Dynamic soft sensor for silica content

Erik Kuitunen, Pauli Anttonen, Joona Lappalainen

January 17, 2025

1 Overview

1.1 Project goal

The Project goal is to predict the silica content of the froth flotation phase outlet stream using dynamic soft sensor. The primary aim of the froth flotation phase is to reduce the concentration of silica ore in iron ore. The process is illustrated in Figure 1. To this end, numerous chemicals, such as collectors, frothers, and depressants, are added. Collectors make iron ore particles hydrophobic and float on top, while depressants prevent silica from attaching to bubbles.[1]

The analyzed dataset contains 6 monthly time series of all process variables and input stream concentrations. [3] The project's main goal is to identify the most important variables and use them to estimate this silica content in the output stream based on them.

1.2 Dataset description

Dataset [3] features time series of process variables from the froth flotation phase of mining process. During the froth flotation phase, silica-rich iron ore is supplied to several flotation tanks to extract silica from the iron ore.

The first column of the dataset features time. Each timestamp includes 180 measurements of process variables and a single value for the output silica and iron concentrations. Input ore concentrations are measured once per hour in columns 2 and 3. Amina and starch are added to the ore, and their flow rates are recorded in columns 4 and 5. The ore pulp flow, pH, and density can be found in columns 6 through 8. Columns 9 through 15 represent airflow in the flotation tanks, and columns 16 through 22 represent tank levels. Silica and iron output concentrations can be found in the last two columns. The histograms of each column can be seen in Figure 2.

Visualization of the non-uniform sample is shown in Figure 3. The dataset includes a discontinuity period from 16.3.2017 to 29.3.2017, which was removed from the data. The process variables were also resampled to match the concentration sampling rate. Resampling was done by averaging samples in the given window at the rate of one hour.

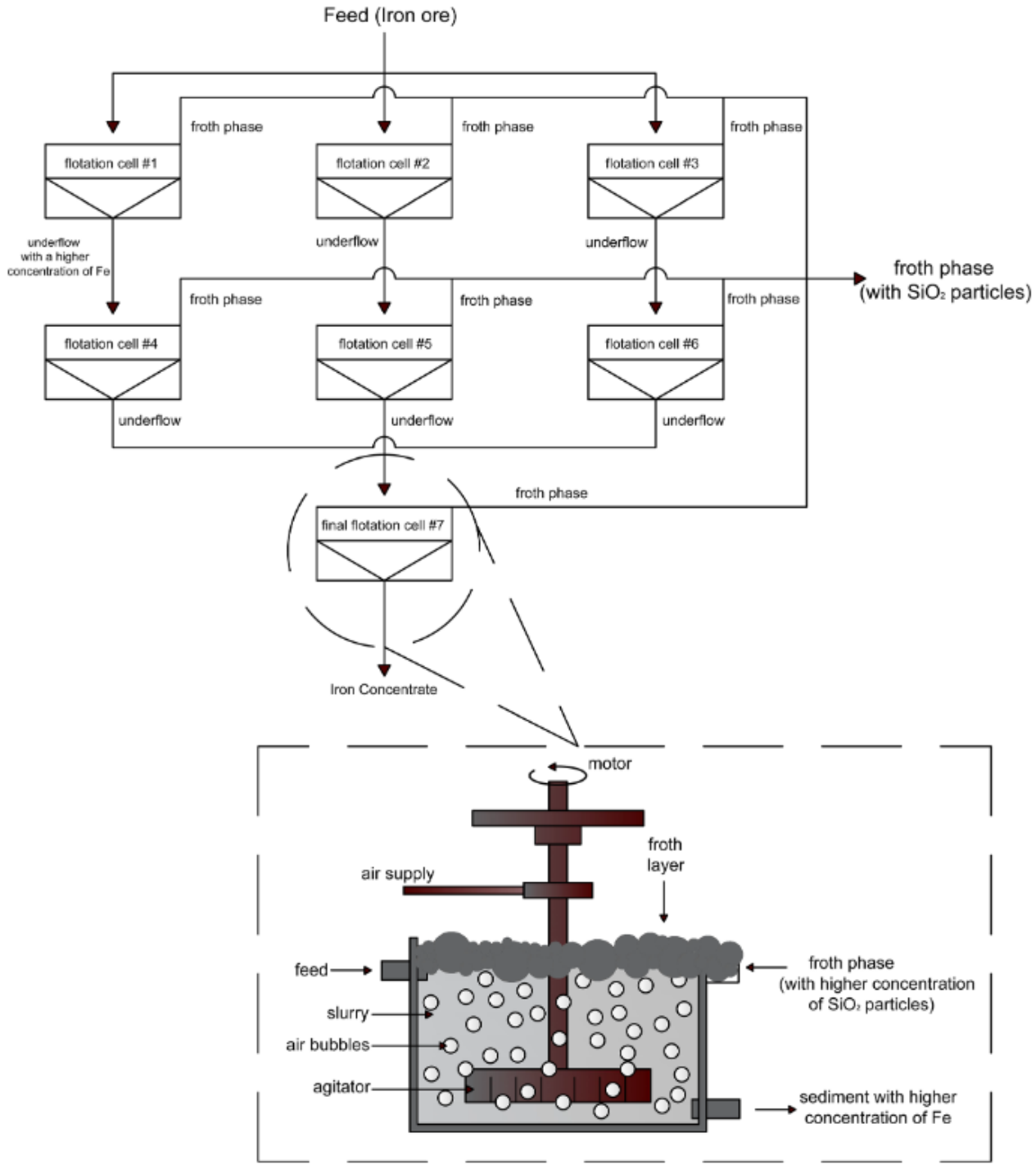

Figure 1: Flotation process [2].

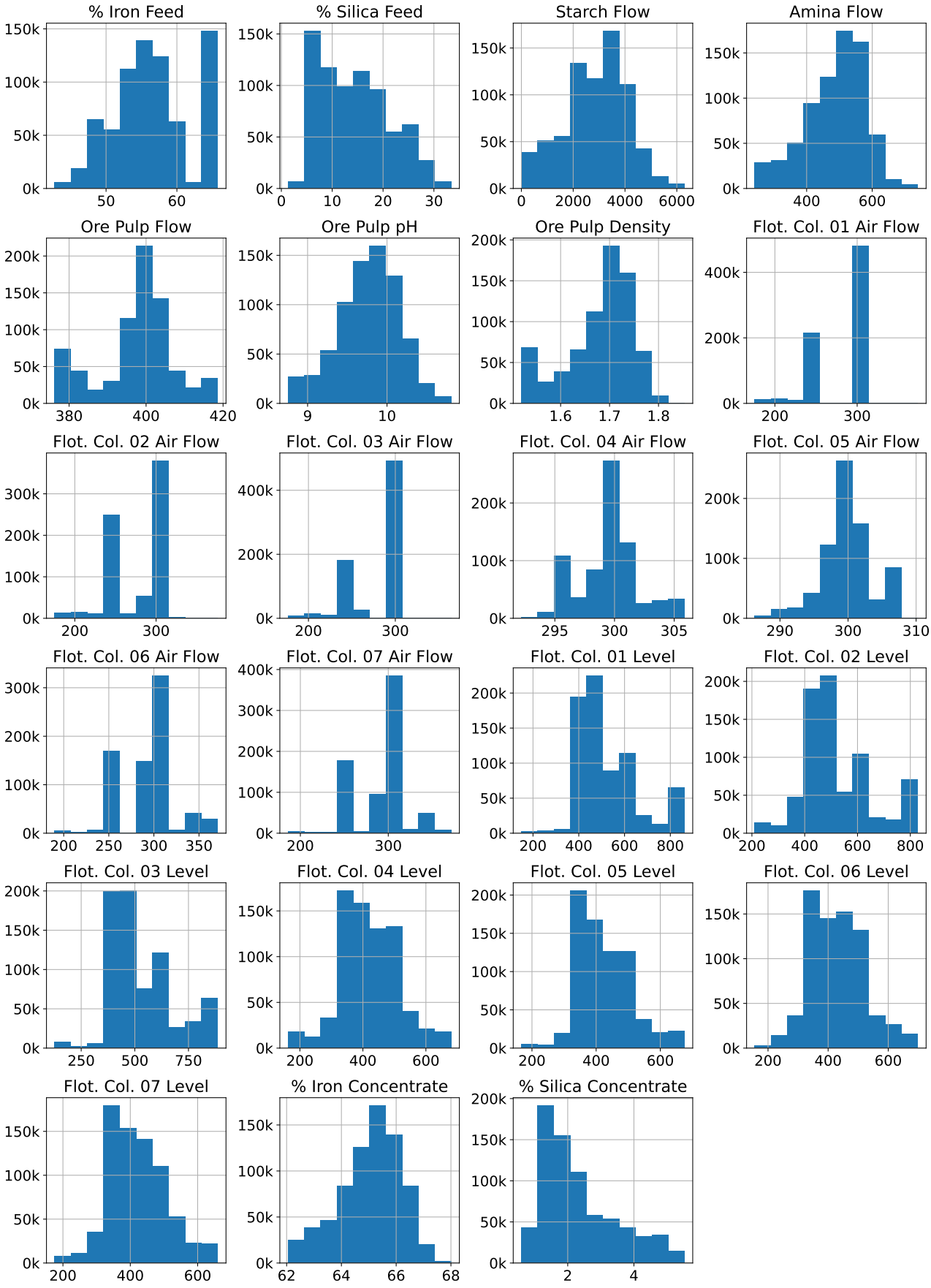

Figure 2: Histograms of the variables of unprocessed data.

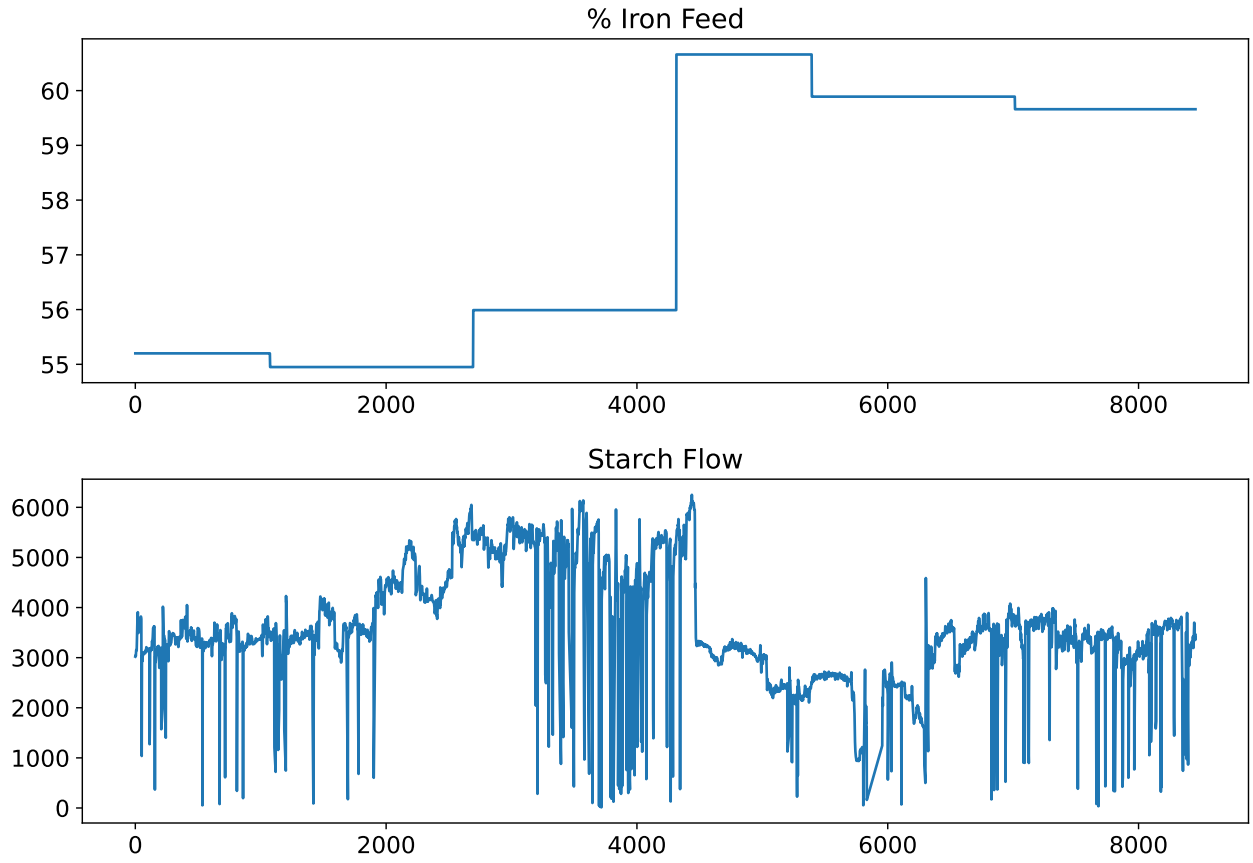

Figure 3: Data from first two days of measurements. The percentage of iron feed has a lower sample rate (once per hour) as opposed to starch flow (180 samples per hour). Note that Iron feed is kept constant for longer period of time than one hour.

2 Methods

2.1 Data Preprocessing

Extending feature space with lagged variables

Assuming silica concentration in a continuous process that changes based on previous values, we can populate the data using lagged variables of the variables. This is done by adding input features to the dataset such that

$$\mathrm{Si}_{\mathrm{out}}^{t-n}(t) = \mathrm{Si}_{\mathrm{out}}(t-n).$$

Several lagged variables have been generated from each original feature, extending the feature space from $M = 21$ features to $\widetilde{M} = 153$. After evaluating their importance, some may be removed.

Data split

The splitting of the data set is done as follows: the data set is divided into calibration-validation partition and test partition. The calibration-validation partition will be used in the process of calibrating the model by determining the optimal amount of variables to use, and which variables to use. Test partition will be used to evaluate the performance of the model.

These splits are illustrated in Figure 4, which shows that the splits are chosen so that there are no long periods of time with constant or missing data inside.

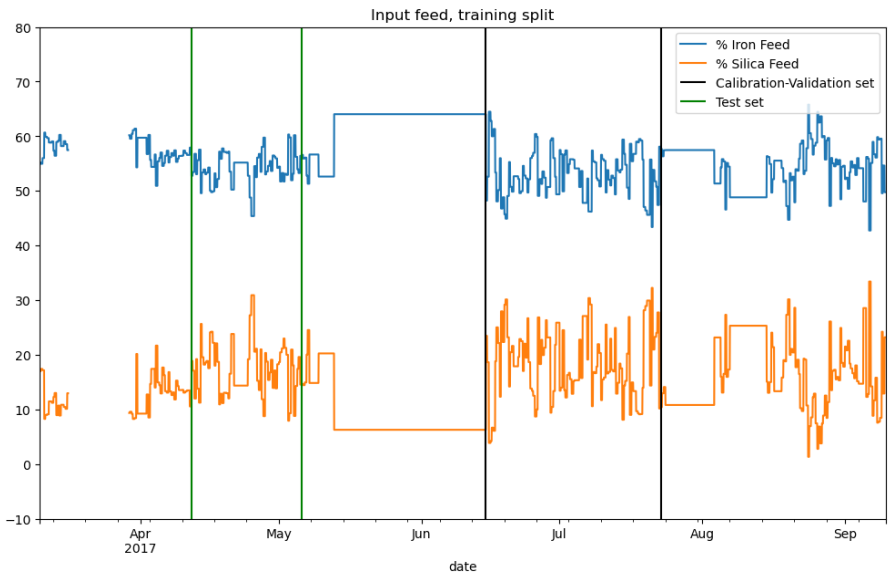

Figure 4: With respect to missing values in the first two features, calibration-validation and test data sets are chosen.

Centering and scaling

Since the independent variables differ in variation ranges, the data is standardized to unit length variation. Thus, it ensures that all the variables are equally important in the model. Also, the variables are mean-centered so that the latent variable means go through origin.

2.2 Partial Least Squares fit

Partial Least Squares (PLS) regression combines elements of principal component analysis (PCA) and multiple regression to predict a dependent variable (Y) from a set of independent variables (X). PLS tries to find components that explain covariance between X and Y whereas PCA only tries to explain variance in X.

$$X = TP^T + E$$
$$Y = UQ^T + F,$$

where T and U are object scores for x and y, P and Q are variable loadings, and E and F represent error or noise respectively.

PLS is chosen as the method for predicting the silica concentration since the data contains many correlated features. The feature space is also relatively large, improving the efficiency of using PLS instead of a regular linear model. We use regular PLS for determining the number of variables to use in the final model, after which dynamic PLS is utilized to find the optimal number of latent variables.

The number of latent variables can be reduced from the original dimensionality to generalize the model. This is done by calculating Q^2 scores for each number of latent variables and dynamic model window sizes:

$$Q^2 = \frac{\sum(Y_{\text{Pred}} - Y_{\text{Val}})^2}{\sum(Y_{\text{Cal}} - \hat{Y}_{\text{Cal}})^2},$$

where Y_{Pred} represents the predicted values of the dependent variable obtained from the model for the validation set, Y_{Val} denotes the actual values of the dependent variable in the validation partition, Y_{Cal} refers to the values of the dependent variable in the calibration partition, and $\hat{Y}_{\text{Cal}}$ is the mean of Y_{Cal}.

The final model is then calibrated using the whole calibration validation data set. The data is tested to confirm the results, and the performance is then evaluated.

2.3 Dynamic PLS model

Dynamic PLS utilizes a sliding window in fitting of the model. Where standard PLS uses all of the calibration data (or folds of it, if using k-fold cross-validation), the dynamic PLS evaluates the model for a window of some chosen size. Various window sizes will be used to determine the most appropriate size for the final model. Since we are building a dynamic soft sensor model, we are determining also the number of LVs using the dynamic PLS. The data inside the window is used to train the model, and three following data points are used to predict the output. Then, R^2 and Q^2 scores are calculated for each window slide step. The algorithm for one window size and one number of LVs is described as follows:

1. Window size S and number of LVs v chosen. Initialize R^2 and Q^2 vectors.
2. Loop over the length of the data subtracted by the window size and the amount of the data used for prediction. Each loop iteration moves the starting point of the window forward by one index.
   (a) Pick calibration data according to the window size and the test data according to the prediction data length.

(b) Fit model using the calibration data

(c) Make predictions using the test data

(d) Check for outliers

(e) Calculate R^2 and Q^2 and append to the corresponding vectors.

3. Calculate the mean of R^2 and Q^2 vectors and append to the mean score matrices.

2.4 Outlier detection

From the data, we can observe that the silica concentration does not exceed 6 percent and naturally cannot fall below 0. If our model suggests values outside these boundaries, we will replace them with the previously measured value. We can also assume that silica is highly dependent on the previously measured values. Referring back to the data set, there is $\pm 2.5\%$ difference between two consequent measures over 99% of the time. Thus, the model also treats values that differ over 2.5% from the previously measured values as outliers. Removing these values improved average performance and Q^2 significantly. In real-life scenarios, these values would be discussed with experts on the process and adjusted accordingly.

2.5 Selection of variables

Not all of the variables from the extended feature space should be used to achieve a more general model. To determine which variables to use, the so-called R^2 method is used. Next, we will shortly describe the algorithm, and the more detailed version can be found in [5].

1. Select V, the number of variables to test in the selection process.
2. Loop over each number of variables to test $v \in [1, 2, ..., V]$, and for each test variable number, fit a PLS model using $m = [1, 2, ..., M]$ variables.
3. Make a prediction using the PLS model and calculate R^2 score for each m. Thus, for each v and M, R^2 score is obtained.
4. Find the largest R^2 score, and select that variable from the list of accepted variables. Remove that variable from the list of m.
5. Repeat for each v.

The algorithm is performed using whole calibration-validation data before the dynamic filter is applied in order to extract the most important variables.

2.6 Modeling workflow

In Figure 5, the modeling workflow is illustrated. The aim is to use PLS method to select the variables for the dynamic PLS method, which is used to choose the number of latent variables and the size of the window to calibrate the final model.

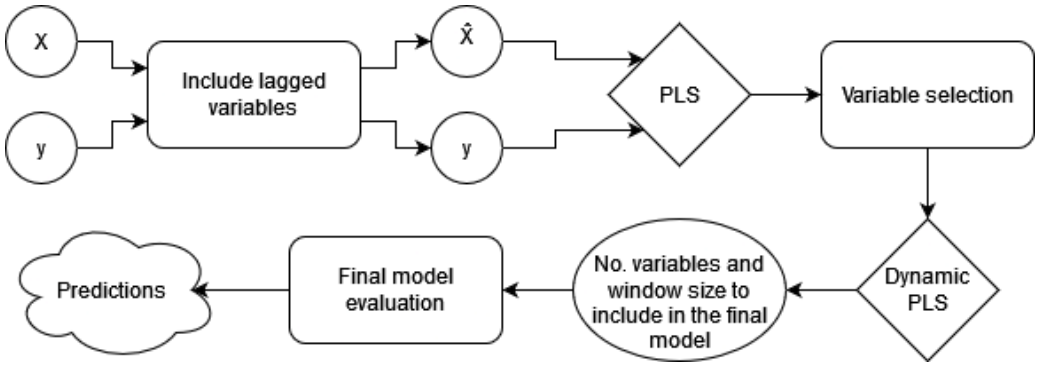

Figure 5: Model workflow

3 Results

The first model is evaluated using all 219 variables, which include both original variables and lagged variables. The results with different windows and No. of LVs is shown in Figure 6.

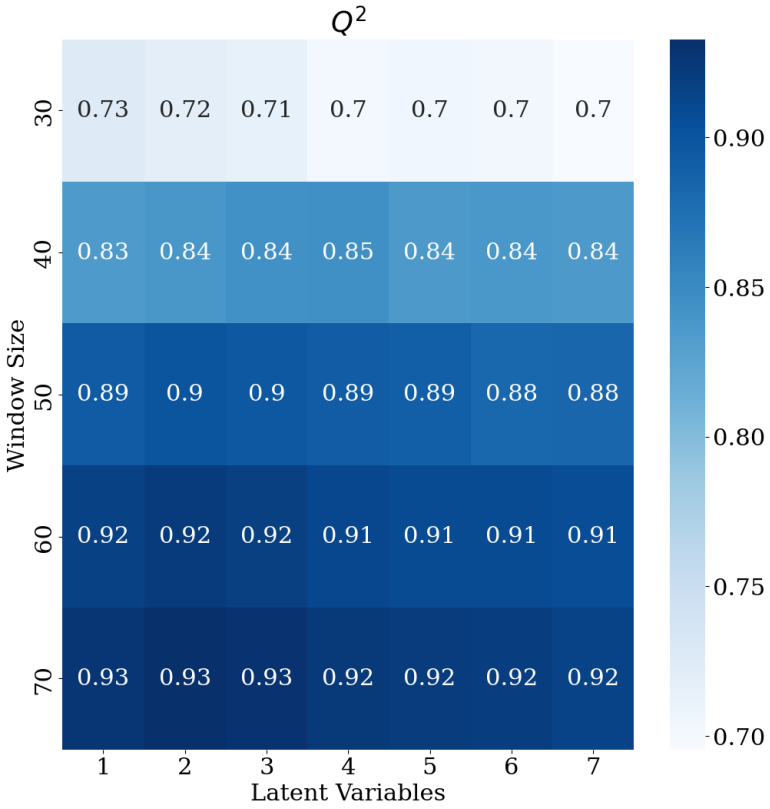

Figure 6: Q^2 before variable selection

The results for the R^2 method are shown in Figure 7. The maximum value of 23 variables is shown with the red dot. Then, these 23 variables are chosen for dynamic model evaluation.

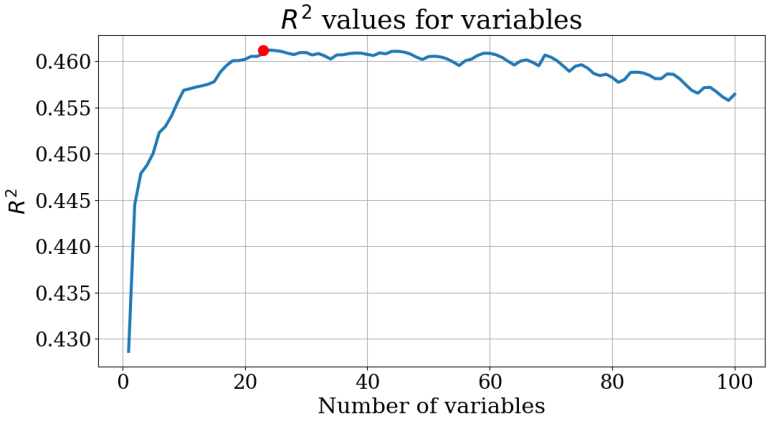

Figure 7: Variable selection criteria

Model evaluation after variable selection is shown in Figure 8. Q^2 value is influenced by the size of the calibration partition and the window size, which reflects the high Q^2 on bigger window

sizes. The Q^2 value increase is relatively low, when moving from 60 to 70, so a window size of 60 is chosen. Three latent variables are chosen for testing the model.

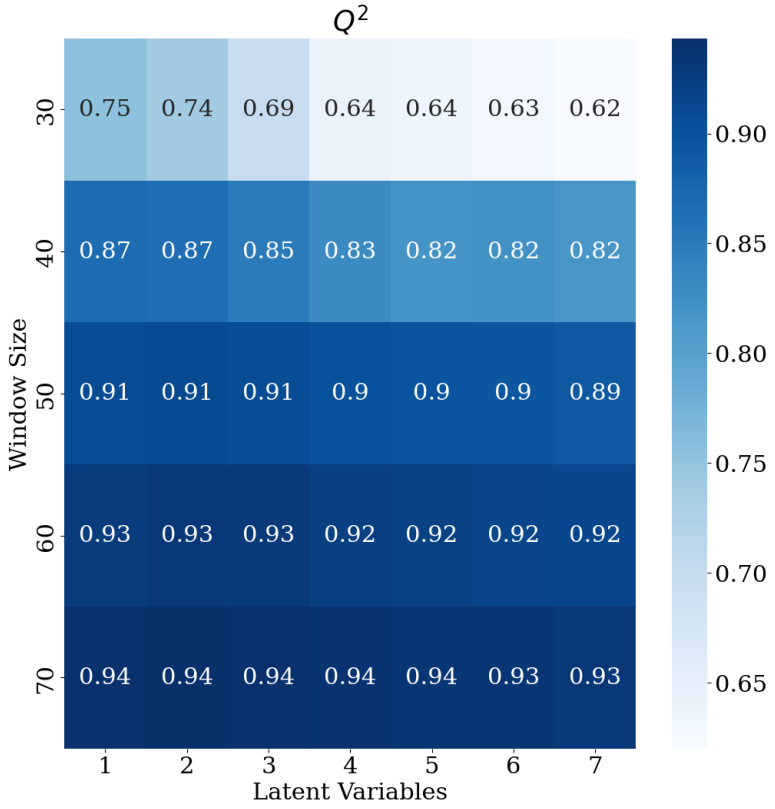

Figure 8: Q^2 after variable selection

Then, the model is tested on the test data that had not been used to calibrate the model. The time series for the first-hour prediction against the true value is plotted in Figure 9. Histograms for future three hours are shown in Figure 10, and scatter plot for predictions and true values is presented in Figure 11.

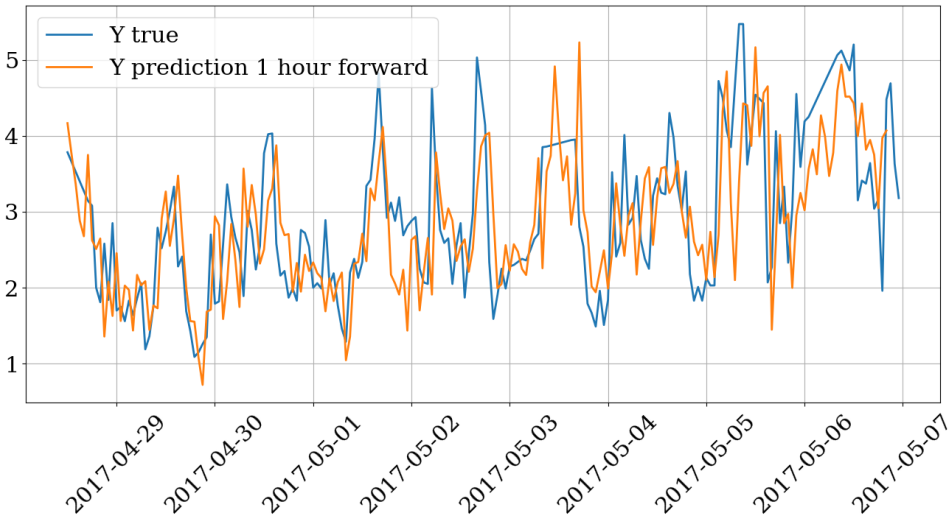

Figure 9: The first-hour prediction against the true value

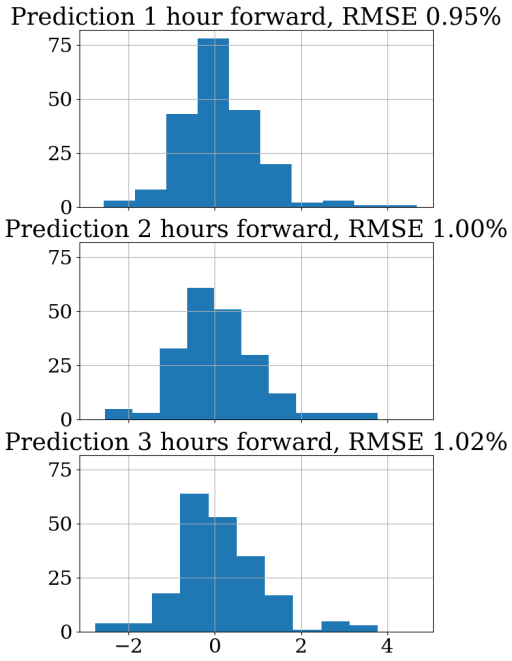

Figure 10: Predictions for one, two and three hours to the future

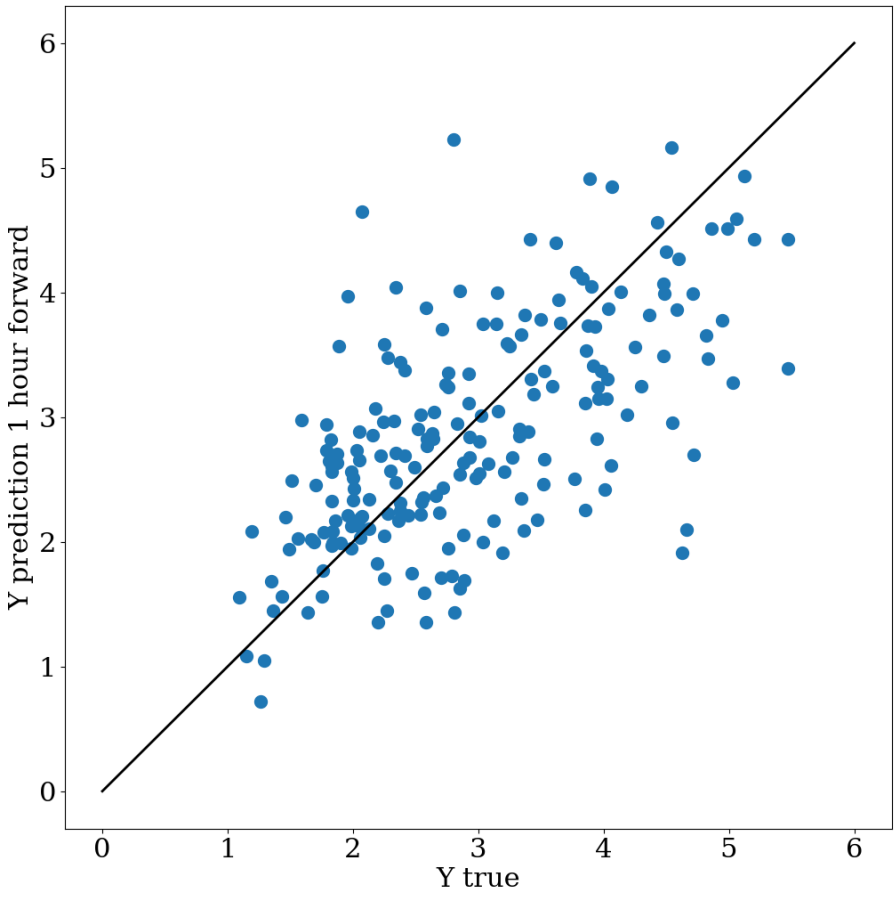

Figure 11: One hour forward predicted values against true values

Variable weights are saved for each step that the window moves through. A matrix is formed from the weights, where each column represents the weights for an individual step. The weight matrix is presented as a heatmap in Figure 12. It can be seen that silica concentrate lag 1 has the greatest amount of high-valued weights. It is difficult to find significant differences between other variables, as they seem to have quite similar amounts of high and low weights. Ore pulp density lag 6 might be the most unimportant variable weight-wise, as it seems to have the most weights that are almost zero.

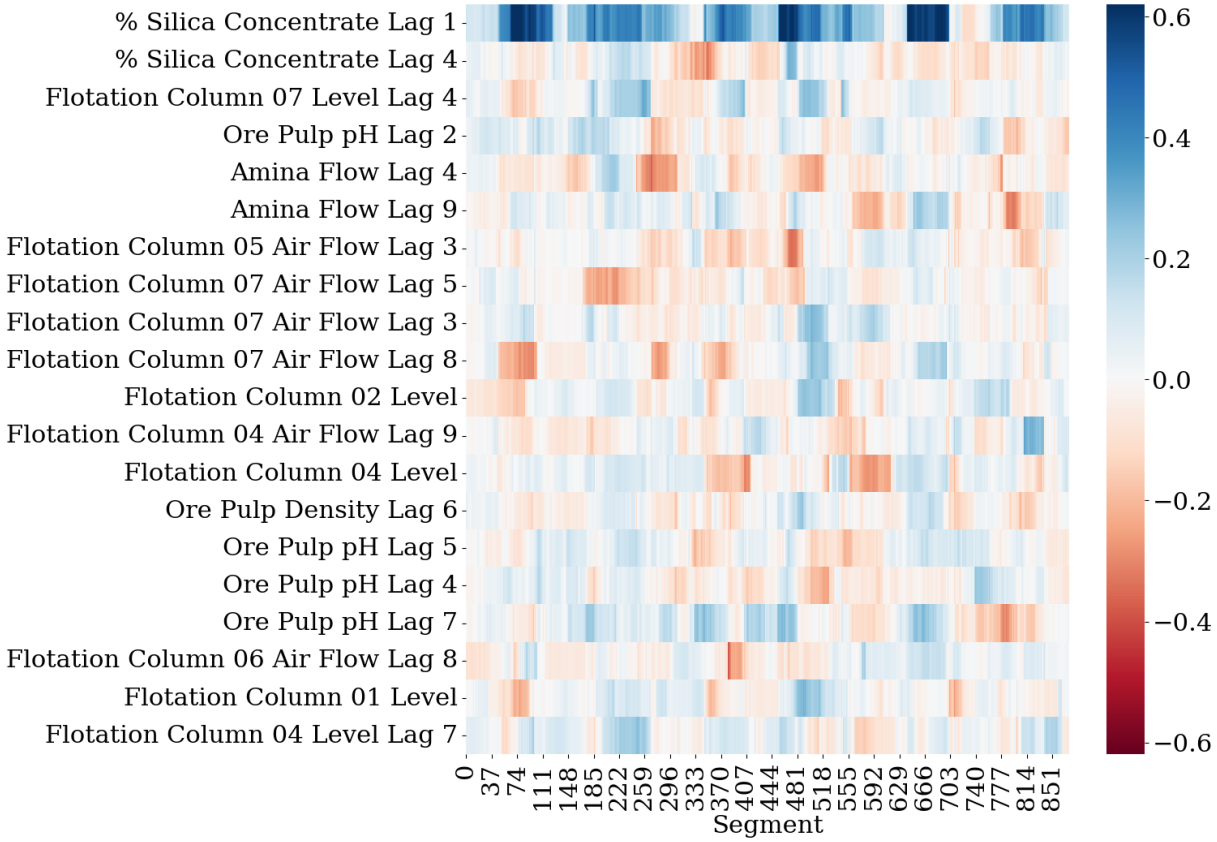

Figure 12: Heatmap of weights of selected variable throughout the travel of the dynamic window.

4 Discussion

Missing values and overall data quality produced some challenges with the analysis. It was quite hard to find a continuous series of observations from the data set, and some missing values might have been overlooked since they are filled with previous valid measurements in the data. Despite that, the overall prediction in the test data produced a reliable prediction of the silica concentration.

Given that we are utilizing lagged variables, there is a slight delay in the prediction after the actual data. As expected, the coefficient for the previous measurement of silica concentration stands out as the most influential by a large margin. A model solely based on this information would probably yield results of comparable accuracy, underscoring the possibility that the poor quality of the gathered data or the unsuitability of the observed variables for this prediction method may be contributing factors.

Variable selection plays a crucial role in the PLS analysis. Adding lagged variables to extend feature space improved results but caused PLS to overfit to train data. Reducing the feature space to a more restricted set of variables reduced overfitting and produced better results. Increasing the window size for the dynamic model also reduced overfitting.

Overall, the model proved to be quite weak. High Q^2 values might confuse since they're mostly dependent on the calibration partition of the data. The test partition is only three measurements, and the calibration data is over ten times bigger, which distorts the results.

As a baseline comparison, we used the model that evaluates the next silica measurement based on the previous one. The model performed similarly on the first-hour prediction and outperformed on the second and third hours. Only the first two hours are shown in the results since they are used for the calibration, but the model is capable of producing results further than that if needed.

References

[1] R. R. Klimpel, "Froth flotation," in *Encyclopedia of Physical Science and Technology (Third Edition)* (R. A. Meyers, ed.), pp. 219–234, New York: Academic Press, third edition ed., 2003.

[2] Y. Pu, A. Szmigiel, and D. B. Apel, "Purities prediction in a manufacturing froth flotation plant: the deep learning techniques," *Neural computing applications*, vol. 32, no. 17, pp. 13639–13649, 2020.

[3] E. M. Oliveira, "Quality prediction in a mining process."