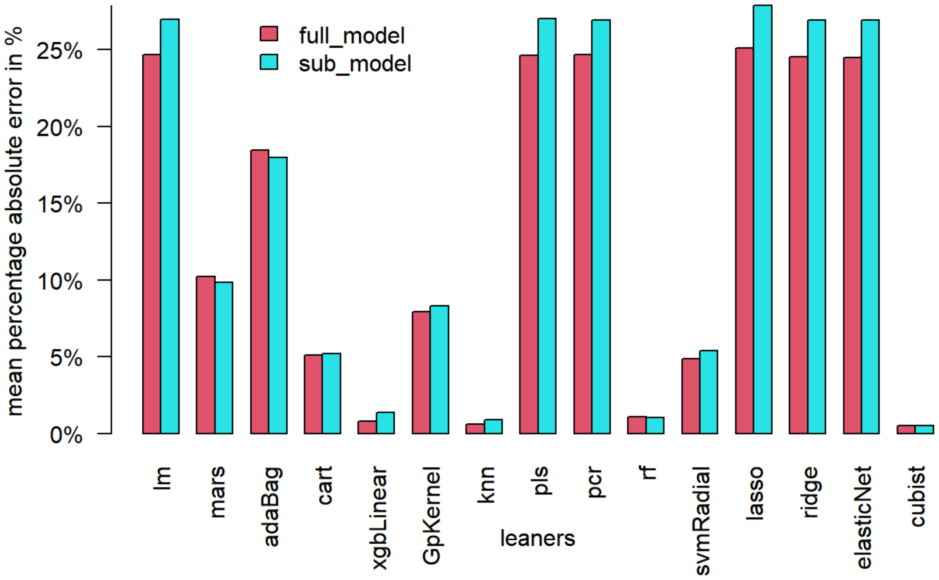
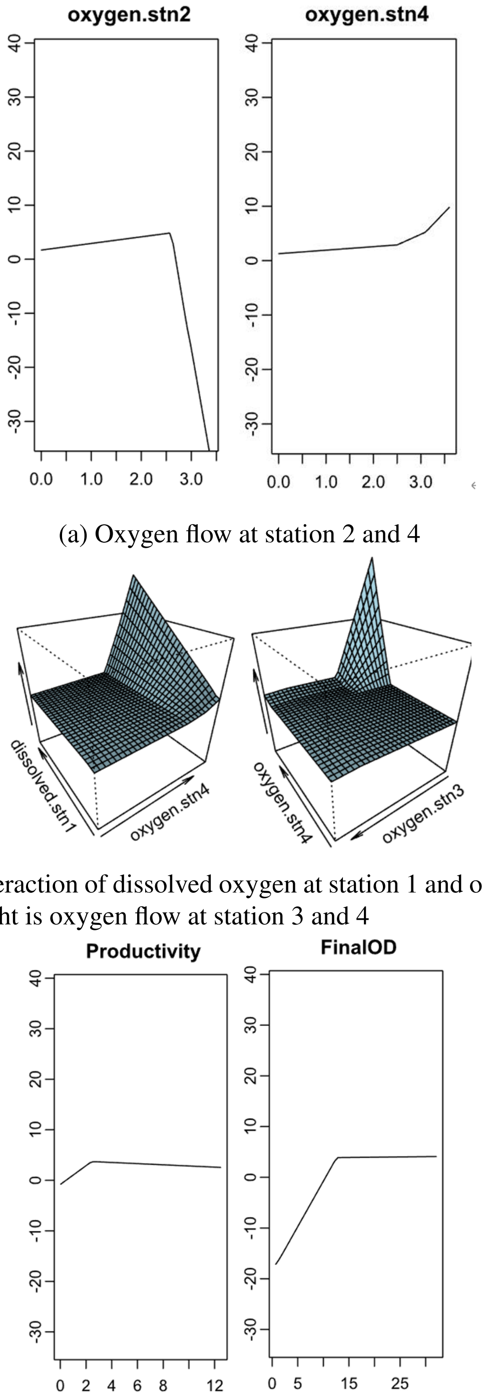


Figure 7: Mean percentage errors of entire dataset and sub-dataset

A MARS

In the test set, MARS outperformed other regression models, including linear regression models, shrinkage regression, and component regression. MARS's remarkable capacity to capture interaction terms makes the model optimum under circumstances that can be expressed explicitly. 55 combinations out of 15 predictors contribute to the final model equation:

$$
\begin{aligned}
Y = 5.5 - 79 * h(Oxygen.stn2 - 2.6) * h(phosphoric - 0.6) + 21 * h(Oxygen.stn2 - 2.6) * h(1.9 - magnesium) \\
+ 37 * h(Oxygen.stn4 - 2.5) * h(1.6 - Oxygen.stn3) - 34 * h(0.81 - NH4.Productivity) \\
+ 78 * h(Oxygen.stn4 - 2.5) * h(0.9 - phosphoric) * h(magnesium - 2.4) \\
+ \cdots
\end{aligned}
$$



(a) Oxygen flow at station 2 and 4

(b) Left is interaction of dissolved oxygen at station 1 and oxygen flow at station 4. Right is oxygen flow at station 3 and 4

(c) Productivity and final Optical Density

Figure 8: Partial relation from MARS. It is demonstrated how acetic acid varies as a function of single predictors or interaction factors.

Figure 8 demonstrates the partial connection (interaction effects are cancelled out by using pmethod=”partdep” in **plotmo** function) between some predictors in a form of basis expansion and acetic acids, where the predictors contributes differently when values are in different levels

In general, the MARS model captures most of the additive effects of predictors at multiple stages and the interaction of various factors, where the influence on acid changes qualitatively upon exceeding a particular threshold; however, tree-based models are more appropriate for representing non-linear fermentation processes. Even a single CART model is 5% more accurate than the MARS model. In other words, variables under different situations (corresponding to different branches of the tree) have varied impacts on the acid. This explains why the M5P model, a mixture of tree structure and regression, performs so well in this project.

B M5P

The final M5P model comprises 19 leaf nodes (i.e., 19 linear regression models), and only the most general rules are formulated as below. Among them, 12.5 $\%$ of the sample is allocated to **Rule 18**, and 9 $\%$ of the sample is allocated to **Rule 6**:

Rule 18:
if $oxygen.stn2 \leq 2.8$, $phosphoric \leq 0.6$, $LEL \geq 3.7$, $oxygen.stn3 \geq 1.5$ and $ammonia.stn1 \leq 1.3$:

$$Y = -35 + 1.53 * Productivity - 0.9 * NH_4.Productivity + 0.17 * Ratio.stn4 \\ - 0.055 * NH_3.online.reading + \cdots$$

Rule 6:
if $NH_4.Productivity \geq 2.2$, $oxygen.stn2 \geq 3$ and $Spare.flow \leq 1.6$:

$$Y = -6.3656917 + 2.19 * Spare.flow - 0.13 * Final.OD + 0.63 * NH_4.Productivity + 0.054 * Ratio.stn4 \\ + 2.4 * Offline.pH - 0.011 * LEL + \cdots$$

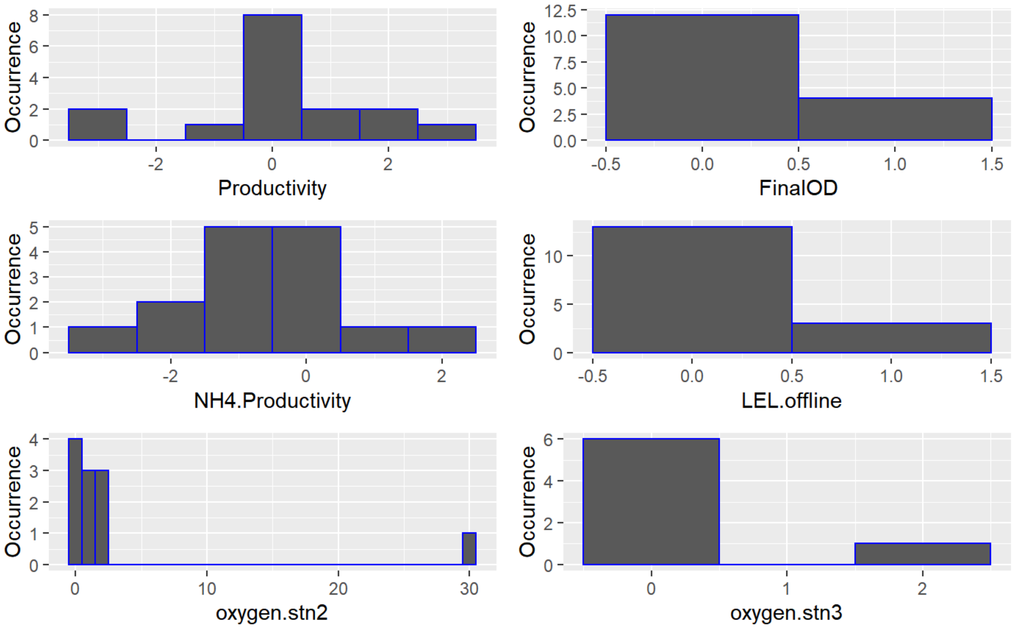


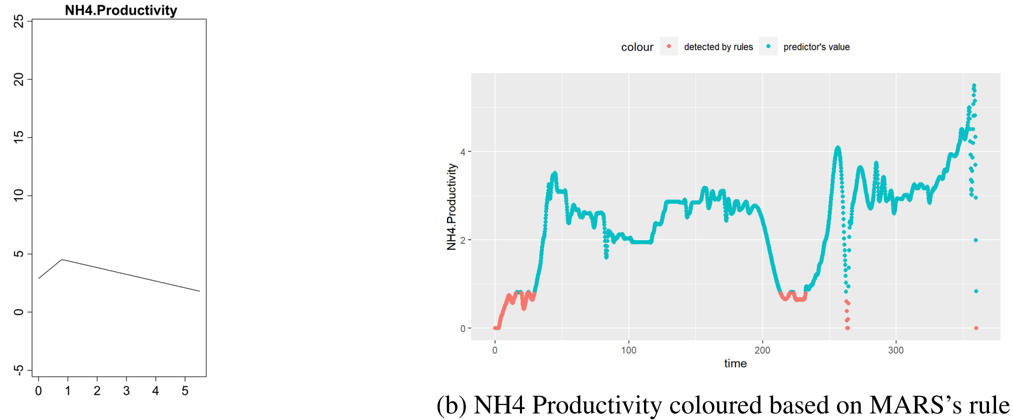
Figure 9: Occurrence of estimated coefficients of important predictors summarised in 19 rules

Figures 8 illustrates the coefficients of predictors under different rule circumstances. As the cubist package incorporates 'rational reconstruction' based on the M5P tree, the model is more extensively rule-based instead of tree-based and hence to some extent over-fitted. Due to the model's complicated parameters, it is difficult to construct an intuitive tree structure. The figures provide a partial illustration of the effect of the predictors. Nonetheless, it is necessary to discuss in greater detail the relevance of the chemical characteristics and the variation of predictors throughout the specific time periods.

VI DISCUSSION

Ultimately, the M5P model with multiple committees can achieve: 0.22% in mean absolute percentage error, and MARS can be improved to: 7.8% via bagging, which are outstanding compared with the random forest and xgboost-linear tree. Despite the fact that the two models favoured distinct variables, it was acknowledged that both NH4 productivity and oxygen flow at station 2 were significant. The M5P model set the split point

for $oxygen.stn2$ at 2.8, whereas the MARS model placed it at 2.6, indicating that this value may represent the maximum limit of some optimal control conditions.
From MARS, the outcome knots in the basis expansion can be tracked back to grip more patterns. Figure 10.a shows a slight increase in acetic acid when the NH4.Productivity is below about 0.8. In contrast, a value over this threshold hurts acid growth, effectively regulating acid production.



(a) NH4 Productivity VS acetic acid

(b) NH4 Productivity coloured based on MARS's rule

Figure 10: Influence of NH4 Productivity

After extracting samples with a value less than the criterion of 0.8 (the red points in the figure 10.b), it can be seen that, with the exception of the drop near 260 hours due to missing offline data, the model indicates a time in which NH4 begins to develop rapidly from a lower and stable value. Thus, a strong correlation is captured. The calculation of NH4 Productivity is:

$$\frac{ammonia.flow}{Nitrogen.in.feed} \times 6.25protein.concentration$$

It indicates the amount of protein and the ammonia-to-nitrogen ratio in the final fermentation product, which is a complex mixture of potential variables and be considered significant. Figure 11 for the final optical density suggests that the previous time periods correlate to the acceleration period and the temporary decay period typical of microbial fermentation.

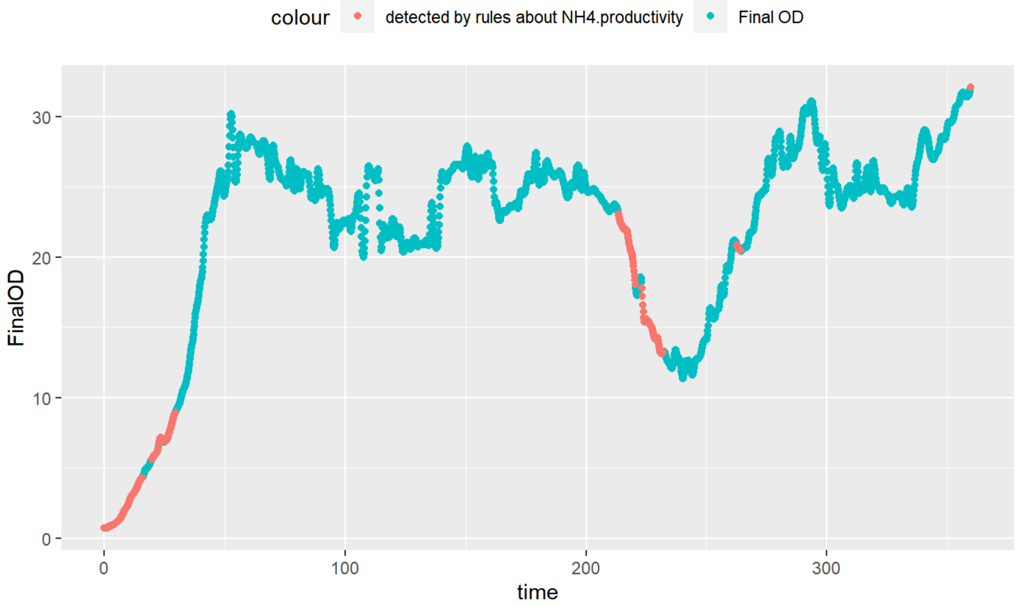


Figure 11: Optical density

As for the M5P model, the comprehensive capacity to partition the sample space makes overfitting a common occurrence. Oftentimes, while tracing back the corresponding samples according to each criteria, a continuous significant time period is observed, indicating overfitting. Due to the large number of conditions and regression equations, it is thus difficult to intuitively grasp the specific affecting factors. Figure 9 depicts a rough histogram of the regression coefficient distribution. Productivity has an approximately normal distribution, which indicates that its effect on acid is divided into several stages. Notably, the LEL variable, which was absolutely irrelevant in MARS, played a substantial role in M5P. Its significance is represented not in the regression coefficient, but in the fact that it is involved in 89% of the rules. That is, differing levels of LEL serve as control conditions for other variables affecting acetic acid production. The distribution of oxygen flow at stations 2 and 3 are right-skewed and centre on the right side of 0. This conclusion is consistent with MARS, which eliminates the possibility of M5P overfitting a portion of the noise. However, the threshold for qualitative change may be influenced by factors other than solely oxygen flow.

More error analysis studies based on similar environmental batches of fermentation processes and the inaccuracy of interpolation matching should be carried out. This paper presents a method for combining data obtained at various time points that does not make extensive verification. A choice has to be made between underfitting and overfitting because we cannot split the dataset using typical methods (splitting most of the calculated samples into the training/testing set doesn't seem to make much sense other than ”providing gorgeous outcomes”). Under the premise of not pursuing accuracy and generalisation, the core of this study is to capture prospective laws as much as feasible using interpretable models. As a consequence, according to the model's criteria, the objective that this essay seeks to achieve is backtracking the actual meaning behind it and analysing it. Simultaneously, under high-dimensional multicollinearity variables, how to begin feature engineering that does not modify the original information (without dimensionality reduction or dimensionality rise) is a topic worth discussing. The method proposed in this paper may be rife with chance because the results of each step of greedy selection will have distinct consequences on future candidate variables. Furthermore, whether it is correct to roughly estimate a comprehensive best candidate subset by weighted integration of many existing learners is under-considered in this paper, since methods comparable to the idea of bagging generally introduce selection bias. Fortunately, the output is not disappointing in this dataset. Different models explore different rules, but they may also complement or even enhance each other. This could be a life philosophy when individuals make decisions: weighting the bests.

VII CONCLUSION

This paper is based on data in practical industry where the the multicollinearity exists, employing cubic interpolation to match offline and online data and sliding window noise reduction method to assure smoothness of predictors without sacrificing too many intriguing bursts. Through forwarding VIF selection, the suggested implementation of an ensemble method determines the significance of variables and eliminates redundancies. The remarkable capability of Multivariate Adaptive Regression Splines and M5P tree to describe nonlinear time-varying issues is uncovered, ensuring accuracy and offering clear representation. The ensembled versions of MARS and M5P can attain performance comparable to black box models (such SVM, xgbboost tree, and random forest). Both models point to significant contributions from oxygen flow at station 2 and NH4, implying potential gains from more research on the design of station 2 and NH4 control during the acceleration phase.

References

Bergstra, J. & Bengio, Y. (2012), ‘Random search for hyper-parameter optimization.’, *Journal of machine learning research* **13**(2).

Biswas, A., Takakuwa, F., Yamada, S., Matsuda, A., Saville, R. M., LeBlanc, A., Silverman, J. A., Sato, N. & Tanaka, H. (2020), ‘Methanotroph (methylococcus capsulatus, bath) bacteria meal as an alternative protein source for japanese yellowtail, seriola quinqueradiata’, *Aquaculture* **529**, 735700.

Calude, C. S. & Longo, G. (2017), ‘The deluge of spurious correlations in big data’, *Foundations of science* **22**(3), 595–612.

Cimander, C., Carlsson, M. & Mandenius, C.-F. (2002), ‘Sensor fusion for on-line monitoring of yoghurt fermentation’, *Journal of biotechnology* **99**(3), 237–248.

Desai, K., Badhe, Y., Tambe, S. S. & Kulkarni, B. D. (2006), ‘Soft-sensor development for fed-batch bioreactors using support vector regression’, *Biochemical Engineering Journal* **27**(3), 225–239.

Fodor, I. K. (2002), A survey of dimension reduction techniques, Technical report, Lawrence Livermore National Lab., CA (US).

Kim, Y., Street, W. N. & Menczer, F. (2002), ‘Evolutionary model selection in unsupervised learning’, *Intelligent data analysis* **6**(6), 531–556.

Liu, H. & Motoda, H. (2012), *Feature selection for knowledge discovery and data mining*, Vol. 454, Springer Science & Business Media.

Liu, K., Xin, J. & Xu, N. (2013), ‘Research progress in single cell protein production and application’, *Chinese Journal of Animal Husbandry* **49**(10), 56–60.

Takiguchi, N., Shimizu, H. & Shioya, S. (1997), ‘An on-line physiological state recognition system for the lysine fermentation process based on a metabolic reaction model’, *Biotechnology and bioengineering* **55**(1), 170–181.

Thibault, J., Acuna, G., Perez-Correa, R., Jorquera, H., Molin, P. & Agosin, E. (2000), ‘A hybrid representation approach for modelling complex dynamic bioprocesses’, *Bioprocess Engineering* **22**(6), 547–556.

Xing, E. P., Jordan, M. I., Karp, R. M. et al. (2001), Feature selection for high-dimensional genomic microarray data, *in* ‘Icml’, Vol. 1, Citeseer, pp. 601–608.

Yuan, Z., Bogaert, H., Devisscher, M., Vanrolleghem, P. & Verstraete, W. (1999), ‘On-line estimation of the maximum specific growth rate of nitrifiers in activated sludge systems’, *Biotechnology and bioengineering* **65**(3), 265–273.

Zheng, J. (2020), Research on soft sensor modelling of the glutamate fermentation process, PhD thesis, Wuxi: Jiangnan University.

Appendices

Forward Selection of predictors through VIF

this is run: 1 ——————————

going to add variable: vessel

if we add variable variable: vessel , the top largest VIFs are:

NH4$_{P}roductivity_{kg_{D}CW_{h}O2_{C}h4_{r}atio_{s}taion4vessel}$

1.32 1.26 1.08

the current variable’s VIF is 1.08

it doesn’t bring a lot, vessel automatically added

this is run: 2 ——————————

going to add variable: O2$_{C}h4_{r}atio_{s}taion5$

if we add variable variable: O2$_{C}h4_{r}atio_{s}taion5, thetoplargestVIFsare:$

O2$_{C}h4_{r}atio_{s}taion4NH4_{P}roductivity_{kg_{D}CW_{h}O2_{C}h4_{r}atio_{s}taion5}$

1.56 1.33 1.26

the current variable’s VIF is 1.26

it doesn’t bring a lot, O2$_{C}h4_{r}atio_{s}taion5automaticallyadded$

this is run: 3 ——————————

going to add variable: oxygen$_{f}low_{s}tn4$

if we add variable variable: oxygen$_{f}low_{s}tn4, thetoplargestVIFsare:$

oxygen$_{f}low_{s}tn4O2_{C}h4_{r}atio_{s}taion5NH4_{P}roductivity_{kg_{D}CW_{h}}$

2.57 2.21 2.08
the current variable's VIF is 2.57
it doesn't bring a lot, oxygen$_flow_stn4automaticallyadded$
this is run: 4 ————————————
going to add variable: oxygen$_flow_stn2$
if we add variable variable: oxygen$_flow_stn2, thetoplargestVIFsare:$
oxygen$_flow_stn2O2_Ch4_ratio_staion5oxygen_flow_stn4$
11.24 10.93 2.62
the current variable's VIF is 11.24
it brings the changes of VIFs(top 3):
O2$_Ch4_ratio_staion5vesseloxygen_flow_stn4$
8.72 0.10 0.05
enter the delte variables: 1 is delete, 0 is not deleted
0
chose to add: oxygen$_flow_stn2$
this is run: 5 ————————————
going to add variable: methane$_flow_stn2$
if we add variable variable: methane$_flow_stn2, thetoplargestVIFsare:$
oxygen$_flow_stn2methane_flow_stn2O2_Ch4_ratio_staion5$
185.86 181.94 11.05
the current variable's VIF is 181.94
it brings too much, methane$_flow_stn2automaticallydeleted$
this is run: 6 ————————————
going to add variable: Level$_on_the_seperator$
if we add variable variable: Level$_on_the_seperator, thetoplargestVIFsare:$
oxygen$_flow_stn2O2_Ch4_ratio_staion5oxygen_flow_stn4$
11.26 10.98 2.70
the current variable's VIF is 1.21
it doesn't bring a lot, Level$_on_the_seperatorautomaticallyadded$
this is run: 7 ————————————
going to add variable: dissolved$_oxygen_at_stn1$
if we add variable variable: dissolved$_oxygen_at_stn1, thetoplargestVIFsare:$
oxygen$_flow_stn2O2_Ch4_ratio_staion5oxygen_flow_stn4$
11.70 11.11 2.70
the current variable's VIF is 2.01
it doesn't bring a lot, dissolved$_oxygen_at_stn1automaticallyadded$
this is run: 8 ————————————
going to add variable: NH3$_online_reading$
if we add variable variable: NH3$_online_reading, thetoplargestVIFsare:$
oxygen$_flow_stn2O2_Ch4_ratio_staion5O2_Ch4_ratio_staion4$
11.79 11.14 4.63
the current variable's VIF is 3.62
it doesn't bring a lot, NH3$_online_readingautomaticallyadded$
this is run: 9 ————————————
going to add variable: phosphoric$_acid_flow$
if we add variable variable: phosphoric$_acid_flow, thetoplargestVIFsare:$
oxygen$_flow_stn2O2_Ch4_ratio_staion5O2_Ch4_ratio_staion4$
12.04 11.15 5.03
the current variable's VIF is 4.33
it doesn't bring a lot, phosphoric$_acid_flowautomaticallyadded$
this is run: 10 ————————————
going to add variable: Productivity$_{kg_DCW_h}$
if we add variable variable: Productivity$_{kg_DCW_{h,thetoplargestVIFsare:}}$

oxygen$_flow_stn2O2_Ch4_ratio_staion5O2_Ch4_ratio_staion4$
12.09 11.17 5.79
the current variable's VIF is 2.47
it doesn't bring a lot, Productivity$_{kg_DCW_{h_a}utomaticallyadded}$
this is run: 11 ————————————
going to add variable: LEL
if we add variable variable: LEL$_{,thetoplargestVIFsare:}$
oxygen$_flow_stn2O2_Ch4_ratio_staion5oxygen_flow_stn4$
12.22 11.17 6.17
the current variable's VIF is 2.53
it doesn't bring a lot, LEL$_{_automaticallyadded}$
this is run: 12 ————————————
going to add variable: Final$_OD$
if we add variable variable: Final$_OD, thetoplargestVIFsare:$
oxygen$_flow_stn2O2_Ch4_ratio_staion5Final_OD$
12.34 11.28 8.83
the current variable's VIF is 8.83
it doesn't bring a lot, Final$_ODautomaticallyadded$
this is run: 13 ————————————
going to add variable: oxygen$_flow_stn3$
if we add variable variable: oxygen$_flow_stn3, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
15.12 11.91 11.38
the current variable's VIF is 6.46
it doesn't bring a lot, oxygen$_flow_stn3automaticallyadded$
this is run: 14 ————————————
going to add variable: oxygen$_flow_stn1$
if we add variable variable: oxygen$_flow_stn1, thetoplargestVIFsare:$
oxygen$_flow_stn1oxygen_flow_stn2Final_OD$
23.70 19.34 14.56
the current variable's VIF is 23.7
it brings too much, oxygen$_flow_stn1automaticallydeleted$
this is run: 15 ————————————
going to add variable: K$_{ppm}$
if we add variable variable: K$_{ppm,thetoplargestVIFsare:}$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
16.45 12.68 11.38
the current variable's VIF is 1.6
it doesn't bring a lot, K$_{ppm_automaticallyadded}$
this is run: 16 ————————————
going to add variable: methane$_flow_stn4$
if we add variable variable: methane$_flow_stn4, thetoplargestVIFsare:$
methane$_flow_stn4oxygen_flow_stn4oxygen_flow_stn2$
398.09 358.74 16.66
the current variable's VIF is 398.09
it brings too much, methane$_flow_stn4automaticallydeleted$
this is run: 17 ————————————
going to add variable: ammonia$_pump_flow_1$
if we add variable variable: ammonia$_pump_flow_1, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
16.56 12.68 11.48
the current variable's VIF is 7.99
it doesn't bring a lot, ammonia$_pump_flow_1automaticallyadded$

this is run: 18 ——————————————
going to add variable: dissolved$_oxygen_at_stn4$
if we add variable variable: dissolved$_oxygen_at_stn4, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
18.83 13.22 11.54
the current variable's VIF is 2.57
it doesn't bring a lot, dissolved$_oxygen_at_stn4automaticallyadded$
this is run: 19 ——————————————
going to add variable: harvest$_flow$
if we add variable variable: harvest$_flow, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
19.34 14.89 11.63
the current variable's VIF is 5.64
it doesn't bring a lot, harvest$_flowautomaticallyadded$
this is run: 20 ——————————————
going to add variable: dissolved$_oxygen_at_stn2$
if we add variable variable: dissolved$_oxygen_at_stn2, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
32.91 15.55 11.87
the current variable's VIF is 9.4
it brings the changes of VIFs(top 3):
oxygen$_flow_stn2dissolved_oxygen_at_stn1oxygen_flow_stn3$
13.57 3.09 1.76
enter the delte variables: 1 is delete, 0 is not deleted
0
chose to add: dissolved$_oxygen_at_stn2$
this is run: 21 ——————————————
going to add variable: magnesium$_potassium_flow$
if we add variable variable: magnesium$_potassium_flow, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
33.10 16.04 11.87
the current variable's VIF is 8.35
it doesn't bring a lot, magnesium$_potassium_flowautomaticallyadded$
this is run: 22 ——————————————
going to add variable: methane$_flow_stn1$
if we add variable variable: methane$_flow_stn1, thetoplargestVIFsare:$
oxygen$_flow_stn2methane_flow_stn1Final_OD$
47.05 28.95 16.33
the current variable's VIF is 28.95
it brings too much, methane$_flow_stn1automaticallydeleted$
this is run: 23 ——————————————
going to add variable: dissolved$_oxygen_at_stn5$
if we add variable variable: dissolved$_oxygen_at_stn5, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODO2_Ch4_ratio_staion5$
33.10 16.32 11.91
the current variable's VIF is 1.49
it doesn't bring a lot, dissolved$_oxygen_at_stn5automaticallyadded$
this is run: 24 ——————————————
going to add variable: O2$_Ch4_ratio_staion2$
if we add variable variable: O2$_Ch4_ratio_staion2, thetoplargestVIFsare:$
oxygen$_flow_stn4oxygen_flow_stn2O2_Ch4_ratio_staion2$
45.99 35.30 29.65
the current variable's VIF is 29.65

it brings too much, O2$_{C}h4_{r}atio_{s}taion2automaticallydeleted$
this is run: 25 ————————————
going to add variable: pH$_{a}t_{s}tn5$
if we add variable variable: pH$_{a}t_{s}tn5, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}DO2_{C}h4_{r}atio_{s}taion5$
34.48 16.42 11.93
the current variable's VIF is 2.02
it doesn't bring a lot, pH$_{a}t_{s}tn5automaticallyadded$
this is run: 26 ————————————
going to add variable: methane$_{f}low_{s}tn3$
if we add variable variable: methane$_{f}low_{s}tn3, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn3methane_{f}low_{s}tn3oxygen_{f}low_{s}tn2$
77.60 73.85 34.61
the current variable's VIF is 73.85
it brings too much, methane$_{f}low_{s}tn3automaticallydeleted$
this is run: 27 ————————————
going to add variable: ammonia$_{t}otalier$
if we add variable variable: ammonia$_{t}otalier, thetoplargestVIFsare:$
ammonia$_{t}otalieroxygen_{f}low_{s}tn2ammonia_{p}ump_{f}low_{1}$
47.65 34.97 23.01
the current variable's VIF is 47.65
it brings too much, ammonia$_{t}otalierautomaticallydeleted$
this is run: 28 ————————————
going to add variable: pH$_{a}t_{s}tn1$
if we add variable variable: pH$_{a}t_{s}tn1, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
34.88 17.43 11.95
the current variable's VIF is 9.97
it brings the changes of VIFs(top 3):
pH$_{a}t_{s}tn5oxygen_{f}low_{s}tn3Final_{O}D$
7.864027 1.178858 1.011691
enter the delte variables: 1 is delete, 0 is not deleted
0
chose to add: pH$_{a}t_{s}tn1$
this is run: 29 ————————————
going to add variable: pH$_{a}t_{s}tn3$
if we add variable variable: pH$_{a}t_{s}tn3, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}DpH_{a}t_{s}tn5$
35.13 18.17 14.82
the current variable's VIF is 11.56
it doesn't bring a lot, pH$_{a}t_{s}tn3automaticallyadded$
this is run: 30 ————————————
going to add variable: ferrus$_{s}ulphate_{f}low$
if we add variable variable: ferrus$_{s}ulphate_{f}low, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
36.84 18.21 14.97
the current variable's VIF is 11.3
it doesn't bring a lot, ferrus$_{s}ulphate_{f}lowautomaticallyadded$
this is run: 31 ————————————
going to add variable: NH4$_{O}ffline_{ppm}$
if we add variable variable: NH4$_{O}ffline_{ppm, thetoplargestVIFsare:}$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
37.63 18.58 15.16

the current variable's VIF is 1.49
it doesn't bring a lot, NH4$_{O}ffline_{ppm_{a}utomaticallyadded}$
this is run: 32 ——————————
going to add variable: dissolved$_{o}xygen_{a}t_{s}tn3$
if we add variable variable: dissolved$_{o}xygen_{a}t_{s}tn3, thetoplargestVIFsare :$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
38.04 19.51 17.72
the current variable's VIF is 3.41
it doesn't bring a lot, dissolved$_{o}xygen_{a}t_{s}tn3automaticallyadded$
this is run: 33 ——————————
going to add variable: trace$_{e}lements_{f}low$
if we add variable variable: trace$_{e}lements_{f}low, thetoplargestVIFsare :$
oxygen$_{f}low_{s}tn2ferrus_{s}ulphate_{f}lowFinal_{O}D$
38.06 27.43 20.41
the current variable's VIF is 12.51
it brings too much, trace$_{e}lements_{f}lowautomaticallydeleted$
this is run: 34 ——————————
going to add variable: spare$_{d}osing_{p}ump_{f}low$
if we add variable variable: spare$_{d}osing_{p}ump_{f}low, thetoplargestVIFsare :$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
38.15 20.01 17.87
the current variable's VIF is 7.76
it doesn't bring a lot, spare$_{d}osing_{p}ump_{f}lowautomaticallyadded$
this is run: 35 ——————————
going to add variable: CO2$_{i}n_{o}ffgas$
if we add variable variable: CO2$_{i}n_{o}ffgas, thetoplargestVIFsare :$
CO2$_{i}n_{o}ffgasoxygen_{f}low_{s}tn2oxygen_{f}low_{s}tn3$
38.63 38.27 23.59
the current variable's VIF is 38.63
it brings too much, CO2$_{i}n_{o}ffgasautomaticallydeleted$
this is run: 36 ——————————
going to add variable: ammonia$_{p}ump_{f}low_{2}$
if we add variable variable: ammonia$_{p}ump_{f}low_{2}, thetoplargestVIFsare :$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
39.13 20.15 18.39
the current variable's VIF is 10.65
it doesn't bring a lot, ammonia$_{p}ump_{f}low_{2}automaticallyadded$
this is run: 37 ——————————
going to add variable: oxygen$_{t}otaliser$
if we add variable variable: oxygen$_{t}otaliser, thetoplargestVIFsare :$
oxygen$_{t}otaliseroxygen_{f}low_{s}tn2oxygen_{f}low_{s}tn3$
101.96 39.17 33.14
the current variable's VIF is 101.96
it brings too much, oxygen$_{t}otaliserautomaticallydeleted$
this is run: 38 ——————————
going to add variable: LEL$_{i}n_{o}ffgas$
if we add variable variable: LEL$_{i}n_{o}ffgas, thetoplargestVIFsare :$
LEL$_{i}n_{o}ffgasLEL_{oxygen_{f}low_{s}tn2}$
42.80 42.46 39.53
the current variable's VIF is 42.8
it brings too much, LEL$_{i}n_{o}ffgasautomaticallydeleted$
this is run: 39 ——————————
going to add variable: CH4$_{i}n_{o}ffgas$

if we add variable variable: CH4$_in_offgas, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
39.44 20.20 18.52
the current variable's VIF is 4.64
it doesn't bring a lot, CH4$_in_offgasautomaticallyadded$
this is run: 40 ——————————
going to add variable: optical$_density$
if we add variable variable: optical$_density, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
39.66 21.57 18.56
the current variable's VIF is 5.74
it doesn't bring a lot, optical$_densityautomaticallyadded$
this is run: 41 ——————————
going to add variable: Cooling$_Water_Return_for_Temped_System$
if we add variable variable: Cooling$_Water_Return_for_Temped_System, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
40.28 22.05 19.28
the current variable's VIF is 5.68
it doesn't bring a lot, Cooling$_Water_Return_for_Temped_Systemautomaticallyadded$
this is run: 42 ——————————
going to add variable: pump$_inlet$
if we add variable variable: pump$_inlet, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
40.30 22.05 19.28
the current variable's VIF is 1.02
it doesn't bring a lot, pump$_inletautomaticallyadded$
this is run: 43 ——————————
going to add variable: oxygen$_pressure_to_fermentor_loop_mixture$
if we add variable variable: oxygen$_pressure_to_fermentor_loop_mixture, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
40.34 22.06 19.93
the current variable's VIF is 3.52
it doesn't bring a lot, oxygen$_pressure_to_fermentor_loop_mixtureautomaticallyadded$
this is run: 44 ——————————
going to add variable: oxygen$_flow_stn5$
if we add variable variable: oxygen$_flow_stn5, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn5$
40.87 22.63 20.82
the current variable's VIF is 20.82
it brings too much, oxygen$_flow_stn5automaticallydeleted$
this is run: 45 ——————————
going to add variable: O2$_Ch4_ratio_staion3$
if we add variable variable: O2$_Ch4_ratio_staion3, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
41.00 22.92 19.93
the current variable's VIF is 17.67
it brings too much, O2$_Ch4_ratio_staion3automaticallydeleted$
this is run: 46 ——————————
going to add variable: methane$_flow_stn5$
if we add variable variable: methane$_flow_stn5, thetoplargestVIFsare:$
oxygen$_flow_stn2Final_ODoxygen_flow_stn3$
40.88 22.59 20.56
the current variable's VIF is 18.71

it brings too much, methane$_{f}low_{s}tn5automaticallydeleted$

this is run: 47 ————————————

going to add variable: Cooling$_{l}oop_{B_{v}}alve_{o}pening$

if we add variable variable: Cooling$_{l}oop_{B_{v}}alve_{o}pening, thetoplargestVIFsare :$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

42.79 22.97 20.54

the current variable's VIF is 2.94

it doesn't bring a lot, Cooling$_{l}oop_{B_{v}}alve_{o}peningautomaticallyadded$

this is run: 48 ————————————

going to add variable: Cooling$_{l}oop_{A_{v}}alve_{o}pening$

if we add variable variable: Cooling$_{l}oop_{A_{v}}alve_{o}pening, thetoplargestVIFsare :$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

42.80 23.83 20.57

the current variable's VIF is 5.72

it doesn't bring a lot, Cooling$_{l}oop_{A_{v}}alve_{o}peningautomaticallyadded$

this is run: 49 ————————————

going to add variable: Methane$_{t}otalier$

if we add variable variable: Methane$_{t}otalier, thetoplargestVIFsare :$

Methane$_{t}otalieroxygen_{f}low_{s}tn2oxygen_{f}low_{s}tn3$

115.96 43.03 33.34

the current variable's VIF is 115.96

it brings too much, Methane$_{t}otalierautomaticallydeleted$

this is run: 50 ————————————

going to add variable: DCW$_{C}PI_{g_{L}}$

if we add variable variable: DCW$_{C}PI_{g_{L,thetoplargestVIFsare:}}$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

42.96 24.65 22.99

the current variable's VIF is 14.77

it doesn't bring a lot, DCW$_{C}PI_{g_{L_{a}utomaticallyadded}}$

this is run: 51 ————————————

going to add variable: partial$_{p}ressure_{.}DP.$

if we add variable variable: partial$_{p}ressure_{.}DP., thetoplargestVIFsare :$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

42.96 24.65 23.02

the current variable's VIF is 1.28

it doesn't bring a lot, partial$_{p}ressure_{.}DP.automaticallyadded$

this is run: 52 ————————————

going to add variable: Offline$_{p}H$

if we add variable variable: Offline$_{p}H, thetoplargestVIFsare :$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

43.32 24.73 23.13

the current variable's VIF is 1.87

it doesn't bring a lot, Offline$_{p}Hautomaticallyadded$

this is run: 53 ————————————

going to add variable: methane$_{p}ressure_{t}o_{f}ermentor_{l}oop_{m}ixture$

if we add variable variable: methane$_{p}ressure_{t}o_{f}ermentor_{l}oop_{m}ixture, thetoplargestVIFsare :$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

44.59 24.98 23.24

the current variable's VIF is 3.7

it doesn't bring a lot, methane$_{p}ressure_{t}o_{f}ermentor_{l}oop_{m}ixtureautomaticallyadded$

this is run: 54 ————————————

going to add variable: Tempered$_{W}ater_{T}emperature_{b}efore_{T}emped_{w}ater_{c}ooler$

if we add variable variable: Tempered$_{W}ater_{T}emperature_{b}efore_{T}emped_{w}ater_{c}ooler, thetoplargestVIFsare :$

oxygen$_{f}low_{s}tn2$
45.07
Tempered$_{W}ater_{T}emperature_{b}efore_{T}emped_{w}ater_{c}ooler$
32.28
Final$_{O}D$
25.29
the current variable's VIF is 32.28
it brings too much, Tempered$_{W}ater_{T}emperature_{b}efore_{T}emped_{w}ater_{c}oolerautomaticallydeleted$
this is run: 55 ——————————
going to add variable: calcium$_{c}hloride_{f}low$
if we add variable variable: calcium$_{c}hloride_{f}low, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
45.70 27.54 25.55
the current variable's VIF is 17.77
it brings too much, calcium$_{c}hloride_{f}lowautomaticallydeleted$
this is run: 56 ——————————
going to add variable: O2$_{C}h4_{r}atio_{s}taion1$
if we add variable variable: O2$_{C}h4_{r}atio_{s}taion1, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
45.74 26.28 24.20
the current variable's VIF is 10.19
it doesn't bring a lot, O2$_{C}h4_{r}atio_{s}taion1automaticallyadded$
this is run: 57 ——————————
going to add variable: Cooling$_{L}oop_{AB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer$
if we add variable variable: Cooling$_{L}oop_{AB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Cooling_{L}oop_{AB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer$
45.93 43.85
oxygen$_{f}low_{s}tn3$
28.09
the current variable's VIF is 43.85
it brings too much, Cooling$_{L}oop_{AB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixerautomaticallydeleted$
this is run: 58 ——————————
going to add variable: Cooling$_{L}oop_{BB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer_{J}1117$
if we add variable variable: Cooling$_{L}oop_{BB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer_{J}1117, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2$
45.97
Cooling$_{L}oop_{BB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer_{J}1117$
42.02
oxygen$_{f}low_{s}tn3$
28.05
the current variable's VIF is 42.02
it brings too much, Cooling$_{L}oop_{BB}roth_{R}eturn_{t}o_{F}ermenter_{m}ixer_{J}1117automaticallydeleted$
this is run: 59 ——————————
going to add variable: sodium$_{h}ydroxide_{f}low$
if we add variable variable: sodium$_{h}ydroxide_{f}low, thetoplargestVIFsare:$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$
45.77 26.28 24.29
the current variable's VIF is 1.06
it doesn't bring a lot, sodium$_{h}ydroxide_{f}lowautomaticallyadded$
this is run: 60 ——————————
going to add variable: Fermentor$_{f}liud_{t}o_{a}mmonia_{OD_{m}eter}$
if we add variable variable: Fermentor$_{f}liud_{t}o_{a}mmonia_{OD_{meter,thetoplargestVIFsare:}}$
oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

45.85 26.28 24.32

the current variable's VIF is 1.74

it doesn't bring a lot, Fermentor$_{f}liud_{t}o_{a}mmonia_{OD_{m}eterautomaticallyadded}$

this is run: 61 ————————————

going to add variable: Tempered$_{W}ater_{T}emperature_{a}fter_{T}emped_{w}ater_{c}ooler$

if we add variable variable: Tempered$_{W}ater_{T}emperature_{a}fter_{T}emped_{w}ater_{c}ooler, thetoplargestVIFsare:$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

45.87 26.28 24.39

the current variable's VIF is 2.2

it doesn't bring a lot, Tempered$_{W}ater_{T}emperature_{a}fter_{T}emped_{w}ater_{c}oolerautomaticallyadded$

this is run: 62 ————————————

going to add variable: Pressure$_{a}t_{t}he_{e}nd_{o}f_{t}he_{l}oop$

if we add variable variable: Pressure$_{a}t_{t}he_{e}nd_{o}f_{t}he_{l}oop, thetoplargestVIFsare:$

oxygen$_{f}low_{s}tn2Final_{O}Doxygen_{f}low_{s}tn3$

45.93 26.33 24.40

the current variable's VIF is 1.16

it doesn't bring a lot, Pressure$_{a}t_{t}he_{e}nd_{o}f_{t}he_{l}oopautomaticallyadded$

Figures of outcomes of full predictors

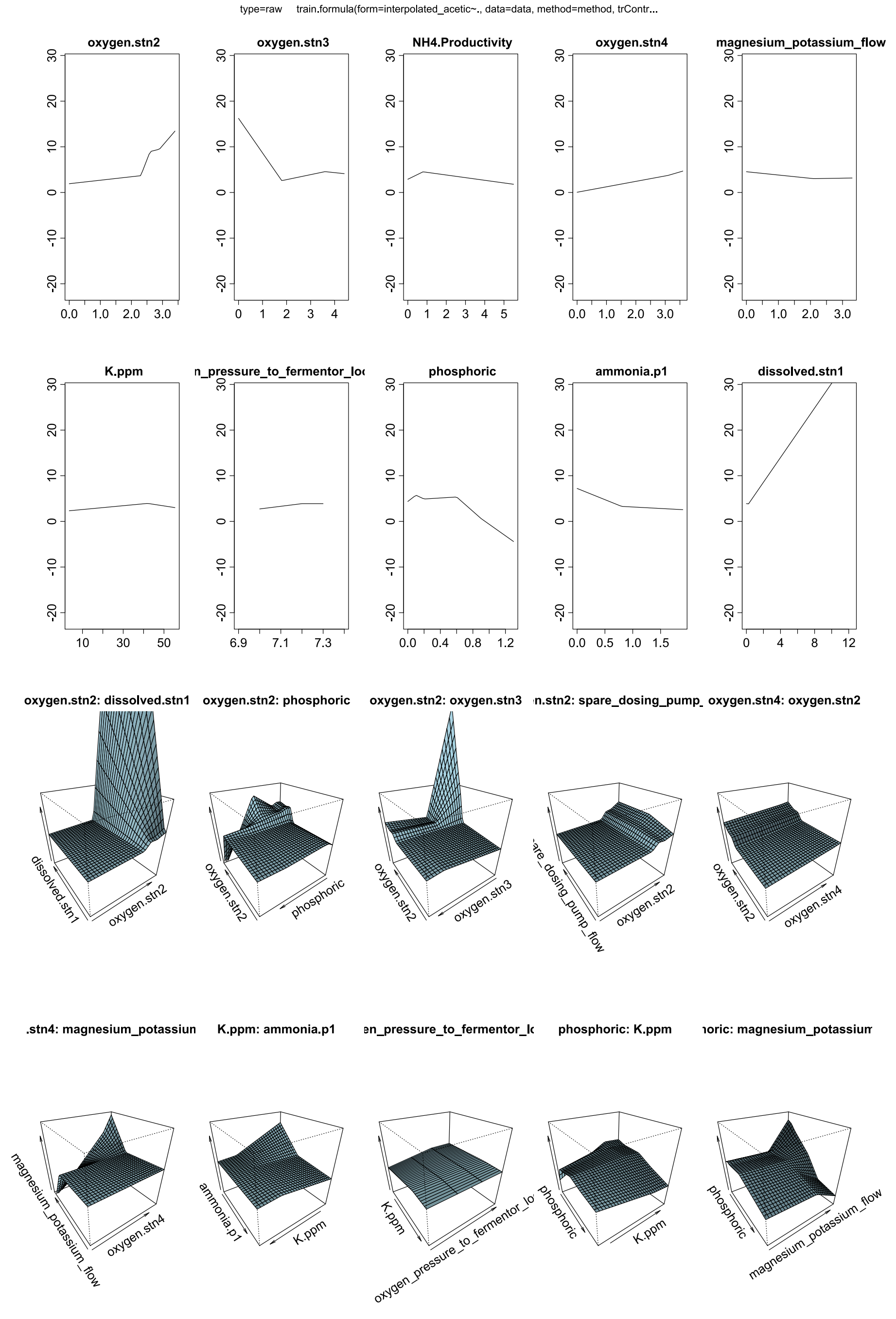


Figure 12: MARS

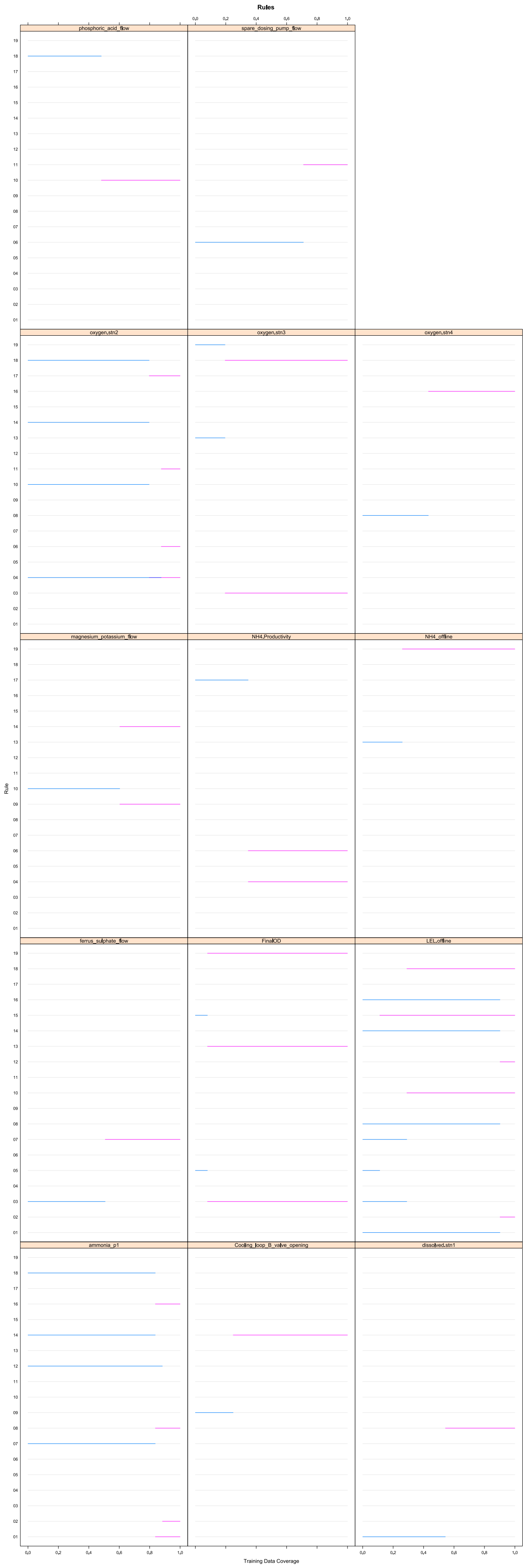


Figure 13: M5P rules

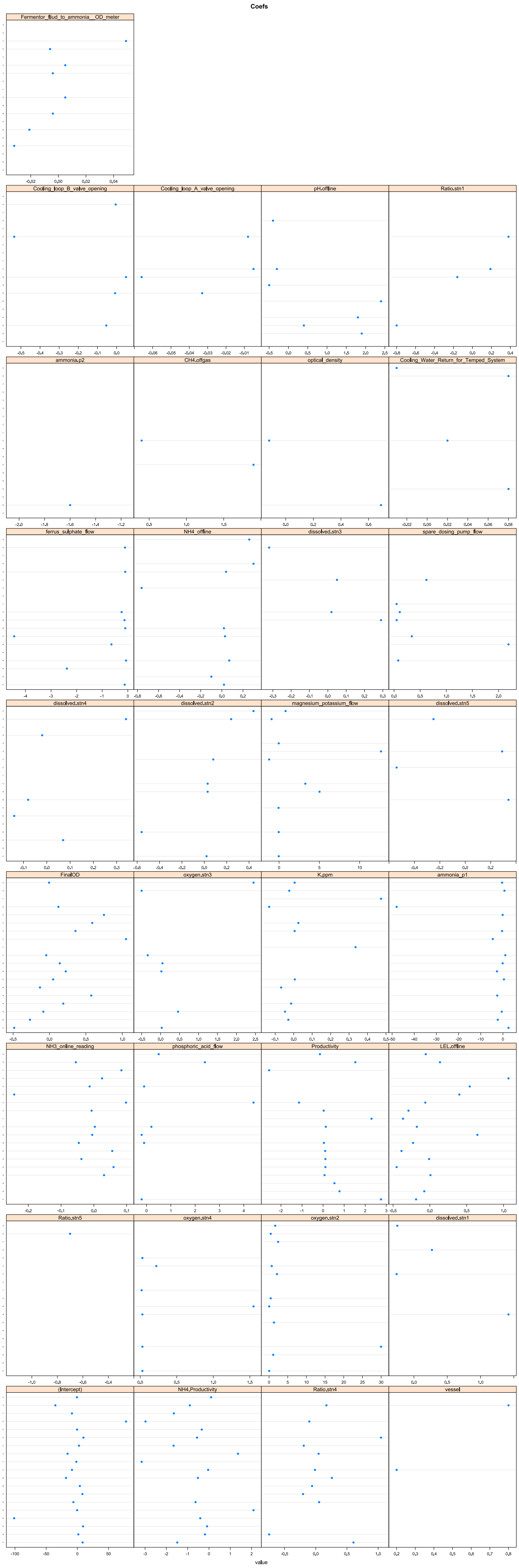


Figure 14: M5P coefficients