CHALMERS
UNIVERSITY OF TECHNOLOGY

Development of statistical models for optimizing the performance of the electrostatic filter in a waste to energy plant

Master’s thesis in Space, Earth, and Environment

MOOKLADA CHAISORN

Department of Space, Earth, and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

MASTER’S THESIS IN SPACE, EARTH, AND ENVIRONMENT

Development of statistical models for optimizing the performance of the electrostatic filter in a waste to energy plant

MOOKLADA CHAISORN

Department of Space, Earth, and Environment
Division of Energy Technology

CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg, Sweden 2021

Development of statistical models for optimizing the performance of the electrostatic filter in a waste to energy plant
MOOKLADA CHAISORN

Master's Thesis
Department of Mechanics and Maritime Sciences
Division of Energy Technology
Chalmers University of Technology
SE-412 96 Göteborg
Sweden
Telephone: + 46 (0)31-772 1000

Cover:
Lillesjöverket, a municipal waste-fired CHP plant (Source: Uddevalla Energi)

Department of Space, Earth, and Environment
Göteborg, Sweden 2021-01-29

Development of statistical models for optimizing the performance of the electrostatic filter in a waste to energy plant
Master's thesis in Space, Earth, and Environment
MOOKLADA CHAISORN
Department of Space, Earth, and Environment
Division of Energy Technology
Chalmers University of Technology

Abstract

Process data for an electrostatic precipitator (ESP) from Uddevella Energi AB is measured with one-hour resolution. There are 23 predictors which are ash concentrations, steam productions, voltages, currents, flue gas properties (volumetric flowrate, temperature, pressure, oxygen, and water content), and exhaust gas compositions (HCl, CO, NO_X, CO_2 and SO_2). The data is preprocessed by removing outliers using standard deviation method and Mahalanobis distance, resulting in 3 different scenarios (s1, s2, and s3). To avoid overfitting, data in each scenario is split into training and test sets for 7 cases having different amount of data in training and test set (i.e., the training/test set percentages of data were: 50-50, 55-45, 60-40, 65-35, 70-30, 75-25, and 80-20). The main predictive models are linear regression and support vector machines (SVM). Each of them is additionally applied with principal component analysis (PCA) and partial least squares (PLS) for dimensionality reduction. Thus, there are 6 models in total (i.e., Linear regression, Principal component regression (PCR), Partial-least square regression (PLSR), SVM, SVM with PCA, and SVM with PLS). From investigation, scenario 2 with outliers removed by standard deviation method gives the best performance in most cases. For the prediction trend, linear regression, PCR and PLSR models have bad prediction at very low and very high efficiency. With all 23 predictors, SVM with PLS give the best prediction trend among 6 models, and case 65-35 provides the best performance with RMSE of 0.0035, R^2 of 0.86, MARE of 0.26% and MaxARE of 1.45%. Feature selection is performed to improve the models. The best predictor combination to be removed is CO_2, SO_2, H_2O, HCl, CO, O_2wet, P_{in}, and NO_X, leaving 15 predictors for the models. Unusual trend of SVM and SVM with PCA from using all predictors is reduced or even disappeared, while all models get improved when this reduced set of 15 predictors is used. SVM with PCA model gives best performance for all splitting cases with 15 predictors and case 50-50 provides the best indicator values with the lowest RMSE of 0.0029, highest R^2 of 0.9161, lowest MARE of 0.19% and MaxARE of 1.94%. Thus, SVM with PCA model with 15 predictors using scenario 2 and case 50-50 is recommended for ESP efficiency prediction.

Key words: Electrostatic precipitator (ESP), Linear Regression, Support Vector Machines (SVM), Principal Component Analysis (PCA), Partial Least Square (PLS)

Contents

List of figures

List of tables

Acknowledgement

This thesis work was executed in cooperation with Uddevalla Energi AB, Sweden from February to October 2020. The work is a part of research project to develop statistical models to optimize ESP performance in a waste-to-energy plant. The project is carried out at the Department of Space, Earth and Environment in Energy Technology Division, Chalmers University of Technology, Sweden.

During this master’s thesis, I would like to thank several people that without them it would be impossible to accomplish this project. First of all, I would like to thank Professor Stavros Papadokonstantakis who is both my supervisor and examiner. With his knowledge and ability to keep up to date on this work, he steered me in the right direction whenever I was getting stuck or off-track. With his immense experience in academia, I could get insightful and relevant feedback to improve. His interest in this topic, as well as his knowledge and eagerness to discuss ideas, have been invaluable to me. I highly appreciate all his effort to guide and give me full support through all stages of the project.

I also extend my gratitude to Pawel Kak, a process engineer at Uddevalla Energi AB. He is the one who suggest this project and grants me this opportunity to carry on this work. During very productive meetings with Pawel, several interesting ideas for the project were discussed. As well as I get continual support and guidance since the first meeting with the company.

Göteborg November 2021-01-29

MOOKLADA CHAISORN

Notations

Upper case letters

A	Effective collection area (m^2)
C	Box constraint
$D(X, \mu)$	Mahalanobis distance
$G(x_i,x_j)$	Gamma matrix, kernel function
I	Current
L_ε	Epsilon loss function
$L(\alpha)$	Lagrangian loss function
N_{De}	Deutsch number, N_{De} = WA/Q
Q	Flue gas flow rate (m^3/s)
V	Voltage
W	Particle migration velocity (m/s)
X_{ij}	*i*th observation on the *j*th predictor variable
Z	Principal component

Lower case letters

b_k	Fitted coefficients
$f_k(X_{ij})$	Scalar-valued function of the predictor variables X_{ij}.
k	Dimensionless parameter with value from 0.4-0.6
r_i	Residual
y_i	*i*th response
$\hat{y}_i$	Predicted response
$\bar{y}$	Mean response

Greek upper-case letters

Σ	Covariance
Φ^{p1}	Loading vector (Φ^{11}, Φ^{21}, …, Φ^{p1}) of first principal component

Greek lower-case letters

α_n, α_n^*	Lagrange multipliers
β_k	*k*th coefficient
β_0	Constant term in the model
ε_i	*i*th noise term, or random error
ε	Epsilon margin
η	ESP efficiency
μ	Mean of the distribution
ξ_n, ξ_n^*	Slack variables
ρ_d	Apparent dust resistivity
σ	Standard deviation of the distribution
σ_{XY}	Covariance of X and Y
$\varphi(x)$	Transformation that maps x to a high-dimensional space

1 Introduction

As environmental regulations on air quality standards have become increasingly stringent over the past few decades, the removal of particulate matter (PM) entrained in the flue gases from various industrial combustion processes is of vital importance. An electrostatic precipitator (ESP) plays a significant role to do this task. It is the most common and highly efficient device that is used to control and reduce PM suspended in the flue gas stream by mean of electrostatic forces. ESP has ability to treat large gas volume at high removal efficiency up to 99.9% with low pressure drop. To maintain high performance of ESP is important. Thus, it becomes the interest of this thesis on how to bring technology to help this together with day-to-day operation in the plant. The main interest is to use machine learning as a technique to find correlations between operating parameters that influence ESP performance.

Machine learning brings together statistics and computer science to enable computers to learn how to do a given task while not being programmed to do so. The algorithm is a trial-and-error process using computational methods to learn information directly from data and find a model that fits the data as best as possible and then make predictions based on that. For examples, one can use machine learning to predict weather temperatures based on a set of relavant measured data, or predict sales based on many important factors, or even use it for image recognition and fraud detection. There are a lot more industrial applications that it can be applied. With the potential of machine learning, ESP process data can be used to train predictive models and then these models can be used to predict ESP efficiency to ensure that it is operated with good performance or to be aware of bad performance that may occur. More specifically, the models use relative parameters that have an impact on ESP performance as predictor variables for effiency prediction. In this work, the models include observed parameters which are ash concentrations, oxygen and water content, and exhaust gas compositions, together with operating parameters such as gas properties (volumetric flowrate, temperatures, and pressures), and electric field (voltage and current). The models are best to be used for exploratory purposes in order to see that what kind of predictors influence the ESP efficiency in what way and in what extent. In other words, to use these models ones must know the values of the predictor variables and must know under which conditions the ESP is operating at a given moment. The usage of the models is not to predict what will happen in the next moment, but to propose models of operation (i.e., combination of parameters) that the ESP can work better.

The simplest model typically used in engineering applications is linear regression that describes a response as a linear function of one or more predictors. More advanced models such as Support Vector Machine (SVM) can be used for non-linear regression to find deviation from the measured data by a small amount, with parameter values that are as small as possible to minimize sensitivity to error. On the other hand, dimensionality reduction techniques such as Principal Component Analysis (PCA) and Partial least-squares (PLS) are commonly applied when dealing with high-dimensional data. PCA uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of linearly uncorrelated variables. PLS regression is a technique used with data that contain correlated predictor variables. Like PCA, this technique constructs new predictors as linear combinations of the original predictors; however, PLS constructs these new predictors by considering the observed response values. This gives PLS reliable predictive power.

This thesis aims to develop statistical models to predict ESP performance using machine learning algorithms based on supervised learning. The main models are linear regression and support vector machine regression. PCA and PLS are additionally applied to both models to reduce dimensionality as there are 23 predictors used in predicting efficiency. The work is executed using MATLAB (R2019b). The objectives of this study are to investigate the model accuracy and generalizability performance, examine effect of outlier removal methods and the effect of training/testing cross validation, as well as perform sensitivity analysis of operating parameters.

2 Theory

This section aims to introduce the reader to the knowledge required to understand and interpret the results of this work. Firstly, an overview of an electrostatic precipitator (ESP) is presented. Secondly, a general overview, types and techniques of machine learning are presented, followed by detailed description of each model used in this study.

2.1 Electrostatic Precipitator (ESP)

An electrostatic precipitator (ESP) is the most common and highly efficient device that is used to control and reduce PM suspended in the flue gas stream by means of electrostatic forces. ESP has the ability to treat large gas volume at high removal efficiency up to 99.9% with low pressure drop. There are two main components inside an ESP chamber which are high-voltage discharge electrode system and a series of neutral grounded collection plates. The high negative voltage provided by a transformer-rectifier (TR) set is applied to the discharge electrode creating an electric field in the space between the electrode and collection plates (Figure 2.1.1). When the flue gas enters an ESP, the dust particles are charged negatively by mobile ions generated at the high voltage electrode. The electrostatic force created by the electric field on the charged particles results in accelerating of the particles towards the collection plates. The charged particle impacts the collection surface where it sticks and loses its charges. With more and more particles, an ash layer is formed. The layer is removed by rapping the plates, causing the dust to fall into hoppers located below the plates. [1-4]

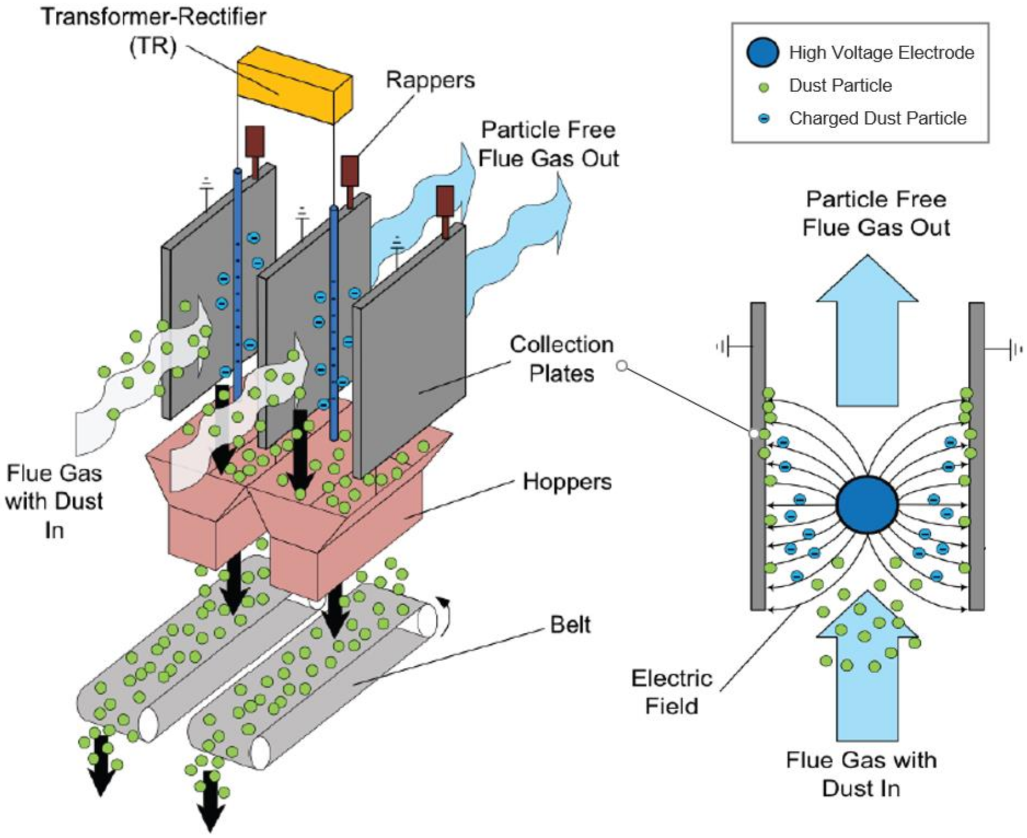


Figure 2.1.1: ESP operating principle and main components [3]

A well-known Deutsch-Anderson equation is the simplest equation for estimating the ESP collection efficiency. It is based on many ideal assumptions which are uniform size distribution and no particle re-entrainment. It considers the particle dielectric constant, but not the resistivity which is the most common factor considered when designing an ESP. Matts-Öhnfeldt equation is a modified Deutsch-Anderson equation which is more commonly used for an ESP design, as shown in Equation (2.1). It includes an additional exponent of a dimensionless parameter k which ranges from 0.4 to 0.6 depending on the dust properties and standard deviation of the particle size distribution. It is used to provide a more conservative estimate of the removal efficiency.[3,5-8]

$$\eta(\%) = \{1 - exp[(-N_{De})^k]\} \times 100\% \qquad (2.1)$$

where
N_{De} is the Deutsch number that is N_{De} = WA/Q
A is the effective collection area (m^2)
Q is the flue gas flow rate (m^3/s)
W is the particle migration velocity (m/s)
k is the dimensionless parameter with value from 0.4-0.6

Apart from the design parameters in Matts-Öhnfeldt equation, a literature review [7] shows that ESP performance can be influenced by several operating parameters such as gas properties (velocity, temperature, density, pressure and humidity), PM properties (size, density, concentration, velocity, shape, adhesivity, resistivity, and dielectric constant) and electric field (voltage, current, and electric field strength). For voltage and current, Equation (2.2) shows the relationship that is often found in many industrial ESP with $n \approx 2$. A higher collection efficiency can be achieved when the ESP is operated at the maximum available voltage. [6,9]

$$\eta = V^n I \qquad (2.2)$$

ESP performance is affected by the apparent dust resistivity (ρ_d) (Figure 2.1.2). For good performance of a dry ESP, ρ_d value should be within 10^2 and $5x10^8$ Ωm. The dust resistivity can be affected by gas temperature, water content, and the gas composition. Usually, the peak value of ρ_d appears at 150-200°C. [6]

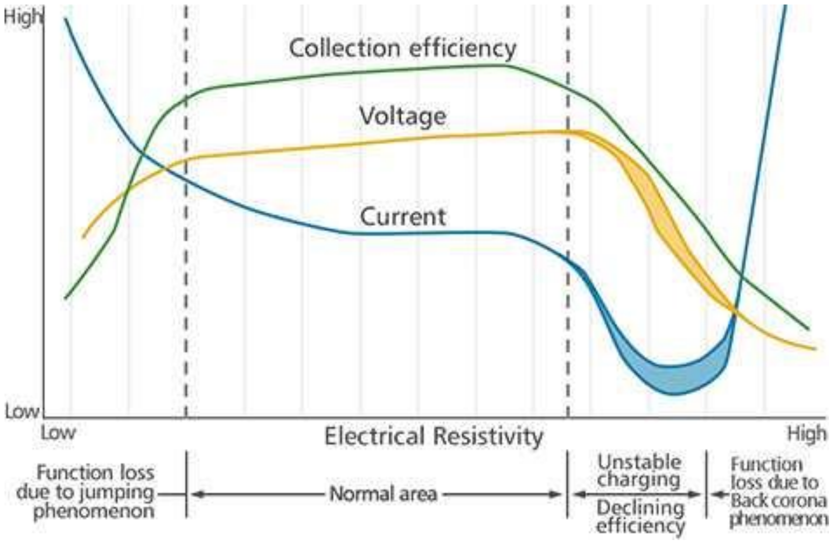


Figure 2.1.2: ESP collection efficiency [10]

More literature reviews [11-13,15] show that ESP efficiency increased with decreasing air flow velocity, increasing of gas temperature, increasing of applied voltage, and decreasing in gas volume. One parameter related to the gas volume is oxygen content. Significant variations in oxygen may indicate large swings in the gas flow rate that may decrease ESP performance. [15] For gas temperature, it can affect the resistivity of the particulate. It can also affect the gas properties to such an extent that they will change the relative levels of voltage and current and the density and viscosity of the gas stream, which affect particle migration parameters. [14-15]

ESP operation depends on electronegative gases (such as oxygen, water vapor, carbon dioxide, and sulfur dioxide/trioxide) to generate an effective corona and to transport the electrons from the discharge electrode to the collection plate. The presence of one or more of these gases is necessary to enhance ESP performance, and the relative level in the gas stream is not always important to ESP operation. In most processes, these gases are available. For CO_2 and O_2, they are often monitored on combustion sources as a measure of excess air and combustion efficiency and not as an indicator of the potential ESP operation. The presence of water vapor and/or acid gases may be useful as resistivity modifiers or conditioners, and they may be necessary for proper ESP performance. On the other hand, they may cause a sticky particulate that is difficult to remove, for example, SO_2 generation in an ESP servicing kraft pulp recovery boilers. Moreover, particle concentrations are usually measured. The difference between the amount of material at the inlet and outlet of the gas streams provides the basis for removal efficiency calculations. [15]

These literature reviews present the importance of each operating parameter as well as how they may relate or affect to one another. In this study, several operating parameters mentioned above are selected to analyze their influence on the ESP collection performance. These parameters are ash concentrations, steam production, voltage, current, volumetric flowrate, temperature, pressure, oxygen and water content, and exhaust gas compositions (HCl, CO, NO_X, CO_2 and SO_2). For simplicity, only parameters that can be measured easily and continuously are taken into account in this thesis work. Since laboratory tests are needed to analyze those important particulate properties such as resistivity and particle size distribution, they are discarded.

2.2 Machine Learning

Machine learning algorithm is a trial-and-error process that use computational methods to learn information directly from data and find a model that fits the data as best as possible. The goal of this model training procedure is to develop a model that can make accurate predictions on new, previously unseen data. The various algorithms adaptively improve the model performance when there are more samples available for learning. There are two types of machine learning which are unsupervised learning, and supervised learning. Various algorithms for each category are shown in Figure 2.2.1.

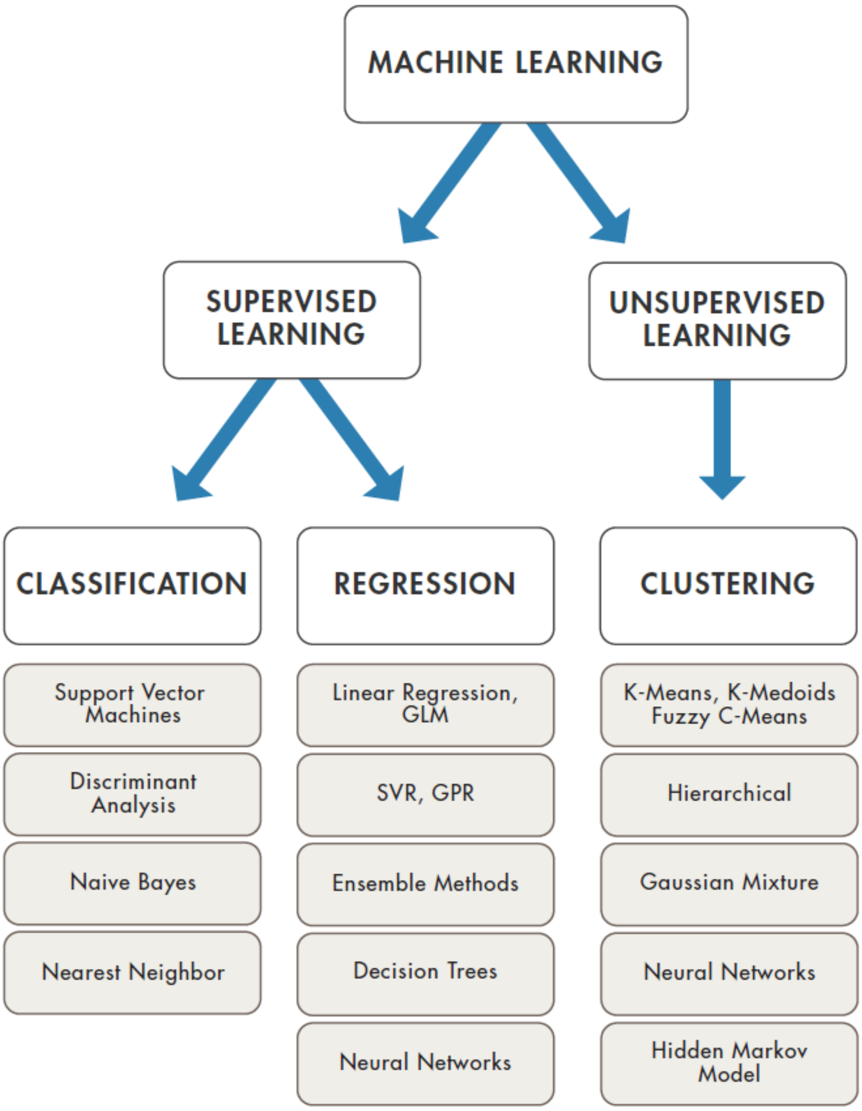


Figure 2.2.1: Machine Learning Algorithms [16]

2.2.1 Unsuperivsed Learning

Unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses. The most common technique for this type of machine learning is *Clustering*. It is used for exploratory data analysis to find groupings in data. Applications for clustering include gene sequence analysis, market research, and object recognition. [16]

2.2.2 Supervised Learning

Supervised machine learning aims to develop a model that makes predictions based on evidence in the presence of uncertainty. Its algorithm takes a known set of input data and known responses of the data (output) and trains a model to generate reasonable predictions for the response to new data, in other words, trains a model on known input and output data so that it can generalize by predicting out of training sample outputs. By comparing the model output to the true output data, the algorithm can improve the statistical model and minimize the error between the two outputs. Supervised learning uses classification and regression techniques to develop predictive models. [16]

Classification techniques predict categorical responses which is the response type that can be labeled to belong on a certain group. Classification models classify input data into categories. Typical applications include medical imaging, image and speech recognition, and credit scoring. *Regression* techniques predict continuous responses such as numerical or signals type of data. Typical applications include electricity load forecasting and algorithmic trading. [16]

This study aims to generate statistical models of the electrostatic filter in a waste to energy plant with high modelling accuracy. Therefore, machine learning algorithms based on supervised learning are selected. As all data is available as continuous variables, regression techniques will be used with the main focus on Linear Regression and Support Vector Machine (SVM). These are further explained in the next section.

2.3 Linear Regression

A linear regression is a model described the relationship between a dependent variable (or response, y) and one or more independent variables (or predictors, X). Suppose that there is a design matrix of n observations on p predictors. A response y_i of the ith observation can be expressed as a function of predictors X_{ij} on that observation. If there is only one predictor in the model (p = 1), it is called a simple linear regression model. If there are more than one predictor, it is known as a multiple linear regression model which can be described as a function shown in a following equation.

$$y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \cdots + \beta_p X_{ip} + \varepsilon_i \qquad i = 1, \ldots, n \qquad (2.3)$$

where

y_i is the ith response
β_k is the kth coefficient
β_0 is the constant term in the model
X_{ij} is the ith observation on the jth predictor variable, j = 1, ..., p.
ε_i is the ith noise term, or random error

In general, a linear regression model can be a model of the form

$$y_i = \beta_0 + \sum_{k=1}^{K} \beta_k f_k(X_{i1}, X_{i2}, \dots, X_{ip}) + \varepsilon_i \qquad i = 1, \dots, n \tag{2.4}$$

where $f_k(X_{ij})$ is a scalar-valued function of the predictor variables, X_{ij}.

The functions $f(X)$ can be in any form including nonlinear functions or polynomials. The linearity in the linear regression models refers to the linearity of the coefficients β_k. That is, the response variable y is a linear function of the coefficients β_k.

The usual assumptions for linear regression models are that the noise terms (ε_i) are uncorrelated, having independent normal distributions of zero mean ($E(\varepsilon_i) = 0$), and constant variance ($V(\varepsilon_i) = \sigma^2$) as shown in equation (2.5) and (2.6). The variance of y_i is the same for all levels of X_{ij} and the responses y_i are uncorrelated.

$$E(y_i) = E\left(\sum_{k=0}^{K} \beta_k f_k\left(X_{i1}, X_{i2}, \dots, X_{ip}\right) + \varepsilon_i\right) = \sum_{k=0}^{K} \beta_k f_k\left(X_{i1}, X_{i2}, \dots, X_{ip}\right) \tag{2.5}$$

$$V(y_i) = V\left(\sum_{k=0}^{K} \beta_k f_k\left(X_{i1}, X_{i2}, \dots, X_{ip}\right) + \varepsilon_i\right) = V(\varepsilon_i) = \sigma^2 \tag{2.6}$$

The fitted linear function is in the form of an equation (2.7)

$$\hat{y}_i = \sum_{k=1}^{K} b_k f_k(X_{i1}, X_{i2}, \dots, X_{ip}) \qquad i = 1, \dots, n \tag{2.7}$$

where $\hat{y}_i$ is the estimated response and b_k are the fitted coefficients

The coefficients are estimated to minimize the mean squared difference between the prediction vector $\hat{y}$ and the true response vector y, that is $(\hat{y} - y)^2/n$. This method is called the method of least squares. Under the assumptions on the noise terms, these coefficients also maximize the likelihood of the prediction vector. In a linear regression model of the form $y = \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p$, the coefficient β_k expresses the impact of a one-unit change in predictor X_j on the mean of the response E(y), provided that all other variables are held constant. The sign of the coefficient gives the direction of the effect. [17,18,19]

2.4 Support Vector Machines (SVM)

Support Vector Machines (SVM) is a supervised learning algorithm which can be used for both classification and regression problems. These are also known as support vector classification (SVC) and support vector regression (SVR), respectively. In this section, detailed description of SVR will be the main focus, as it is used as predictive models in this study.

2.4.1 Support Vector Regression (SVR)

Support Vector Regression (SVR) has the flexibility to define how much error is acceptable in a model. It constructs a hyperplane in multi-dimensional space to fit the dataset in the best possible way with a predefined or threshold error value. [20] SVR is different from a simple regression in a sense that simple regression tries to minimize the error rate while SVR model tries to fit the error within a certain threshold.

Several important terms associated with SVR are presented as follows:
Kernel is a function used to map a lower-dimensional data points into higher dimensional data points. There are many types of kernel such as Polynomial Kernel, Gaussian Kernel, Sigmoid Kernel, etc.
Hyperplane: In SVM, a hyperplane is a line used to separate two data classes in a higher dimension than the actual dimension. In SVR, a hyperplane is a line that is used to predict continuous value.
Support Vector: Data points that lie closest to the boundary. The distance of the points is minimum or least. The support vectors can be on the boundary lines or outside it.
Boundary Line are two parallel lines drawn to the two sides of Support Vector with the error threshold value (***ε***). These lines create a margin between data points.

Characteristics of SVR are usage of kernels, absence of local minima, and capacity control on margin. It contains all the main features that characterize maximum margin algorithm. [21] The Figure 2.4.1 shows an example of one-dimensional linear regression function with epsilon intensive band. Slack variables (ξ) measure the cost of the errors on the training points. These are zero for all points that are inside the band.

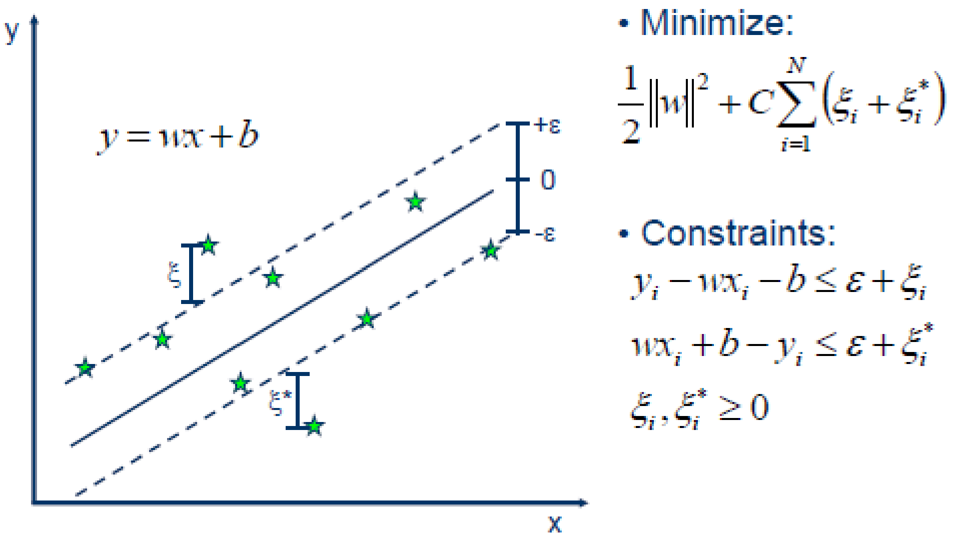

y
$y = wx + b$
+ε
0
-ε
ξ
ξ*
x
• Minimize:
$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$
• Constraints:
$y_i - wx_i - b \le \varepsilon + \xi_i$
$wx_i + b - y_i \le \varepsilon + \xi_i^*$
$\xi_i, \xi_i^* \ge 0$

Figure 2.4.1: One-dimensional linear regression with epsilon intensive band [22]

2.4.1.1 Linear SVM Regression: Primal Formula

For a training data set that has x_n as a multivariate set of N observations with observed response values y_n, the goal of SVR is to find a function $f(x)$ that deviates from y_n by a value no greater than ε for each training point x, and at the same time is as flat as possible. Linear function $f(x)$ with the minimal norm value ($\beta'\beta$) can be in a form (2.8). Together with slack variables, they allow regression errors to exist up to the value of ξ_n and ξ^*_n. This leads to the objective function, known as the primal formula (2.9) [23-24]. SVR is formulated as minimization of the following functional:

$$f(x) = x'\beta + b \tag{2.8}$$

$$J(\beta) = \frac{1}{2}\beta'\beta + C\sum_{i=1}^{N}(\xi_n + \xi_n^*) \tag{2.9}$$

subject to following constraints:

$\forall n$: $y_n - (x_n'\beta + b) \leq \varepsilon + \xi_n$
$\forall n$: $(x_n'\beta + b) - y_n \leq \varepsilon + \xi_n^*$
$\forall n$: $\xi_n^* \geq 0$
$\forall n$: $\xi_n \geq 0$

The constant C is the box constraint, a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin (ε) and helps to prevent overfitting (regularization). This value determines the trade-off between the flatness of $f(x)$ and the amount up to which deviations larger than ε are tolerated.

The linear ε-insensitive loss function ignores errors that are within ε distance of the observed value by treating them as equal to zero. The loss is measured based on the distance between observed value y and the ε boundary. This can be described as:

$$L_\varepsilon = \begin{cases} 0 & \text{if } |y - f(x)| \leq \varepsilon \\ |y - f(x)| - \varepsilon & \text{otherwise} \end{cases} \tag{2.10}$$

2.4.1.2 Linear SVM Regression: Dual Formula

As for the dual formula, a Lagrangian function from the primal function is constructed by introducing nonnegative multipliers α_n and α^*_n for each observation x_n. This leads to the dual formula, where it minimizes

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i'x_j + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N}y_i(\alpha_i^* - \alpha_i) \tag{2.11}$$

subject to the constraints

$\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0$
$\forall n$: $0 \leq \alpha_n \leq C$
$\forall n$: $0 \leq \alpha_n{*} \leq C$

The β parameter can be completely described as a linear combination of the training observations using the equation

$$\beta = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)\,x_n \tag{2.12}$$

The function used to predict new values depends only on the support vectors:

$$f(x) = \sum_{n=1}^{N} (\alpha_n - \alpha_n^*)(x_n' x) + b \tag{2.13}$$

The Karush-Kuhn-Tucker (KKT) complementarity conditions are optimization constraints required to obtain optimal solutions. For linear SVM regression, these conditions are

$$\forall n\colon \alpha_n (\varepsilon + \xi_n - y_n + x_n'\beta + b) = 0$$
$$\forall n\colon \alpha_n^* (\varepsilon + \xi_n^* + y_n - x_n'\beta - b) = 0$$
$$\forall n\colon \xi_n (C - \alpha_n) = 0$$
$$\forall n\colon \xi_n^* (C - \alpha_n^*) = 0$$

These conditions indicate that all observations strictly inside the epsilon tube have Lagrange multipliers $\alpha_n = 0$ and $\alpha_n^* = 0$. If either α_n or α_n^* is not zero, then the corresponding observation is called a *support vector*.

2.4.1.3 Nonlinear SVM Regression: Primal Formula

Some regression problems cannot adequately be described using a linear model. In such a case, the Lagrange dual formulation can be extended to nonlinear functions.

A nonlinear SVM regression model can be obtained by replacing dot product $x_1'x_2$ with a nonlinear kernel function $G(x_1,x_2) = <\varphi(x_1),\varphi(x_2)>$, where $\varphi(x)$ is a transformation that maps x to a high-dimensional space. Table below describes several semidefinite kernel functions. [23]

Table 2.4.1: Kernel functions

Kernel Name	Kernel Function
Linear (dot product)	$G(x_j, x_k) = x_j' x_k$
Gaussian	$G(x_j, x_k) = \exp(-\Vert x_j - x_k \Vert^2)$
Polynomial	$G(x_j, x_k) = (1 + x_j' x_k)^q$, where q is in the set {2, 3, …}

The *Gram matrix* is an n-by-n matrix that contains elements $g_{i,j} = G(x_i,x_j)$. Each element $g_{i,j}$ is equal to the inner product of the predictors as transformed by φ. However, it can use the kernel function to generate Gram matrix directly. Using this method, nonlinear SVM finds the optimal function $f(x)$ in the transformed predictor space. [23]

2.5 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised statistical technique that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables. It is primarily used for dimensionality reduction in machine learning to reduce multidimensional data to lower dimensions while retaining most of the information. [25,26]

Model overfitting is a frequent problem associated with high dimensionality. If a model is overfitting, it loses the ability to generalize other observations beyond the training dataset. This makes it much harder for the model to predict the response correctly when dimensionality of the dataset increases. To avoid such a problem, PCA is applied to remove redundant features so that the model becomes more efficient as PCA helps boosting the learning rates and diminishing computational costs. [25]

PCA transforms an *n*-dimensional feature space into a new *n*-dimensional space of orthogonal components, called principal components. Principal components have axis direction that minimizes projection error and maximizes variance. The total number of principal components generated is equal to the dimensionality of the feature set. They are determined in order of decreasing variance. It means that the first principal component captures most of the variance. The second principal component is the direction of maximum variance which is not accounted by the first component and so on. For a given dataset with *n* observations and *p* predictors (X^1, X^2..., X^p), the principal component can be expressed as follows:

$$Z^1 = \Phi^{11}X^1 + \Phi^{21}X^2 + \Phi^{31}X^3 + \cdots + \Phi^{p1}X^p \qquad (2.14)$$

where Z^1 is first principal component. Φ^{p1} is the loading vector comprising of loadings (Φ^{11}, Φ^{21}, …, Φ^{p1}) of first principal component. The loadings are constrained to a sum of square equals to 1 as large magnitude of loadings may lead to large variance. It also defines the direction of the principal component (Z^1) along which data varies the most. It results in a line in *p* dimensional space which is closest to the *n* observations. Closeness is measured using average squared Euclidean distance. X^1, X^2, …, X^p are normalized predictors that have mean of zero and standard deviation of one. Similarly, the second principal component can be computed from (2.14) by replacing Z^2 and Φ^{p2} (Φ^{12}, Φ^{22}, …, Φ^{p2}) into the equation. [27]

Figure 2.5.1 shows an example of principal components in two-dimensional data. The blue dots indicate original data. The black vectors are principal components generated by applying PCA. The size of the vectors indicates how much variance is explained by that component. Since the two components are orthogonal to each other, it means that they are uncorrelated. [27]

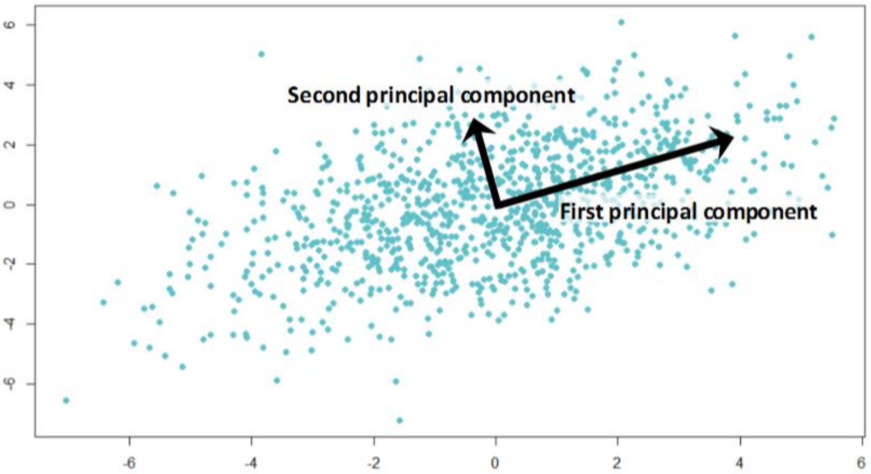


Figure 2.5.1: Principal components for two-dimensional data [27]

PCA reduces dimensionality by discarding the principal components beyond a chosen threshold of explained variance. In general, the threshold can be 90-95% depending on the data. It aims to capture the maximum amount of variance with the fewest number of components. For example, suppose that the dataset has *n* observations with *p* predictor variables. The corresponding principal components will be in total of *p* axes. As shown in Figure 2.5.2 below, the principal components are plotted with their explained variance, but here the first 9 (out of *p*) principal components explain more than 95% variance of the data. Since the transformed variables contain almost the same amount of information as in the original data, the rest components can be discarded if the chosen threshold is 95%. In such a case, it would be assumed that the components that contain the last few percent of explained variance are likely to represent noise more than information.

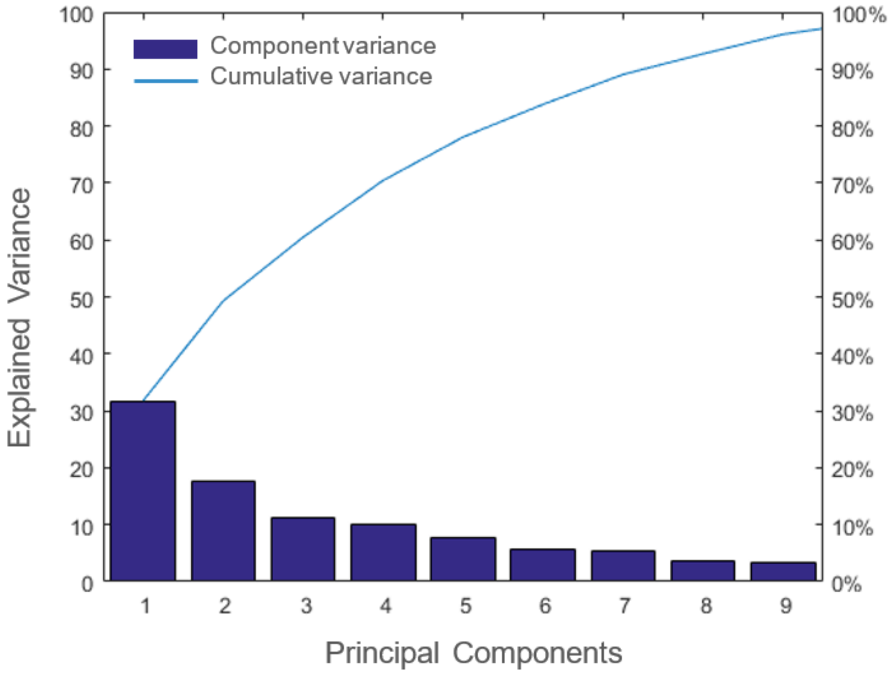


Figure 2.5.2: Explained variance of principal components in high-dimensional data

For a matrix X of *n* observations and *p* predictors, the dimensionally reduced form is given by:

$$z^{(i)} = \Phi_{reduce}^{T} x^{(i)} \tag{2.15}$$

To reduce dimensionality of data, the first *k* columns of the *n* x *n* matrix Φ are selected to form *n* x *k* matrix Φ_{reduce}. Since Φ_{reduce}^{T} is *k* x *n* matrix and $x^{(i)}$ is *n* x 1 vector, the product $z^{(i)}$ is *k* x 1 vector with reduced dimensions. The approximate reconstruction in the higher dimension can be computed from the following equation, giving $x^{(i)}_{approx}$ as *n* x 1 vector with the original number of dimensions.

$$x_{approx}^{(i)} = \Phi_{reduce} \cdot z^{(i)} \tag{2.16}$$

In order to determine the number of principal components that are retained during the dimensionality reduction, the following two metrics are considered. The objective of PCA is to minimize the projection error given by (2.17) and the total variation in the data is given by (2.18).

$$\frac{1}{p}\sum_{i=1}^{p}\left\| x^{(i)} - x_{approx}^{(i)} \right\|^2 \tag{2.17}$$

$$\frac{1}{p}\sum_{i=1}^{p}\left\| x^{(i)} \right\|^2 \tag{2.18}$$

The rule of thumb is, choose the smallest value of *k*, such that,

$$\frac{\frac{1}{p}\sum_{i=1}^{p}\left\| x^{(i)} - x_{approx}^{(i)} \right\|^2}{\frac{1}{p}\sum_{i=1}^{p}\| x^{(i)} \|^2} \le 0.01\ (or\ 1\%) \tag{2.19}$$

That is 99% of the variance is retained. As a consequence, the number of dimensions reduced is significant since many features are highly correlated. Generally, values such as 95−90% variance retention are used. [28]

2.6 Partial Least Square (PLS)

Partial least-squares (PLS) regression is a technique used with data that contain correlated predictor variables. This technique constructs new predictor variables, known as components, as linear combinations of the original predictor variables. PLS constructs these components while considering the observed response values. This gives PLS reliable predictive power. [29,30]

Partial least-squares (PLS) is different from multiple linear regression and PCA that it takes response values into account. Multiple linear regression finds a combination of the predictors that best fit a response. PCA finds combinations of the predictors with large variance, reducing correlations. However, PLS finds combinations of the predictors that have a large covariance with the response values. Thus, PLS combines information about the variances of both the predictors and the responses, while also considering the correlations among them. [29]

Partial least squares (PLS) model is based on the principal components on both the independent data and the dependent data. The idea is to find the principal scores of $X \in \mathbb{R}^{n\times p}$ and $Y \in \mathbb{R}^{n\times q}$ and use them to build a regression model between the scores. PLS regression is based on the basic latent component decomposition and can be expressed as [30,31]:

$$X = TP^T + E \tag{2.20}$$

$$Y = TQ^T + F \tag{2.21}$$

The matrix X is decomposed into two matrices, $T \in \mathbb{R}^{n\times d}$ which is the matrix that produces d linear combinations (scores) and $P^T \in \mathbb{R}^{d\times p}$ which is matrix of coefficient referred as X-loading, plus an error matrix $E \in \mathbb{R}^{n\times p}$. Similarly, Y is decomposed into T, $Q^T \in \mathbb{R}^{d\times q}$ (Y-loadings) and $F \in \mathbb{R}^{n\times q}$ (random errors). The matrix T often denoted as 'latent variables' or 'scores' is estimated as the linear combinations as follows [30,31]:

$$T = XW \tag{2.22}$$

where W are referred as the weights. There are many different approaches of finding W. One of that is the statistically inspired modification of PLS (SIMPLS). The criterion of SIMPLS is stated as:

$$w_j = \underset{w}{\operatorname{argmax}}\, w^T \sigma_{XY} \sigma_{XY}^T w \tag{2.23}$$

subject to $w^T w = 1, w^T \Sigma_{XX} w_j = 0$, for j = 1, …, k-1
where w_j are the columns of W and σ_{XY} is the covariance of X and Y.

When T is estimated, loadings are estimated by ordinary least squares for the model (2.21). The regression matrix for PLS is formulated as

$$\beta^{PLS} = WQ^T \tag{2.24}$$

Since

$$Y = TQ^T + F = XWQ^T + F = X\beta^{PLS} + F \tag{2.25}$$

The latent components are then used for prediction in place of the original variables: once T is constructed, Q^T is obtained as the least squares solution of Equation (2.21):

$$Q^T = (T^T T)^{-1} T^T Y \tag{2.26}$$

and the fitted response matrix $\hat{Y}$ may be written as

$$\hat{Y} = T(T^T T)^{-1} T^T Y \tag{2.27}$$

For an uncentered raw observation $x_o{}'$, the prediction $\hat{y}_0'$ of the response is given by:

$$\hat{y}_0' = \frac{1}{n}\sum_{i=1}^{n} y_i{}' + B^T \left(x_o - \frac{1}{n}\sum_{i=1}^{n} x' \right) \tag{2.28}$$

2.7 Outlier Detection

Observed data usually are multidimensional that has higher chance of having unusual observations. The problem is that a few outliers is always enough to distort the results of data by altering the mean performance, increasing variability, etc. Therefore, outlier detection should be of concern. In this section, outlier detection by standard deviation method and by Mahalanobis distance are mainly discussed.

2.7.1 Mean and Standard Deviation Method

One of the most simplest statistical tools for outlier detection is the Z-score, which the mean and standard deviation of the residuals are calculated and compared. Z-score indicates how far the value of the data point is from its mean for a specific feature. A Z-score with value of 1 means that the data point is 1 standard deviation away from its mean. Typically, Z-score values greater than +3 or less than -3 are considered outliers. [32] Z-score can be expressed as follows:

$$Z\ score = \frac{x_i - \mu}{\sigma} \tag{2.29}$$

where σ is the standard deviation and μ is the mean of the distribution of feature x, and x_i is the value of the feature x for the ith sample.

However, this method can fail to detect outliers since all the outliers increase the standard deviation. The more extreme the outlier, the more the standard deviation is affected. Figure 2.7.1 below shows outlier detection by this method in which data points within 3 standard deviations are remained.

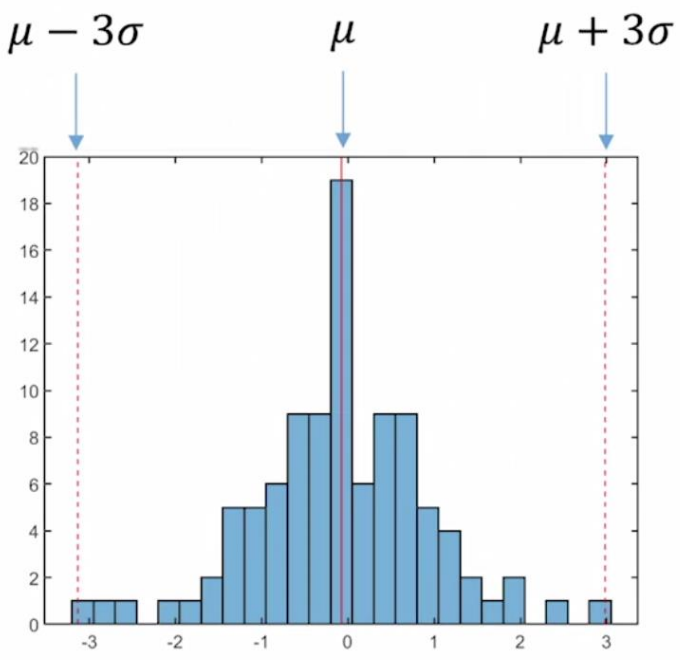


Figure 2.7.1: Outlier detection by mean and standard deviation method

Outlier detection based on simple statistical tools generally assume that the features have normal distributions while neglecting the correlation between features in a multivariate dataset. More advanced method for outlier detection based on machine learning can handle correlated multivariate dataset, detect abnormalities within them, and do not assume a normal distribution of the features.[32] One method is to use Mahalanobis distance as explained in the following section.

2.7.2 Mahalanobis Distance

From geometric point of view, the Euclidean distance is the shortest possible distance between two points. However, the Euclidean distance measure does not consider the correlation between highly correlated variables. It assigns equal weight to such variables. Consequently, correlated variables get excess weight by Euclidean distance.

An alternative approach is Mahalanobis distance that scale the contribution of individual variables to the distance value according to the variability of each variable. This approach differs from Euclidean distance as it considers the correlations between variables. The Mahalanobis distance is a measure between a sample point and a distribution which represents how far x is from the mean in number of standard deviations. This measure can be used to detect outliers if there are any outliers that do not behave as normal as usual observations at least in one dimension. The Mahalanobis distance from a vector x to a distribution with mean μ and covariance Σ is defined by the following equation (2.30). If the covariance matrix is the identity matrix, the Mahalanobis distance reduces to the Euclidean distance. [33]

$$D(X, \mu) = \sqrt{(X-\mu)^T \Sigma^{-1} (X-\mu)} \tag{2.30}$$

2.8 Statistical Indicators

In this section, the regression metrics that are commonly used when evaluating regression models are presented. The equations use y_i to refer to the actual response values of the model, where i = 1, 2, …, n, and $\hat{y}_i$ to refer to the model's predicted response values. The value n is the number of observations in the data set, and $\bar{y}$ is the mean of the actual response values.

Residual (r_i) is the model error for each data point.

$$r_i = y_i - \hat{y}_i \tag{2.31}$$

Mean Absolute Error (MAE) is the average magnitude of the residuals. This is an easy-to-interpret metric that has the same units as the response.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2.32}$$

Mean Square Error (MSE) is the average of the squared residuals. Most types of regression will minimize this term to train the model. Because of the squaring term, it is more sensitive to large errors and outliers than the MAE.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.33}$$

Root Mean Square Error (RMSE) is the square root of the MSE. It has the same units as the response (like MAE), but also emphasizes large errors and outliers (like MSE). Ideally, this should be as close to 0 as possible.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.34}$$

Sum of squared errors (SSE) is the sum of the squared residuals (as opposed to the average value MSE). Used to calculate R^2.

$$SSE = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.35}$$

Sum of squares total (SST) is a measure of the variance of the data points ($\bar{y}$ is the mean response). Used to calculate R^2. It is as an error metric when the "model" is a simple baseline model that always uses the mean as the predicted value.

$$SST = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{2.36}$$

R^2 is the relative difference in the total error obtained by fitting a model. If a model fits the data well, the model error is small and R^2 will be close to 1. If the model fits the data poorly, then the model error is large and R^2 will be close to 0. This metric is also called the Coefficient of Determination.

$$R^2 = \frac{SST - SSE}{SST} \tag{2.37}$$

Mean Absolute Percentage Error (MAPE) is the average relative error, reported as a percentage. It measures how large the residuals are relative to the scale of the data, e.g., if the MAPE is 20%, the model predictions are off by an average of 20%. Ideally, it should be as close to 0 as possible.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{2.38}$$

3 Methodology

This chapter presents the machine learning workflow used in this thesis work as well as the procedure to implement different models for ESP efficiency predictions. The platform used in this work is MATLAB (R2019b), which has several built-in functions and applications for training models.

3.1 Machine Learning Workflow

This project aims to build and develop predictive models that are able to predict ESP efficiency (response variable) from several measured operating parameters (predictors). Supervised machine learning with regression is selected and used to create and train models. Machine learning workflow used in this study is shown in the Figure 3.1.1 below. Note that the word “parameters”, “predictors” and “features” may be used interchangeably in the later part of the report. The same thing also goes to “efficiency” and “response”.

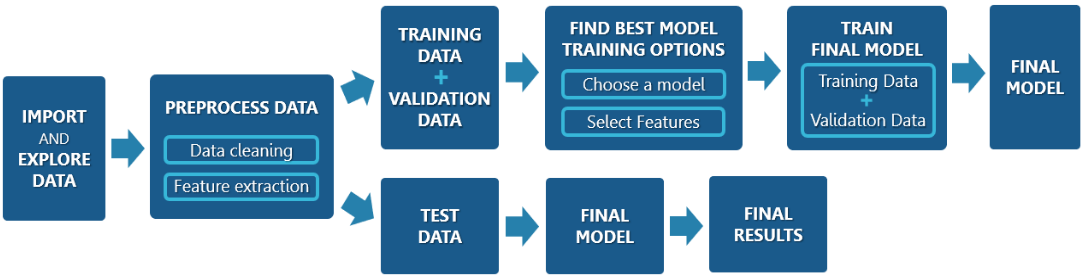


Figure 3.1.1: Machine learning workflow

First of all, the workflow starts by importing, exploring, and preprocessing data. The data is visualized to see correlation between parameters and cleaned by removing outliers. After that, the cleaned data will be split into training, validation, and test data sets. The training data is used during training models along with the validation data that is used to prevent model overfitting. Then, the trained models are used to predict the response by using the test dataset as an input to see if the trained model can generalize with the new dataset that was never used during training and validation. However, the training process to find the best model is an iterative process. Several models may be chosen to train and see if they give good results as well as only relevant features may be selected if they provide better prediction and less errors. After these steps, the final and selected models are compared by using various statistical indicators. More details for each step will be further explained in the later sections.

3.2 Data

Selecting data to analyze is an important step. The more data of good quality we have, the better for the analysis and prediction. In this study, the ESP process data was measured and collected by process engineers from a waste to energy power plant, Uddevalla Energi AB, in Sweden. The data is obtained from 30 November 2019 until 30 April 2020. It has one-hour resolution which means that each observation is measured every hour during the day. Therefore, there are in total of 3672 observations from a total of 153-day measurement.

As mentioned earlier, several process parameters are selected according to either of their direct or indirect effect on efficiency, and only parameters that can be measured continuously in one-hour basis are chosen for simplicity. There are in total 24 parameters to consider. One of them is the efficiency which is a response variable, and the rest 23 variables are called predictors. Thus, the data matrix has the size of 3672 observations (rows) and 24 features (columns). The relevant process parameters with units and variable names are listed in Table 3.2.1 below. Noted that variable names are used as a short form in MATLAB codes and figures.

Table 3.2.1: ESP process parameters

Parameters	**Units**	**Variable name**
Concentration of ash	mg/Nm^3	Ash_in, Ash_out
Efficiency	%	Efficiency
Steam Production	kg/s	Steam, RealSteam
Voltage	kV	V1, V2, V3
Current	mA	I1, I2, I3
Volumetric flowrate	Nm^3/h	Q
Temperature	°C	T_in, T_out
Pressure	mbar	P_in, P_out
O2wet, O2dry, H2O	%	O2wet, O2dry, H2O
Exhaust gas composition	mg/Nm^3	HCl, CO, NOX, CO2, SO2

Some parameters such as ash concentration, temperature, and pressure are measured at the inlet and outlet of the ESP. Thus, their names are followed by “in” or “out” to indicate the location. The ESP at the plant has double chambers and each chamber has three stages along the gas flow direction. Operating parameters such as voltage and current are measured at each stage, therefore, they are named according to which stage they were measured. For steam production, there are two related variables named as “Steam” and “RealSteam”. Actually, “Steam” is referred to steam set point that is an input set in the system. “RealSteam” is referred to real steam production which is an output from the system that should be as close to the set points as possible, but it is also influenced by many other parameters in the process.

3.3 Data Visualization and Preprocessing

After we collect and prepare all the data of interest, they are ready for data analysis. Firstly, all necessary and important data is imported and explored. This step is crucial as we can see how the data trends look like by exploring through different kind of visualization such as histograms and other plots. It may show interesting trends, for example, extreme outliers can be noticed easily and thus must be removed. Missing data and known error on certain dates are removed as well. This step is called data cleaning. Moreover, some correlation between predictors may also be seen. The data are further preprocessed and normalized. Different outlier removal methods are applied to the cleaned data. There are 3 different scenarios to consider depending on how outliers are removed, which are as follows:

- scenario 1: no outliers removed
- scenario 2: outliers removed by standard deviation method
- scenario 3: outliers removed by Mahalanobis distance

Scenario 1 serves as a base case for the other two. In all scenarios, the dataset was cleaned by removing observations with missing data and known error. From investigation, ESP has malfunctioned on certain dates. This results in too high efficiency being measured during these days which are 11-13 February, 6 April, and 25 February during 11.00-15.00. Thus, these observations are removed, resulting in 3568 rows of observations left. For more convenient interpretation, all 24 parameters are plotted and compared to each other for each scenario (see Appendix A).

In scenario 2, outliers are removed by standard deviation method using +/-3 standard deviations. It is important to exclude outliers so that they cannot affect statistical values such as mean and variance as well as prediction results. However, this method of removing outliers is more suitable for data with normal distribution. Several predictors such as steam set point, oxygen and water content, and exhaust gas compositions are more likely to have constant values, giving narrow and high peaks for histograms without having a normal distribution. Meanwhile, voltage and current tend to have skewed distribution rather than normal distribution. Therefore, this method discards quite a lot of observations leaving 2948 observations for this scenario. According to Appendix A, better trends can be seen for each parameter where those extreme outliers were removed. However, it clearly shows that data in consecutive dates during March was removed as shown in Figure 3.3.1.

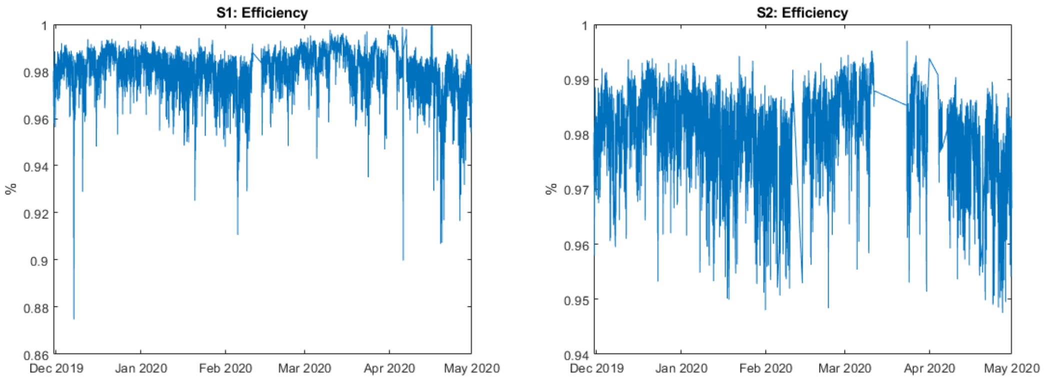


Figure 3.3.1: Efficiency plots for scenario 1 and 2

In scenario 3, outliers are removed by Mahalanobis distance. This method is different from the standard deviation method in scenario 2. The standard deviation method would discard the whole row of observation as an outlier if there is just one or more predictor value outside +/- 3 standard deviation of itself. However, the Mahalanobis distance method (i.e., using *mahal* function in MATLAB) would tell whether the whole row is an outlier compared to other rows and it scales the contribution of individual variables to the distance value according to the variability of each variable. It also considers the correlations between variables. As a result, less observations were discarded in this method, resulting in 3390 observations left. Comparing Figure 3.3.1 and Figure 3.3.2, it clearly shows that several outliers were removed in scenario 3 compared to scenario 1, and less data were removed during March compared to scenario 2. It can be noticed that wider range of data remaining in scenario 3.

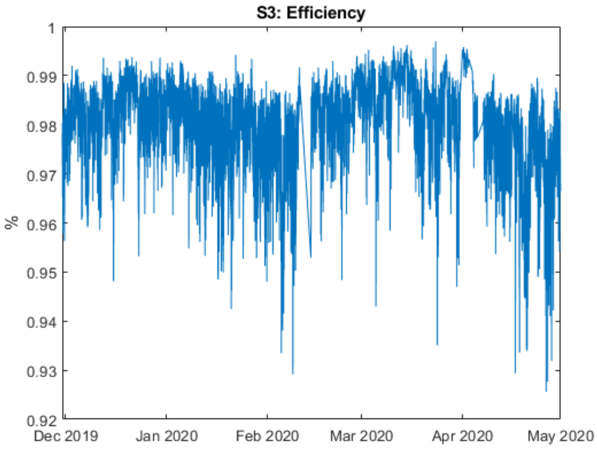


Figure 3.3.2: Efficiency plot for scenario 3

3.4 Data Splitting

After outlier removal, data for each scenario are mainly split into training set and test set. Now, there are 3 different scenarios of outlier removal. Data from each scenario is split into training and test sets for 7 cases (i.e., 50-50, 55-60, 60-40, 65-35, 70-30, 75-25, and 80-20). Noted that these numbers represent how much data is split into either training or test sets. The first number indicates splitting percentage for training set whereas the second number is for test sets. For example, the 80-20 case means that 80% of the data will be training set and the rest 20% will be the test set.

For each splitting case, there are two ways of splitting for either interpolation or extrapolation performance. For interpolation performance, data is split randomly regardless of their chronological order so that it has enough information to interpolate when predicting the response. However, for extrapolation, data is split by chronological order. That is, data from the first few months would fall into training set while data from the later months would be the test set. In this case, it is useful for predicting the future values such as predicting ESP efficiency in the following months. Models will be tested with the test set which is data from the future that the model had never seen and been trained before. This is to see if the model is able to extrapolate when predicting the response. Data from each splitting case is randomly split for 100 files in the case of interpolation and is also split by date to 1 file for extrapolation case. Thus, there are in total of 3 x 7 x 101 = 2121 preprocessed data files.

3.5 Predictive Modelling

Now, all preprocessed data is available and ready to use for training predictive models. Predictive modeling is a technique used to predict an event or outcome with the help of an equation-based model that describes the phenomenon under consideration. The model is trained for a number of states, expressed as combination of predictor values, and to be used other combinations of the predictor values that may occur in the future. The model parameters help explain how model inputs influence the outcome. In this study, several models were trained using Regression Learner App in MATLAB. The program requires training dataset as an input. All operating parameters are specified as predictors and efficiency is the response of interest. In addition, the program has a function that users can decide whether validation shall be taken into account. Here, cross-validation is selected with *k*-fold technique, where *k* is set to be 5 as default. This technique is one of the most popular techniques for cross validation. It partitions part of the training set into *k* folds and uses each fold to validate the model trained using the remaining folds. The process is repeated *k* times so that each fold is used once for validation. However, it can take a long time to execute as the model needs to be trained repeatedly. Thus, *k* value shall not be set too high for high-dimentional data or advanced models to avoid the problem.

In this study, the focused and selected models are linear regression, and support vector machines (SVM). **Linear regression** describes a continuous response variable as a linear function of one or more predictor variables. It is simple, fast to train, easy to interpret, and often the first model to be fitted to a new dataset. It is best used as a baseline model for evaluating other, more complex, regression models. Meanwhile, **Support Vector Machines (SVM)** is a more advanced model that find deviation from the measured data by a small amount, with parameter values that are as small as possible to minimize sensitivity to error. It is best used for high-dimensional data. Using high dimensional data when training can lead to complex models that overfit the data. It means that a model exactly predicts the training data (overfitting) but generalizes poorly to new data like test sets. Cross-validation technique is used to avoid this. The two focused models are additionally applied with principal component analysis (PCA) and partial least square (PLS) for dimensionality reduction so that only relevant information is retained in the models. Therefore, this results in total of 6 models as follows.

- Linear Regression
- Linear Regression with PCA, or Principal Component Regression (PCR)
- Linear Regression with PLS, or Partial Least Square Regression (PLSR)
- Support Vector Machines
- Support Vector Machines with PCA
- Support Vector Machines with PLS

PCR and PLSR are methods to model a response variable when there are a large number of predictors, that are highly correlated or even collinear. Both methods construct new predictors, known as components, as linear combinations of the original predictors, but in different ways. PCR creates components to explain the observed variability in the predictors, without considering the response at all. On the other hand, PLSR does take the response into account, and therefore often leads to models that are able to fit the response with fewer components.

Next, the trained models will use test sets as a new input to predict the response. Since not all parameters that we have will be useful, sensitivity analysis of predictors will be conducted on these models to see which predictors are best used for predicting efficiency. This is the concept of feature selection which is part of model improvement. Lastly, all models are compared using statistical indicators to see the model performance. It should be noted that indicators from training sets will be averaged values from 100 random splits.

4 Results

Prediction performance of ESP efficiency is compared and discussed on the effect of different outlier removal methods, models, and splitting cases as well as the effect of operating parameters will be presented.

4.1 Correlation

This section mainly discusses the correlation between each parameter which are presented in a form of correlation matrix (Figures 4.1.1-4.1.4). The correlation matrix shows the histogram of each parameter in the diagonal and the linear correlations for all pairs. Moreover, Pearson correlation coefficient are also presented which can tell how strong the pair is correlated. The correlation can be used as a tool for feature selection as we can compare the strengths of the linear relationships between the predictors and the response and discard weakly related features. Moreover, relatively strong correlations between predictors justify the use of PCA and PLS for dimentionality reduction. The value close 1 indicates strong positive linear relationship, and positive value means that one parameter tends to increase with another one. The value close to -1 indicates strong negative linear relationship, and negative value means that one parameter tends to increase when another one decreases. If the value close to 0, it indicates weak or no linear relationship.

The correlation between efficiency and all predictors is mainly focused. Obviously, the efficiency is greatly correlated with ash concentrations. The lesser the outlet ash, the better the efficiency. Other correlated predictors that have coefficients range from 0.3-0.4 are V1, V2, V3, I1, and I2. The rest of predictors seem to have very weak or no linear relationship with efficiency as shown in Figure 4.1.1 to Figure 4.1.4. This is because these predictors tend to have mostly constant values at a certain level or at zero, or they are being controlled variables in the process. Thus, ash concentrations, voltages and currents seem to be the most important predictors. More investigation on this will be further discussed in the later sections.

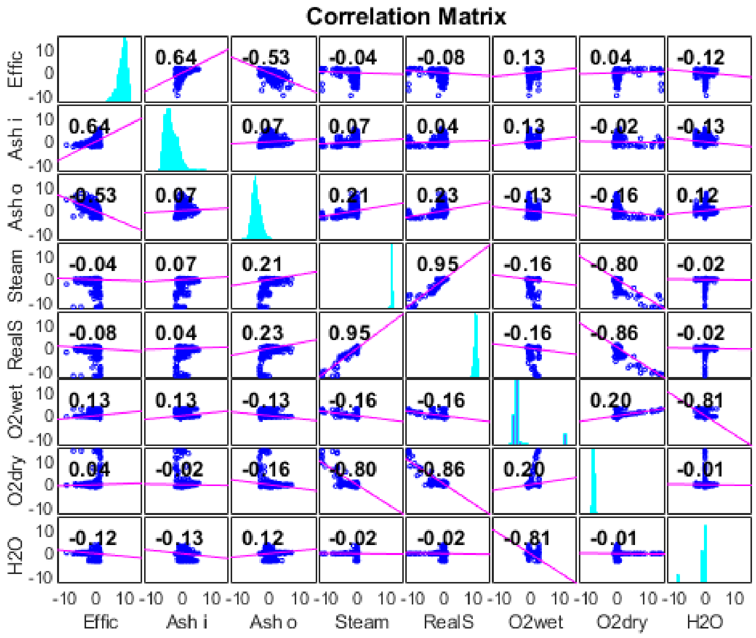


Figure 4.1.1: Correlation matrix of efficiency, ash concentration, steam production, oxygen, and water content from scenario 1

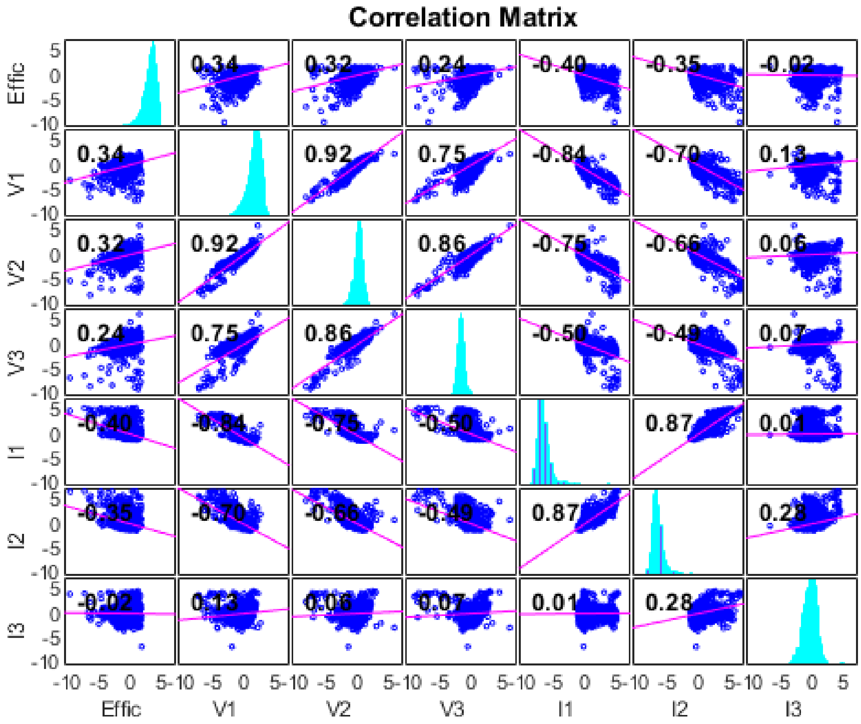


Figure 4.1.2: Correlation matrix of efficiency, voltages, and currents from scenario 1

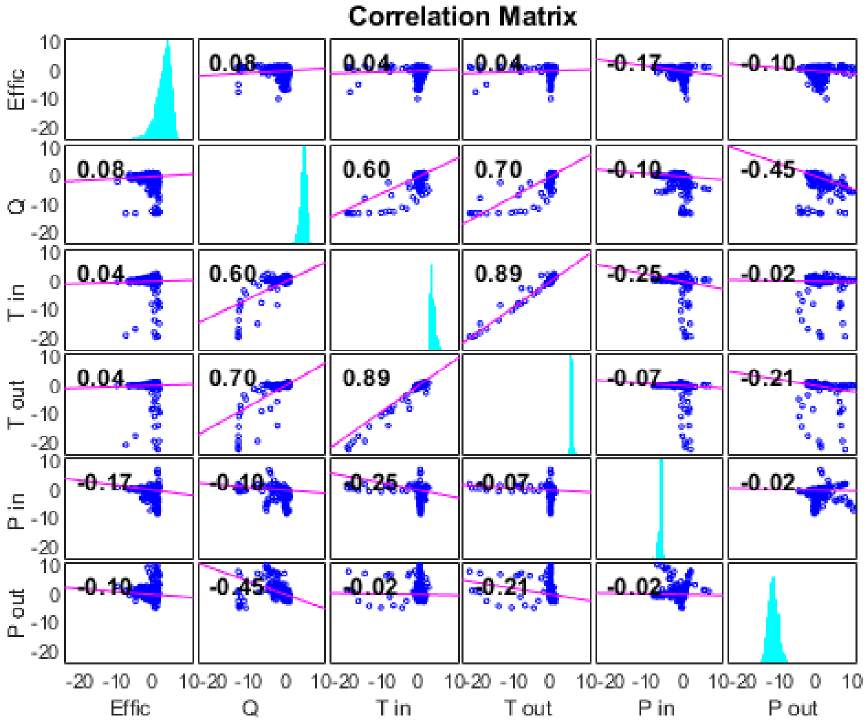


Figure 4.1.3: Correlation matrix of efficiency, volumetric flowrate, temperatures, and pressures from scenario 1

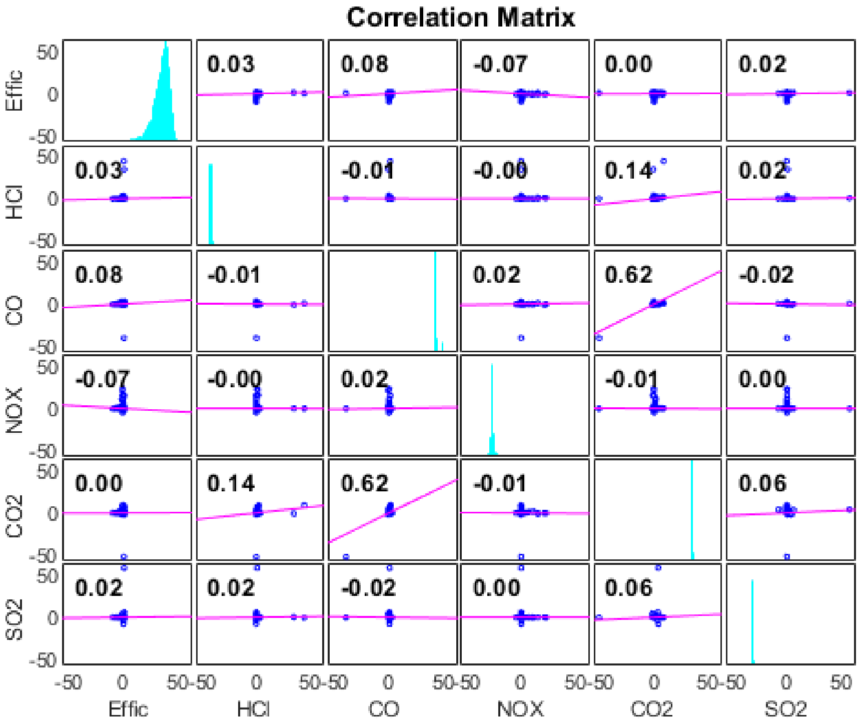


Figure 4.1.4: Correlation matrix of efficiency and exhaust gas from scenario 1

4.2 Effect of Outlier Removal Method

Different outlier removal methods are firstly compared and discussed here to see the effect on efficiency prediction. For the ease of comparison, the interpolation performance is selected to see the general trend of the model. For all 3 scenarios, both training and test sets were used to predict the response and they are compared to see how well the model can generalize with new data like test sets. The dataset presented here is from 50-50 splitting case (interpolation file 1) with linear regression models and the presented efficiency is in terms of true and predicted response (Figure 4.2.1).

With linear regression model, all 3 scenarios show the same trend but differ in term of outliers. Obviously, scenario 1 has the most outlier as only extreme outliers and known error were removed. Some data point from test sets are outside the trend, especially at very low efficiency that the prediction is scattered and not very accurate. Scenario 2 and 3 show quite similar results, both of them having much less outliers. However, scenario 3 is more similar to scenario 1 that is less accurate at the very low efficiency. Among the 3 of them, scenario 2 seems to perform the best prediction which is aligned well with training set with high accuracy at both very low and high efficiency. However, this has trade-off with discarding a lot of data points. It should be noted that similar trends with respect to the best outlier removal method is also observed for other splitting cases and models.

In addition to the trend, statistical indicators can be used to tell the prediction performance and to confirm the results (Table 4.2.1). There are 4 main indicators of interest which are RMSE, R^2, MARE, and MaxARE. Noted that MARE is computed in the same way as MAPE in Equation (2.38), but it is reported in fraction not percentage. Since the table presents values for both training and test sets, indicator values of the test sets is more focused for predicting out of sample values. Firstly, all scenarios show small error, RMSE, which ideally should be close to 0. For R^2, ideally it should be close to 1 as much as possible. However, all 3 scenarios using linear regression do not present that good trend as the R^2 value is only around 0.73 up to 0.85. For MARE and MaxARE, they show how much the model predictions are off by an average. Ideally, they should be as close to 0 as possible, where MaxARE shows the possible maximum error. Scenario 2 gives the best results for all these statistical indicators with the lowest RMSE, highest R^2, and lowest MARE and MaxARE. This indicates that if linear regression model with scenario 2 is to be used for predicting future efficiency, it would give on average an error of 0.24%. For example, if the true efficiency is performed at 98%, this model would predict in the range of 97.76 – 98.24% in average, which is an acceptable range. However, with the MaxARE as high as 0.0174, it is also possible that sometime the model may predict the response off by 1.74% with respect to the predicted value of 98%. This results in the largest possible range of 96.29 – 99.71%, which is more than acceptable. This much off prediction is mainly presented at both very low and very high efficiency.

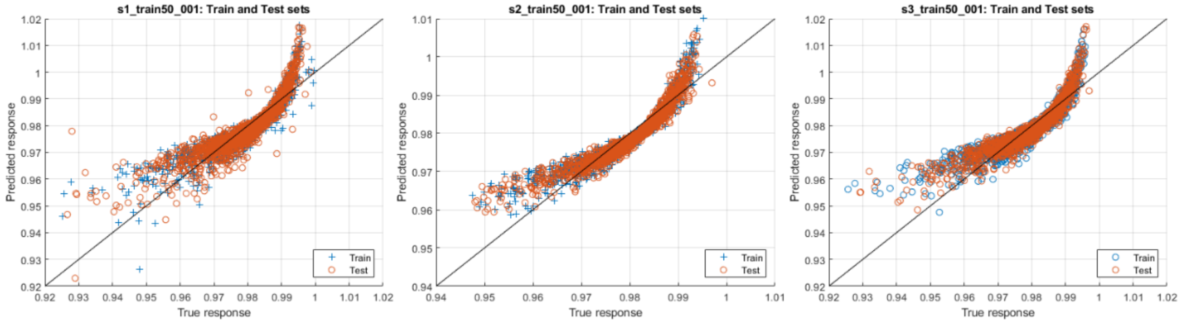


Figure 4.2.1: Comparison on different outlier removal methods with linear regression model and splitting case 50-50

Table 4.2.1: Statistical indicators for linear regression (Case 50-50 interpolation)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0048	0.0056	0.0033	0.0032	0.0043	0.0042
R^2	0.8038	0.7291	0.8518	0.8546	0.8016	0.8127
MARE	0.0033	0.0035	0.0025	0.0024	0.0031	0.0030
MaxARE	0.0434	0.0901	0.0172	0.0174	0.0332	0.0332

4.3 Prediction Trends

This section aims to compare prediction trend from simple and advanced models, which are linear regression and SVM, respectively. The test sets of all 3 scenarios were used to predict the response, and they are compared to each other.

4.3.1 Linear Regression and SVM

The dataset presenting here is also from 50-50 splitting case (interpolation file 1) with linear regression and SVM models (Figure 4.3.1). The prediction trend can be seen from how well the dataset matches with the diagonal line. The more data lies on the diagonal line, the better the model performance. Linear regression model seems to lose ability to predict the response at the boundaries, i.e. at very low and very high efficiency. The model tends to predict the response beyond the possible maximum value which is at 1 or 100%. Although the model has a good prediction on the response in the middle range, predicting beyond maximum point seems to be the big disadvantage of using linear regression. However, SVM shows much better performance as most predicted data lies very well on the diagonal line except at the very low value but only in small portion. Moreover, SVM predicts the response within the limit and it seems to be very accurate for high efficiency, which is the range of most interest.

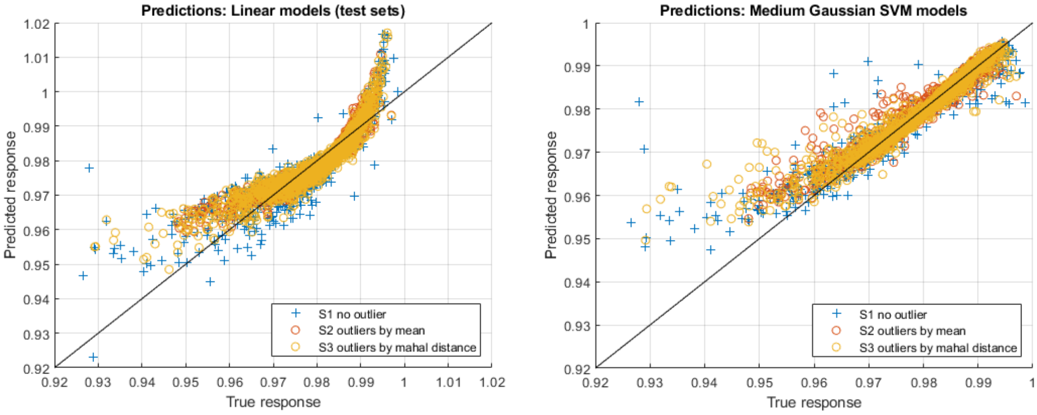


Figure 4.3.1: Case 50-50 interpolation on linear regression and SVM

The same statistical indicators for SVM model are also presented in Table 4.3.1. By comparing Table 4.2.1 and Table 4.3.1, SVM model presents much better results than that of linear regression for most cases. Similarly, SVM model also shows that scenario 2 gives the best values for all indicators. It provides lower RMSE, with R^2 as high as 0.96 for training set and 0.92 for test set. The model also results in as low as 0.15% MARE, but in a bit larger MaxARE of 1.86%. This large error can be found in Figure 4.3.1 having few points scattering in the middle range.

Table 4.3.1: Statistical indicators for SVM (Case 50-50 interpolation)

Statistical Indicators	Data set and Scenario					
	Train_s1	Test_s1	Train_s2	Test_s2	Train_s3	Test_s3
RMSE	0.0034	0.0047	0.0018	0.0025	0.0027	0.0031
R^2	0.9143	0.8211	0.9633	0.9242	0.9393	0.9142
MARE	0.0014	0.0017	0.0011	0.0015	0.0013	0.0016
MaxARE	0.0621	0.1062	0.0154	0.0186	0.0305	0.0306

4.4 Effect of Models

Earlier section presented prediction trend of only linear regression and SVM models on the interpolating prediction performance. However, for the models to be more useful, we are more interested in predicting future ESP efficiency based on data we have. Thus, extrapolation performance using several models is discussed. Figure 4.4.1 shows six prediction plots from all six models which are linear regression, PCR and PLSR on the top row as well as SVM, SVM with PCA and SVM with PLS on the bottom row. All models were performed using all 23 predictors and the data representing in this figure is from a splitting case 65-35. This case can present the best trend among other cases. More details on effect of splitting case will be further discussed in the later section. Appendix B presents the results for the rest of the splitting cases.

From Figure 4.4.1, SVM with PLS give the best prediction trend among 6 models as most of data points align well with the diagonal line. This can be confirmed with the statistical indicators, by comparing Table 4.4.1 to Table 4.4.6. This model results in R^2 as high as 0.86 with 0.26% MARE and MaxARE 1.45%.

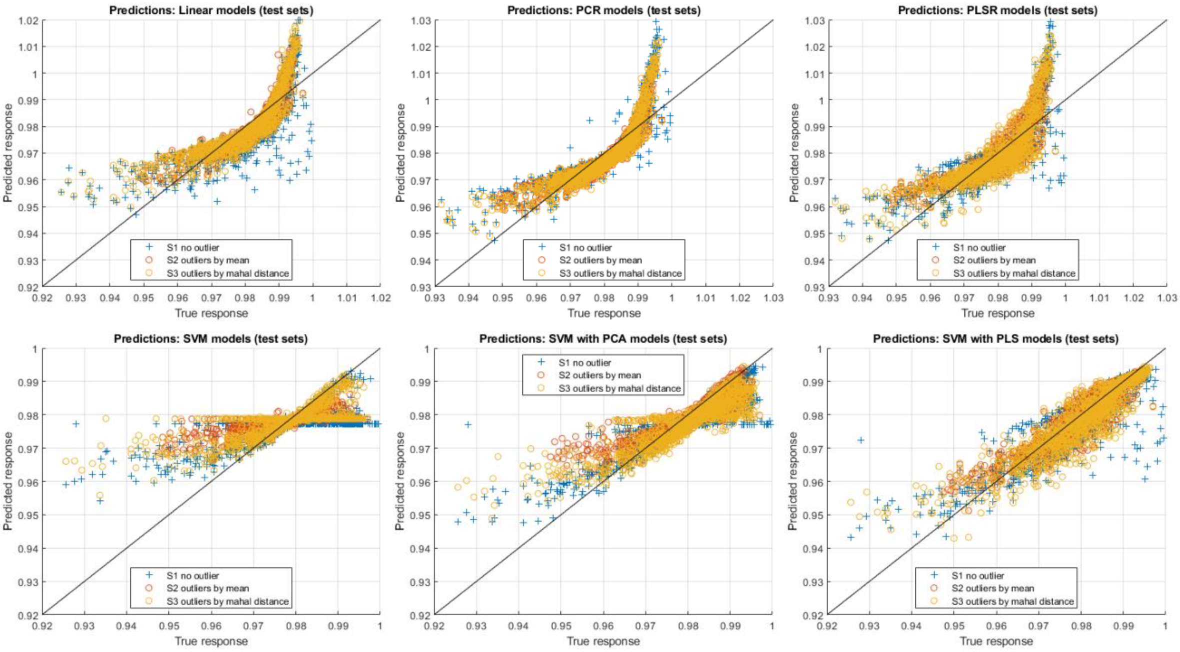


Figure 4.4.1: Extrapolation performance on several models of case 65-35 (Linear, PCR, PLSR, SVM, SVM PCA, and SVM PLS)

4.4.1 Linear Regression, PCR, PLSR

In this section, 3 linear regression base models are compared to each other to see how PCA and PLS techniques improve the model performance. From Figure 4.4.1, all 3 models show very similar trend, and it is hard to distinguish between them. However, all of them predict the efficiency in some cases as high as 102-103% which is impossible in reality. Both PCR and PLSR were specified with 95% of explain variance. For PCR, there are 12 components left in scenario 1, and 14 components left in scenario 2 and 3 to explain this 95% variance. For PLSR, a lot less components were used to explain 95% variance which are 2 components in scenario 1, and 3 components in both scenario 2 and 3.

For statistical indicators, scenario 2 gives the best results for all 3 models (Table 4.4.1 to Table 4.4.3). Comparing indicator value from test sets in scenario 2, it shows that PCA has the best performance with the lowest RMSE, highest R^2 and lowest MaxARE. However, MARE is the same for linear regression and PCR. This means that by applying PCA to linear regression model can help improve marginally the model performance. Nonetheless, applying PLS seems to give the opposite as it results in the worst results among the 3 models.

Table 4.4.1: Statistical indicators for linear regression (Case 65-35)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0036	0.0082	0.0028	0.0047	0.0031	0.0065
R^2	0.8610	0.6107	0.8830	0.7536	0.8750	0.6962
MARE	0.0024	0.0058	0.0021	0.0038	0.0022	0.0048
MaxARE	0.0655	0.0693	0.0159	0.0178	0.0265	0.0343

Table 4.4.2: Statistical indicators for PCR (Case 65-35)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0036	0.0082	0.0028	0.0045	0.0032	0.0065
R^2	0.8563	0.6382	0.8813	0.7755	0.8697	0.7063
MARE	0.0025	0.0052	0.0021	0.0038	0.0023	0.0049
MaxARE	0.0677	0.0978	0.0159	0.0172	0.0276	0.0336

Table 4.4.3: Statistical indicators for PLSR (Case 65-35)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0053	0.0087	0.0037	0.0053	0.0044	0.0077
R^2	0.7098	0.5988	0.7980	0.6977	0.7593	0.6219
MARE	0.0033	0.0062	0.0028	0.0041	0.0033	0.0058
MaxARE	0.0795	0.0831	0.0180	0.0200	0.0348	0.0358

4.4.2 SVM with PCA and PLS

In this section, 3 SVM based models are compared to each other to see how PCA and PLS techniques improve the model performance. From Figure 4.4.1, it obviously shows that SVM with PLS gives the best prediction trend. There is an unusual trend of horizontal line prediction for both SVM and SVM with PCA which did not appear with interpolation performance. These horizontal lines are results of using all 23 predictors as an input although not all of them significantly affect the response. The models took every predictor equally significant when training, nevertheless, some of them were zero for a certain period, such as oxygen and water content as well as exhaust gas compositions (Appendix A). In addition, it could be the results from some predictors that have totally different trend in training and test set, for instance having fluctuation in training set but measured as zero in test set (e.g., as in the case of NOx). More details will be furthered discussed with parameter sensitivity analysis in later section.

Similarly, number of components for SVM with PCA to explain 95% variance are the same as PCR which is 12 components left in scenario 1, and 14 components left in scenario 2 and 3. For SVM with PLS, less components were used which are 2 components in scenario 1, and 3 components in both scenario 2 and 3. For statistical indicators, scenario 2 gives the best results for all 3 models, as shown in Table 4.4.4 to Table 4.4.6. It also shows that applying either PCA or PLS to SVM model can help improving the performance. SVM with PLS has the best performance for most of the indicators.

Table 4.4.4: Statistical indicators for SVM (Case 65-35, 23 predictors)

Statistical Indicators	Data set and Scenario					
	Train_s1	Test_s1	Train_s2	Test_s2	Train_s3	Test_s3
RMSE	0.0032	0.0098	0.0015	0.0077	0.0019	0.0085
R^2	0.8986	0.5184	0.9714	0.5348	0.9594	0.5433
MARE	0.0011	0.0068	0.0009	0.0062	0.0010	0.0061
MaxARE	0.0990	0.0862	0.0161	0.0287	0.0257	0.0468

Table 4.4.5: Statistical indicators for SVM with PCA (Case 65-35, 23 predictors)

Statistical Indicators	Data set and Scenario					
	Train_s1	Test_s1	Train_s2	Test_s2	Train_s3	Test_s3
RMSE	0.0028	0.0064	0.0014	0.0047	0.0018	0.0062
R^2	0.9246	0.7982	0.9735	0.8009	0.9651	0.7777
MARE	0.0009	0.0038	0.0008	0.0033	0.0010	0.0047
MaxARE	0.0889	0.0858	0.0157	0.0247	0.0239	0.0352

Table 4.4.6: Statistical indicators for SVM with PLS (Case 65-35, 23 predictors)

Statistical Indicators	Data set and Scenario					
	Train_s1	Test_s1	Train_s2	Test_s2	Train_s3	Test_s3
RMSE	0.0049	0.0063	0.0025	0.0035	0.0034	0.0051
R^2	0.7524	0.7642	0.9023	0.8594	0.8535	0.8092
MARE	0.0024	0.0042	0.0019	0.0026	0.0025	0.0038
MaxARE	0.0841	0.0838	0.0103	0.0145	0.0237	0.0303

4.5 Effect of splitting cases

From the previous section, SVM with PLS gives the best result on both prediction trend and statistical indicators. This section will compare several splitting cases for this model to see the effect of splitting case on model performance. Figure 4.5.1 shows all splitting cases of SVM with PLS models, while corresponding results for all other models are provided in Appendix B. Statistical indicators of this model for all splitting cases are presented in Table 4.5.1. The values are from test sets in scenario 2 since they provide the best results. Appendix C provides the corresponding values for all other scenarios.

Figure 4.5.1 shows that when training set percentage becomes higher such as case 70-30, 75-25 and 80-20, the prediction trend will be off the diagonal line, especially at very low efficiency. As a consequence, they could not perform well on the statistical indicators as well. The case that provides the best prediction trend and indicator values is 65-35. It provides the lowest RMSE, MARE of 0.26% and MaxARE of 1.45% (Table 4.5.1).

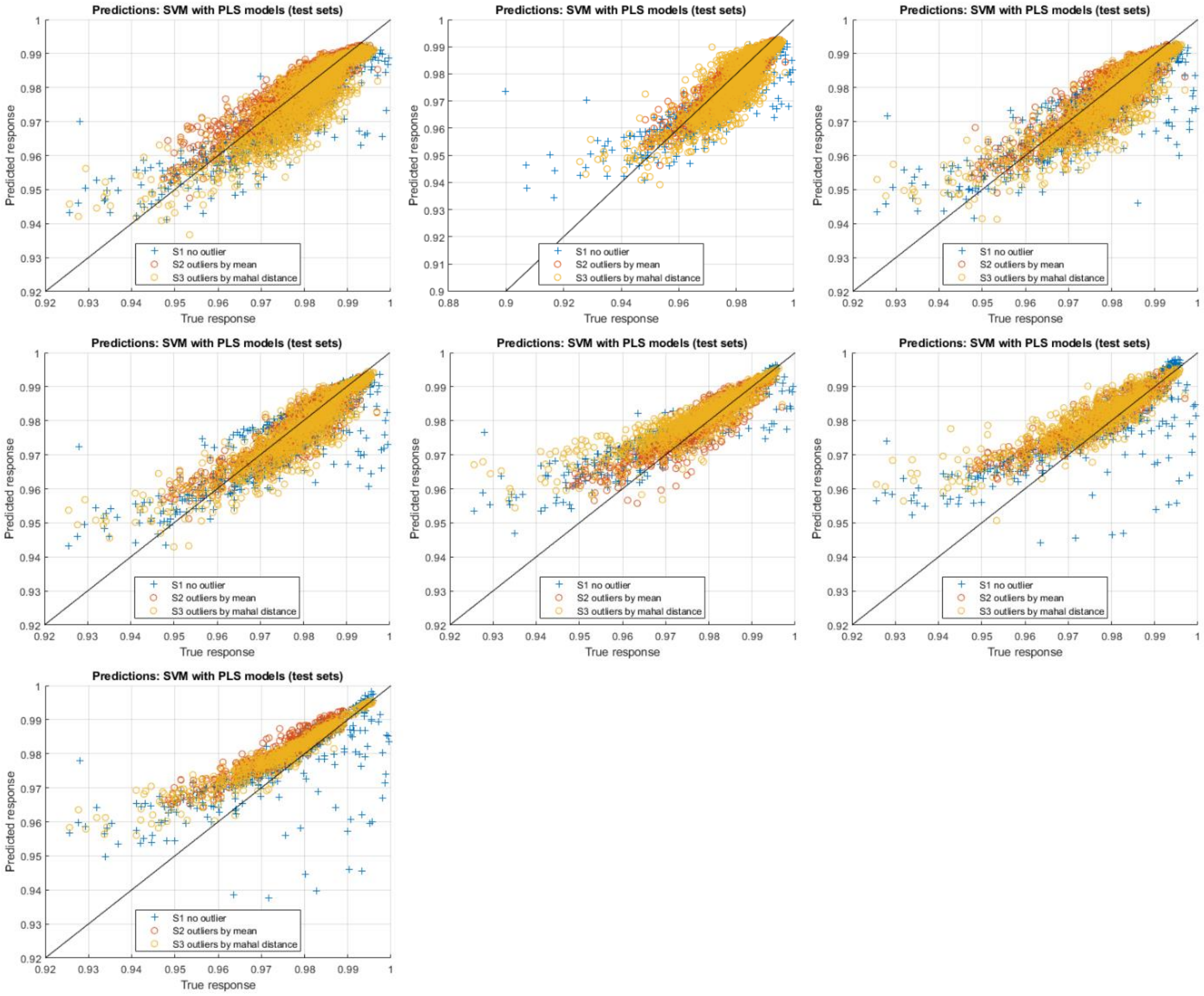


Figure 4.5.1: SVM with PLS performance on all splitting cases; a) top row: 50-50, 55-45, 60-40, b) middle row: 65-35, 70-30, 75-25, c) bottom row: 80-20

Table 4.5.1: Statistical indicators for SVM with PLS (test set in Scenario 2)

Statistical Indicators	**Splitting cases**						
	50-50	**55-45**	**60-40**	**65-35**	**70-30**	**75-25**	**80-20**
RMSE	0.0041	0.0039	0.0041	0.0035	0.0039	0.0057	0.0069
R^2	0.8000	0.8208	0.8074	0.8594	0.8491	0.8718	0.8937
MARE	0.0032	0.0031	0.0032	0.0026	0.0029	0.0045	0.0059
MaxARE	0.0169	0.0155	0.0209	0.0145	0.0167	0.0204	0.0221

4.6 Effect of parameters

In this section, sensitivity analysis of each parameter will be performed to see their effect on the efficiency. Since not all predictors have a significant impact on the response, this analysis can help to discard some of them that may be irrelevant. For the ease of sensitivity analysis performance, SVM models that previously show unusual trend will be used. The splitting case 50-50 of scenario 2 will be used for sensitivity analysis since it has the most data points on test set.

Firstly, test set with all 23 predictors was used to predict the response. The result show unusual trend as a horizontal line as shown in Figure 4.6.1. It indicates that no matter how ESP is operated, the model will always predict the same value of response as constant around 98%, while in reality the ESP efficiency would significantly fluctuate. Therefore, the model should be improved by removing some predictors.

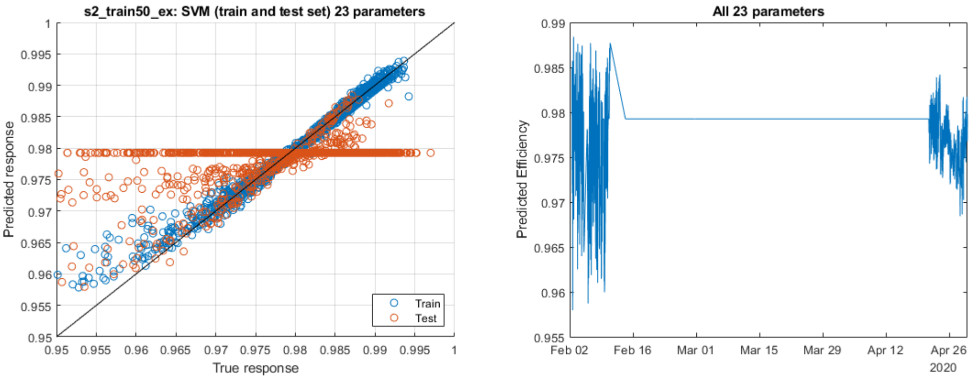


Figure 4.6.1: Predicted efficiency using scenario 2, SVM and 23 predictors

Next, data trend for each operating parameter is investigated (Appendix A). It shows that some parameters such exhaust gas compositions (H_2O, HCl, CO, NO_X, CO_2 and SO_2) have significantly different trends in training and test sets which may affect the model testing. For H_2O, it has mostly constant value at the beginning, then, it starts drastically fluctuating during late February to mid of March and then it remains zero before getting back to the same level during late April. This fluctuation and zero measurement are in the test sets. Other exhaust gas compositions such as HCl, CO, CO_2 and SO_2 have very similar trend to each other. Mostly, they are measured as zero, but only during late February until mid of March fluctuate, which appears only in test set. Meanwhile, NO_X gives an opposite trend. There is fluctuation of NOx in training set, but for the test set, it remains mostly constant at certain level since the beginning of March and it starts fluctuating again in late April. Therefore, the first try to remove some predictors will start from these exhaust gas compositions.

Now, several combinations of predictors are removed to test the model as well as each predictor is removed one by one to see how it affects the response prediction. The first try is to remove a combination such H_2O and NO_X. Unfortunately, they do not effect on the constant predicted output but only results in slightly higher of the last output peak during late April (Figure 4.6.2a). Next, steam production parameter was removed, but it does not seem to have any effect (Figure 4.6.2b). By further removed SO_2, it gives better result with more fluctuated prediction in the later part, which is during April (Figure 4.6.2c). Next, the rest exhaust gas compositions such as CO_2, CO and HCl were tested by removing one by one. When CO_2 is further removed, it also gives better results with much less constant prediction. There is more fluctuation in the middle and the last output peak (Figure 4.6.2d). However, removing CO shows little to no effect on the response as it slightly changes the middle peak (Figure 4.6.2e). When HCl is removed, it gives better result with more fluctuation of the peak during mid of February (Figure 4.6.2f). From this analysis, we see that removing some predictors such as SO_2, CO_2 and HCl affect the efficiency the most whereas removing H_2O, NO_X, CO, and Steam seemed to have little to no effect.

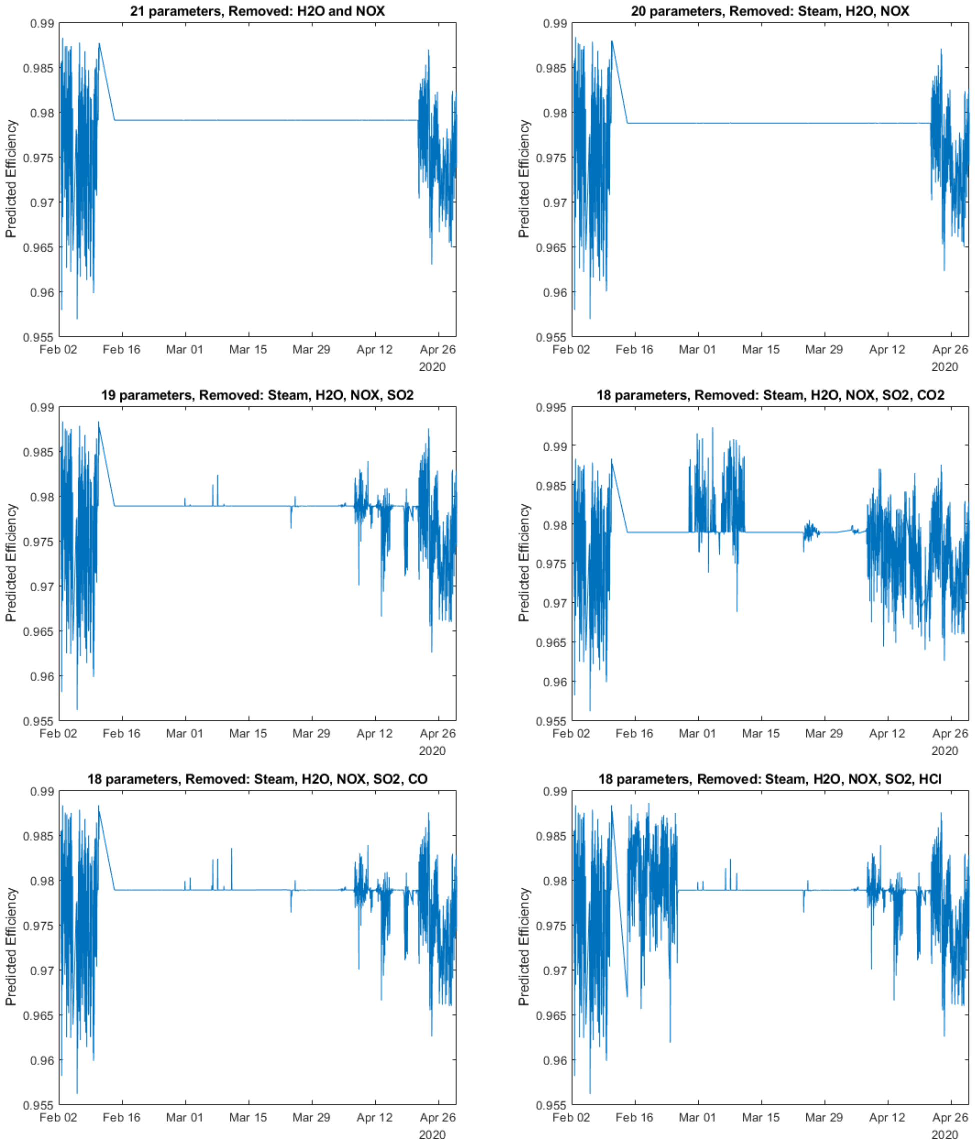


Figure 4.6.2: Predicted efficiency on SVM (s2 case 50-50) when removing: a) H_2O and NO_X, b) Steam, H_2O, and NO_X, c) Steam, H_2O, NO_X and SO_2, d) Steam, H_2O, NO_X, SO_2 and CO_2, e) Steam, H_2O, NO_X, SO_2 and CO, f) Steam, H_2O, NO_X, SO_2 and HCl

The next focus is to remove simultaneously more than one predictor that affect the response the most. The chosen combination is CO_2 and SO_2. Although, these two previously give better result, removing just two of them does not make any good prediction. There is slightly change of middle peak (Figure 4.6.3a). By further removing H_2O, it gives much better result with less constant output and more fluctuation, especially during the beginning of March and mid of April (Figure 4.6.3b). When HCl is further removed, it also gives better results with less constant prediction; there is output fluctuation during the middle of February (Figure 4.6.3c). Next, CO is further removed; the middle peak at the beginning of March fluctuates more (Figure 4.6.3d).

Furthermore, when wet oxygen content parameter (O_2wet) is removed, it gives better result with more fluctuation of middle peak at the beginning of March. It can be seen that the second and third peak are now connected (Figure 4.6.4a). Next, when inlet temperature (T_{in}) is removed, it only results in slightly change of the middle peak (Figure 4.6.4b). Removing inlet pressure (P_{in}) also gives better result with more fluctuation of the middle peak during late March (Figure 4.6.4c). Lastly, NO_X is removed with the combination of CO_2, SO_2, H_2O, HCl, CO, O_2wet, and P_{in}. By removing this combination, it provides more fluctuation of the last output peak (Figure 4.6.4d). Other predictors are also tested, but removing them does not affect the output and thus the model is not improved. These predictors are dry oxygen content (O_2dry), steam productions (both set points and real production: Steam and RealSteam), volumetric flowrate (Q), outlet pressure (P_{out}), outlet temperature (T_{out}), voltages (V_1, V_2, and V_3), and currents (I_1, I_2, and I_3).

Statistical indicators are calculated and used to confirm whether removing such combinations help improving the model performance. Table 4.6.1 presents 4 indicator values which are RMSE, R^2, MARE and MaxARE. Noted that each column corresponds to several predictors being removed including the combination of previous column as well. These combinations match with Figure 4.6.3 and Figure 4.6.4. For example, the second column refers to the combination in Figure 4.6.3b, and the last column refers to the combination in Figure 4.6.4d. The indicators show that the more predictors removed, the better the performance as it gives better values of all indicators. Thus, removing 8 predictors gives the best performance as it has the lowest RMSE of 0.0041, highest R^2 of 0.83, lowest MARE of 0.26% a lowest MaxARE of 2.54%. MARE is reduced by more than half compared to the case of using 23 predictors.

Thus, the best predictor combination to be removed is CO_2, SO_2, H_2O, HCl, CO, O_2wet, P_{in}, and NO_X. Removing this combination improves the model in such a way that all data points could get predicted (Figure 4.6.5). Figure 4.6.5 also shows that the unusual horizontal trend is reduced, however, it does not disappear, which is the result from bad prediction during late March and late April. Scatter plot shows that test set fits quite well with the training set.

Table 4.6.1: Statistical indicators for SVM (test set in scenario2, case 50-50)

Statistical Indicator	**Removing predictors**							
	None	**CO_2 SO_2**	**H_2O**	**HCl**	**CO**	**O_2wet**	**P_{in}**	**NO_X**
RMSE	0.0079	0.0078	0.0064	0.0057	0.0050	0.0047	0.0045	0.0041
R^2	0.2741	0.2960	0.5675	0.6648	0.7296	0.7674	0.7954	0.8300
MARE	0.0062	0.0061	0.0047	0.0040	0.0033	0.0030	0.0028	0.0026
MaxARE	0.0326	0.0326	0.0326	0.0293	0.0293	0.0290	0.0262	0.0254

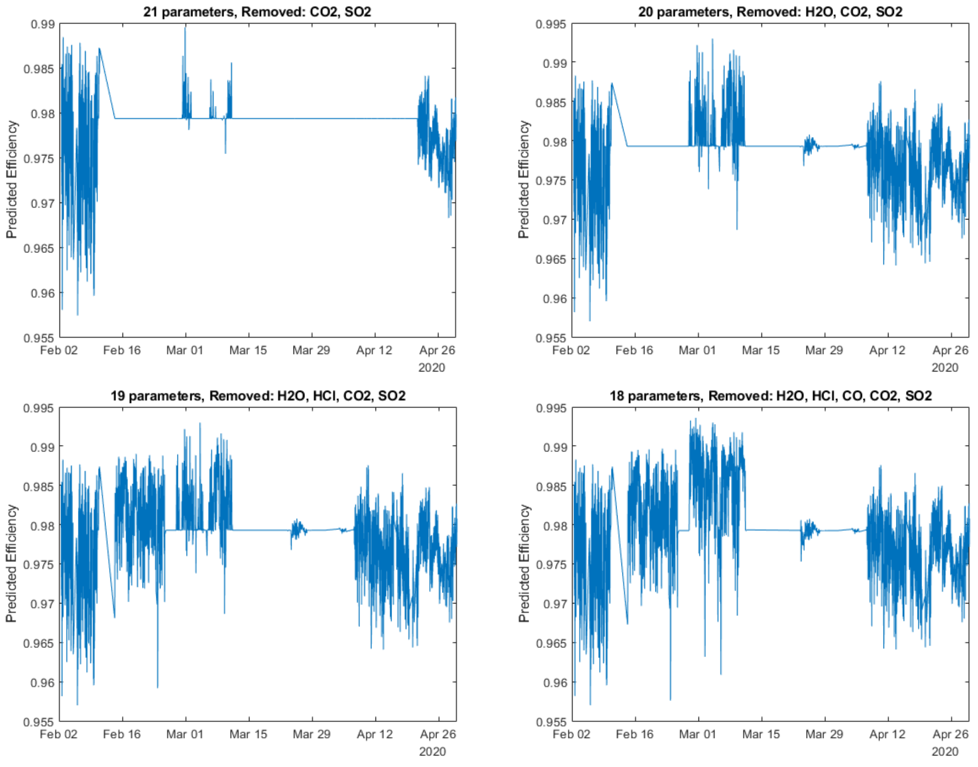


Figure 4.6.3: Predicted efficiency on SVM (s2 case 50-50) when removing: a) CO_2 and SO_2, b) H_2O, CO_2 and SO_2, c) H_2O, HCl, CO_2 and SO_2, d) H_2O, HCl, CO, CO_2,and SO_2

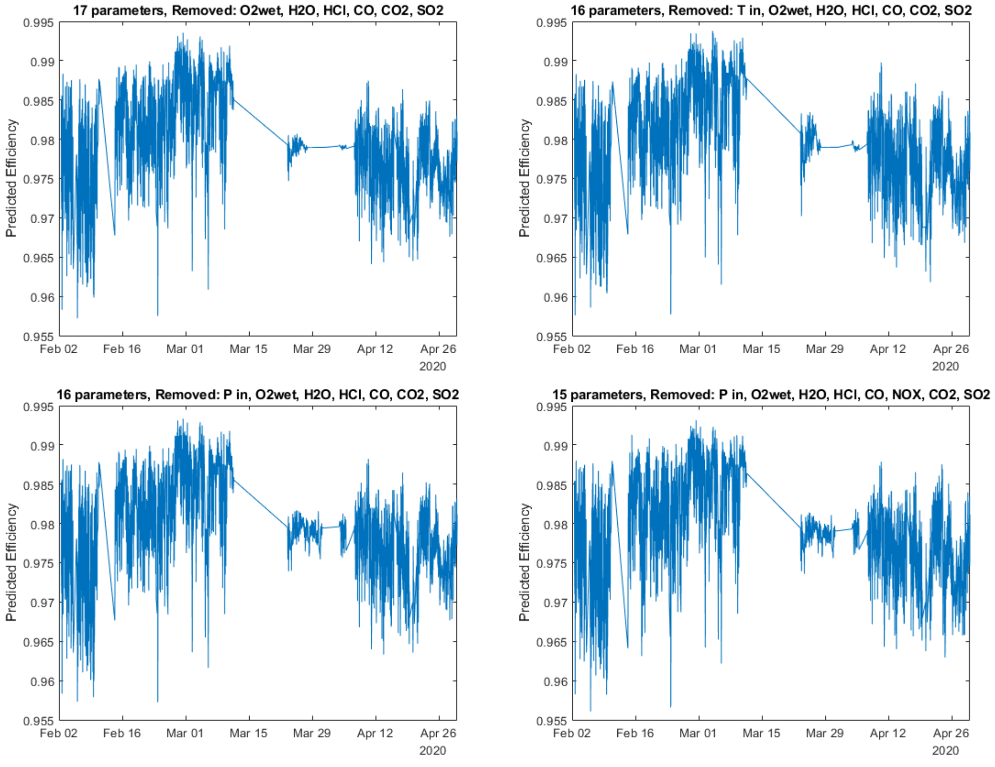


Figure 4.6.4: Predicted efficiency on SVM (s2 case 50-50) when removing: a) O_2wet, H_2O, HCl, CO, CO_2, and SO_2, b) T_{in}, O_2wet, H_2O, HCl, CO, CO_2,and SO_2, c) P_{in}, O_2wet, H_2O, HCl, CO, CO_2,and SO_2, d) P_{in}, O_2wet, H_2O, HCl, CO, NO_X, CO_2, and SO_2

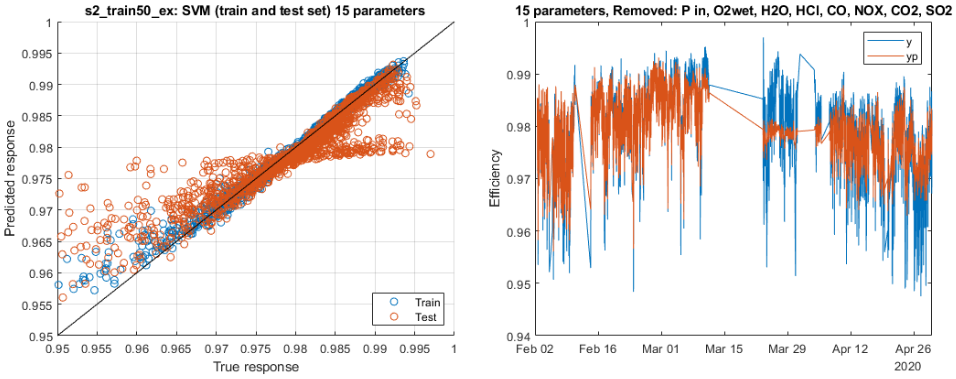


Figure 4.6.5: Predicted efficiency using scenario 2, SVM and 15 predictors

Moreover, the best predictor combination found in the sensitivity analysis is applied to other models to see if they provide similar improvement. Since it is found that splitting case 65-35 of scenario 2 gives the best result so far, this predictor combination will be removed from this case. Table 4.6.2 shows statistical indicators from test set of all 6 models with 15 predictors remained. Figure 4.6.6 shows the model performance in scatter plots comparing training and test sets as well as plots of efficiency with time comparing true and predicted response of the test set.

By removing 8 predictors (CO_2, SO_2, H_2O, HCl, CO, O_2wet, P_{in}, and NO_X), the prediction trends are improved for all models (i.e., comparing Figure 4.4.1 with Figure 4.6.6). The linear regression, PCR, and PLSR models have better alignment of training and test sets and their efficiency prediction has less error and closer to maximum limit. However, the comparison of the statistics in Table 4.6.2 and Table 4.6.3 shows only marginal improvement for these models. On the contrary, the prediction trends and statistics of SVM-based models are significantly improved, especially SVM and SVM with PCA. It clearly shows that the horizontal line is reduced in SVM and disappeared in SVM with PCA while the trend and statistics are the same for SVM with PLS.

Next, statistical indicators for all models with 15 and 23 predictors are compared in Table 4.6.2 and Table 4.6.3. When all 23 predictors are used, SVM with PLS provides the best performance. All models improve when using 15 predictors. Table 4.6.2 shows that SVM with PCA provides the best performance with the lowest RMSE, highest R^2 of 0.9, and lowest MARE of 0.24% whereas SVM with PLS gives the lowest MaxARE of 1.46%. Further investigation on other splitting cases shows that SVM with PCA has the best performance for all splitting cases with 15 predictors (Appendix D and E). More specifically, the SVM with PCA of case 50-50 provides the best indicator values with the lowest RMSE of 0.0029, highest R^2 of 0.9161, and lowest MARE of 0.19% whereas the lowest MaxARE is from SVM PLS in case 55-45 with value of 1.42%.

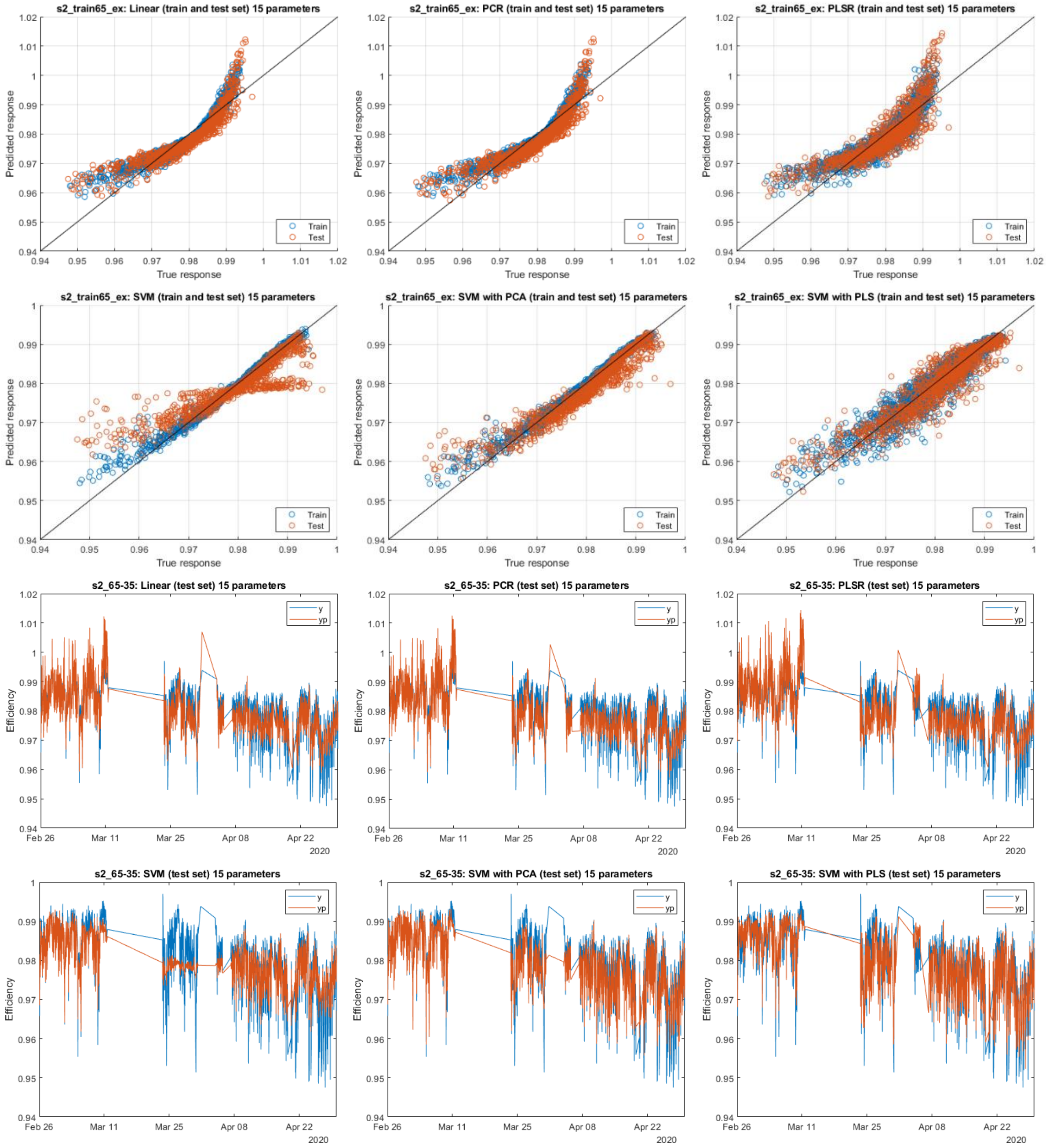


Figure 4.6.6: Performance of 6 models with scenario 2 case 65-35 and 15 predictors

Table 4.6.2: Statistical indicators for all models (test set, s2, case 65-35, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0043	0.0044	0.0052	0.0047	0.0032	0.0035
R^2	0.7873	0.7860	0.6996	0.7997	0.9055	0.8573
MARE	0.0034	0.0037	0.0041	0.0031	0.0024	0.0027
MaxARE	0.0181	0.0175	0.0193	0.0252	0.0173	0.0146

Table 4.6.3: Statistical indicators for all models (test set, s2, case 65-35, 23 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0047	0.0045	0.0053	0.0077	0.0047	0.0035
R^2	0.7536	0.7755	0.6977	0.5348	0.8009	0.8594
MARE	0.0038	0.0038	0.0041	0.0062	0.0033	0.0026
MaxARE	0.0178	0.0172	0.0200	0.0287	0.0247	0.0145

5 Conclusion

ESP process data obtained from 30 November 2019 until 30 April 2020 with one-hour resolution has 23 predictors which are ash concentrations (Ash_{in} and Ash_{out}), steam productions (Steam and RealSteam), voltages (V_1,V_2 and V_3), currents (I_1, I_2 and I_3), volumetric flowrate (Q), temperature (T_{in} and T_{out}), pressure (P_{in} and P_{out}), oxygen and water content (O_2wet, O_2dry, and H_2O), and exhaust gas compositions (HCl, CO, NO_X, CO_2 and SO_2). It should be noted that these 23 predictors can be groupped as i) flue gas input parameters (Q, T_{in}, and P_{in}) that can be affected by the process system before the ESP, ii) ESP process parameters (V_1, V_2, V_3, I_1, I_2, and I_3) that can be controlled for the ESP operation, and iii) ESP output parameters (T_{out}, P_{out}, O_2wet, O_2dry, H_2O, HCl, CO, NOx, CO_2 and SO_2) that can only be observed. However, out of this list of parameters, the most problematic one is outlet ash concentration (Ash_{out}) as it is similar to the model output (i.e., it cannot be controlled to improved the ESP operation). The rest of the variables can be in some extent controlled or known before the ESP operation so that the ESP controller can use them as inputs to the model in order to see what would be expected. The obtained data is preprocessed by removing missing data, and known errors. Then, the data is cleaned by removing outliers using different methods and results in 3 different scenarios which are s1: no outliers removed, s2: outliers removed by standard deviation method, and s3: outliers removed by Mahalanobis distance. Data in each scenario is split into training and test sets for 7 cases having different percentage of training and test set. The splitting cases are 50-50, 55-45, 60-40, 65-35, 70-30, 75-25, and 80-20. The main models are linear regression and SVM. Each of them is additionally applied with PCA and PLS for dimensionality reduction. Thus, there are 6 models in total.

Firstly, correlation between parameters is calculated and it is found that efficiency correlates with ash concentrations and V_1, V_2, V_3, I_1, and I_2 the most. Effect of outlier removal methods is also investigated. It shows that scenario 2 with outliers removed by standard deviation method gives the best performance in most cases, although the standard deviation method discards a lot of data points. For the prediction trend, linear regression fails to predict efficiency at very low and very high efficiency. The big disadvantage of using linear regression base models is that it predicts the response beyond the maximum possible limit of 100%. The SVM based models provide better performance as data points align very well with the diagonal line, in the case of interpolation performance. However, SVM and SVM with PCA give unusual horizontal line when predicting future efficiency values. With all 23 predictors, SVM with PLS give the best prediction trend among 6 models. All 7 splitting cases are compared. The results show that case 65-35 provides the best performance with R^2 of 0.86, MARE of 0.26% and MaxARE of 1.45%.

Finally, sensitivity analysis is performed to see how each predictor affects the efficiency. In this context, feature selection is performed to improve the models by removing several predictor combinations. It is found that the best predictor combination to be removed is CO_2, SO_2, H_2O, HCl, CO, O_2wet, P_{in}, and NO_X. Thus, there are 15 predictors left on each model. Unusual trend of SVM and SVM with PCA from using all predictors is significantly reduced and generally all models are improved in some extent when the aforementioned combination of predictors is removed. Moreover, SVM with PCA model gives best performance for all splitting cases with 15 predictors. More specifically, SVM with PCA of case 50-50 provides the best indicator values with the lowest RMSE of 0.0029, highest R^2 of 0.9161, and lowest MARE of 0.19% whereas the lowest MaxARE of 1.42% is from SVM with PLS in case 55-45.

In conclusion, SVM with PCA model with 15 predictors is recommended for ESP efficiency prediction. To get the closest prediction, outliers should be removed by standard deviation method and the data should be split by half for training and test set (case 50-50). Cross validation should be used when training the model to prevent overfitting. This model prediction is off by the average of around 0.2% with maximum different up to 1.42%. Moreover, these models can be used in optimization scenarios, where the ESP controller finds the optimal predictor values of ESP operation for various scenarios of flue gas properties. Again, in this way, the most problematic predictor is outlet ash concentration (Ash_{out}) as neither the ESP operator nor the process engineers that control the process prior to the ESP can affect this predictor (whereas Ash_{in} can be in some extent affected by the type of waste burned etc., but even if it cannot be affected it is an important input parameter that it makes sense to be used for any kind of predictive model of the ESP).

6 References

[1] Swaminathan M R, *Flow and Turbulence Studies in Electrostatic Precipitator,* Chennai: Faculty of Mechanical Engineering, Anna University, 2010.

[2] S. M. E Haque, M. G. Rasul, M. M. K Khan, "Fine particulate emission control by optimizing process parameters of an electrostatic precipitator," in *WSEAS International Conference. Proceedings. Mechanical Engineering Series. No. 5. World Scientific and Engineering Academy and Society*, 2010.

[3] Ezzat Jaroudi, Ivan Sretenovic, Greg Evans, Honghi Tran, "Factors affecting particulate removal efficiency of kraft recovery boiler electrostatic precipitators: a technical review," *TAPPI Journal,* vol. 17, no. 5, pp. 273-283, 2018.

[4] K J McLean, "Electrostatic precipitators," *IEE Proceedings,* vol. 135, no. 6, pp. 347-361, 1988.

[5] Thi-Cuc Le, Guan-Yu Lin, Chuen-Jinn Tsai, "The Predictive Method for the Submicron and Nano-Sized Particle Collection Efficiency of Multipoint-to-Plane Electrostatic Precipitator," *Aerosol and Air Quality Research,* vol. 13, pp. 1404-1410, 2013.

[6] A. Mizuno, "Electrostatic Precipitation," *IEEE Transactions on Dielectrics and Electrical Insulation,* vol. 7, no. 5, pp. 615-624, 2000.

[7] R. B. Manuzon, *Electrostatic Precipitation Technologies for the Mitigation of Particulate Matter Emissions from Poultry Facilities,* United States: The Ohio State University, 2012.

[8] J. Katz, "The Electrostatic Precipitatr: Application and Concepts," in *Handbook of Powder Science and Technology,* New York, 1997, pp. 753-770.

[9] Thiago Batista Soeiro, *High Efficiency Electrostatic Precipitator Systems with Low Effects on the Mains,* Zurich, Switzerland: ETH Zurich, 2012.

[10] "Air Quality Control Systems: Electrostatic Precipitators (ESP)," Mitsubishi Power, October 2020. [Online]. Available: https://power.mhi.com/products/aqcs/lineup/dust-collector/.

[11] Usama K., Yasin K., "Performance Evaluation of Two Stages Electrostatic Precipitator Novel Design Under Loading Conditions," in *IEEE,* 2016.

[12] Usama K., Abderrahmane B., Falah A., Yasin K., Abdulrehman A-A., "Experimental and Analytical Study for Performance of Novel Design of Efficient Two-Stage Electrostatic Precipitator," *IET Science, Measurement & Technology, The Institution of Engineering and Technology,* 2018.

[13] Sabah O. H. Al-Shujairi a, "Comparing Electrostatic Precipitator Performance of Two-Stage with Single-Stage to Remove Dust from Air Stream," *International Journal of Scientific & Engineering Research,* vol. 4, no. 2, 2013.

[14] Bao-Yu Guo, Ai-Bing Yu, Jun Guo, "Numerical modeling of electrostatic precipitation: Effect of Gas Temperature," *Journal of Aerosol Science, Elsevier,* vol. 77, pp. 102-115, 2014.

[15] *Operation and Maintenance Manual for Electrostatic Precipitators,* United States: Air and Energy Engineering Research Laboratory, Environmental Protection Agency, 1985.

[16] MathWorks, "Machine Learning in MATLAB," [Online]. Available: https://se.mathworks.com/help/stats/machine-learning-in-matlab.html. [Accessed October 2020].

[17] MathWorks, “What Is a Linear Regression Model?,” [Online]. Available: https://se.mathworks.com/help/stats/what-is-linear-regression.html. [Accessed October 2020].

[18] Neter, J., M. H. Kutner, C. J. Nachtsheim, and W. Wasserman., Applied Linear Statistical Models, The McGraw-Hill Companies Inc., 1996.

[19] Seber, G. A. F, Linear Regression Analysis, John Wiley and Sons Inc., 1977.

[20] R. Rastogi, “Support Vector Regression and it’s Mathematical Implementation,” 5 June 2020. [Online]. Available: https://medium.com/@rahulrastogi1104/support-vector-regression-and-its-mathematical-implementation-b6377898cd74. [Accessed October 2020].

[21] “Support Vector Machine Regression,” [Online]. Available: http://kernelsvm.tripod.com/. [Accessed October 2020].

[22] “Support Vector Machine - Regression (SVR),” [Online]. Available: http://www.saedsayad.com/support_vector_machine_reg.htm. [Accessed October 2020].

[23] MathWorks, “Understanding Support Vector Machine Regression,” [Online]. Available: https://se.mathworks.com/help/stats/understanding-support-vector-machine-regression.html. [Accessed October 2020].

[24] Vapnik V., The Nature of Statistical Learning Theory, New York: Springer, 1995.

[25] H. Goonewardana, “PCA: Application in Machine Learning,” 28 February 2019. [Online]. Available: https://medium.com/apprentice-journal/pca-application-in-machine-learning-4827c07a61db. [Accessed October 2020].

[26] A. Kumar, “Principal Component Analysis with Python,” 3 October 2018. [Online]. Available: https://www.geeksforgeeks.org/principal-component-analysis-with-python/?ref=lbp. [Accessed October 2020].

[27] A. Vidhya, “PCA: A Practical Guide to Principal Component Analysis in R & Python,” 21 March 2016. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/pca-practical-guide-principal-component-analysis-python/. [Accessed October 2020].

[28] S. S. Azam, “Principal Component Analysis,” 22 April 2020. [Online]. Available: https://machinelearningmedium.com/2018/04/22/principal-component-analysis/. [Accessed October 2020].

[29] MathWorks, “Partial Least Squares Regression and Principal Components Regression,” [Online]. Available: https://se.mathworks.com/help/stats/partial-least-squares-regression-and-principal-components-regression.html. [Accessed October 2020].

[30] Anne-Laure Boulesteix, Korbinian Strimmer, “Partial least squares: a versatile tool for the analysis of high-dimensional genomic data,” *Brief Bioinform,* vol. 8, no. 1, pp. 32-44, 2007.

[31] D. F. Soylu, “Dimension Reduction Methods for Predicting Financial data,” Department of Mathematics, Uppsala University, Uppsala, Sweden, 2015.

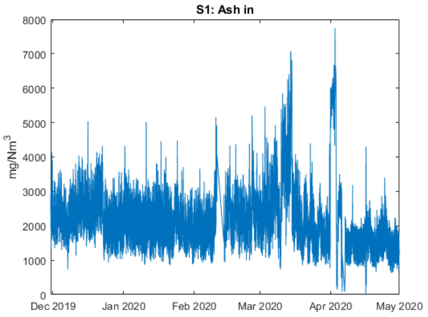
[32] Siddharth Misra, Oghenekaro Osogba, Mark Powers, “Unsupervised outlier detection techniques for well logs and geophysical data,” *Machine Learning for Subsurface Characterization,* pp. 1-37, 2020.

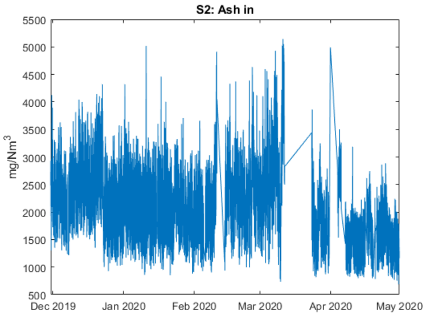
[33] H. Ghorbani, “Mahalanobis Distance and Its Application for Detecting Multivariate Outliers,” *Ser. Math. Inform.,* vol. 34, no. 3, pp. 583-595, 2019.

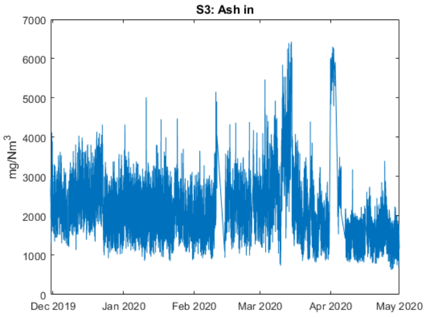
7 Appendix

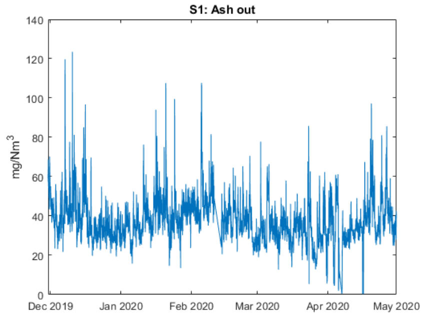
7.1 Appendix A: ESP Process Data

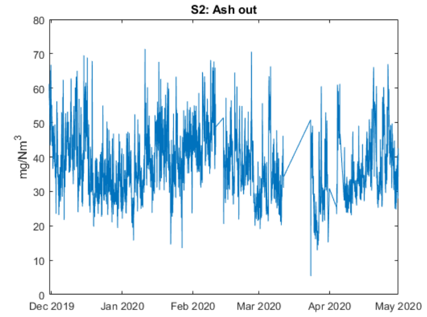
ESP process data with one-hour resolution, measured from 30 November 2019 to 30 April 2020, are cleaned and treated differently. This splits into 3 scenarios which are S1: no outliers removal, S2: outliers removed by standard deviation method and S3: outliers removed by Mahalanobis distance. Figures below clearly show that extreme outliers from the original data are removed in S2 and S3, especially in S2 that several dates during March (12th -22nd) were removed.

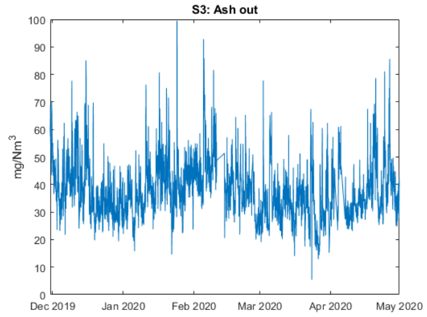


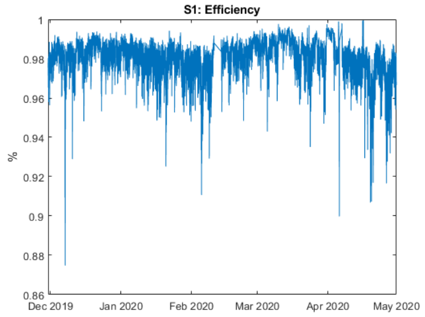


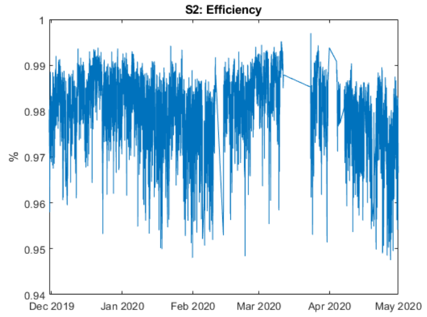


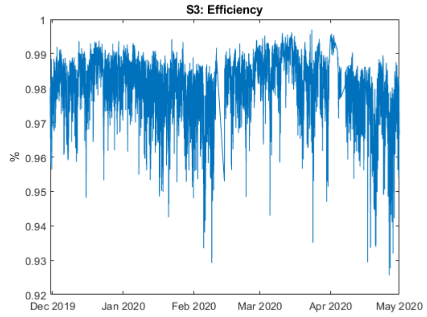


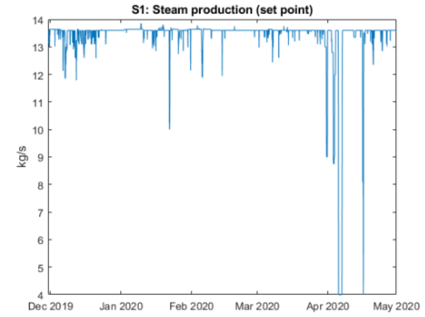


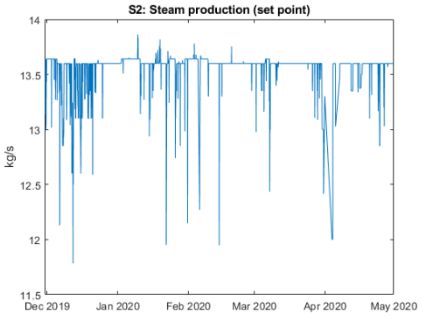


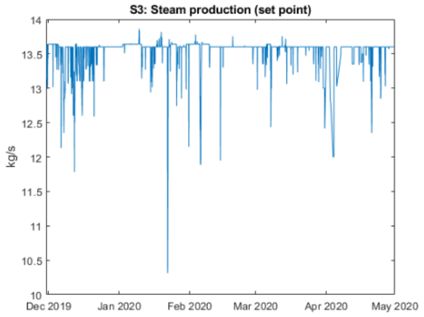


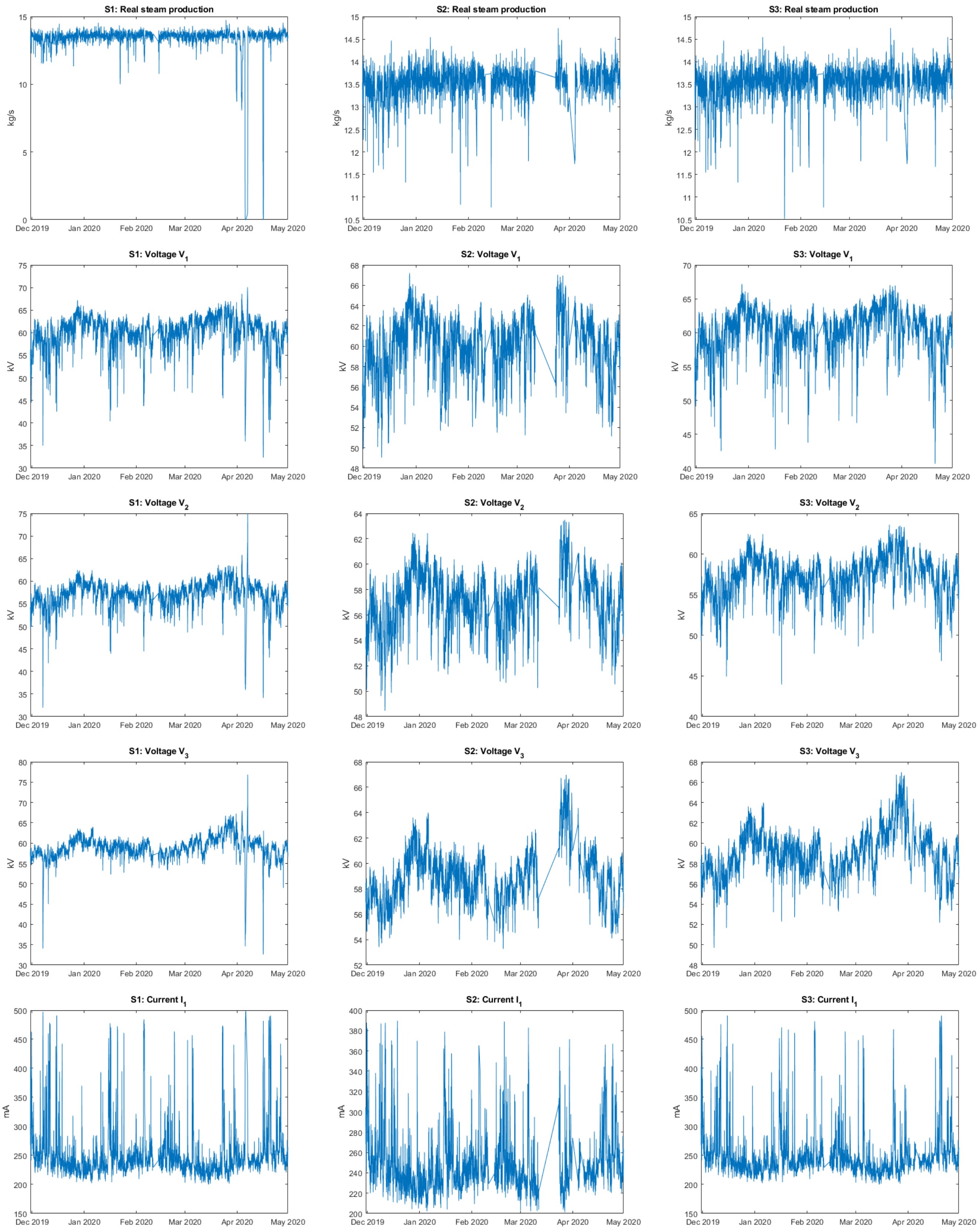


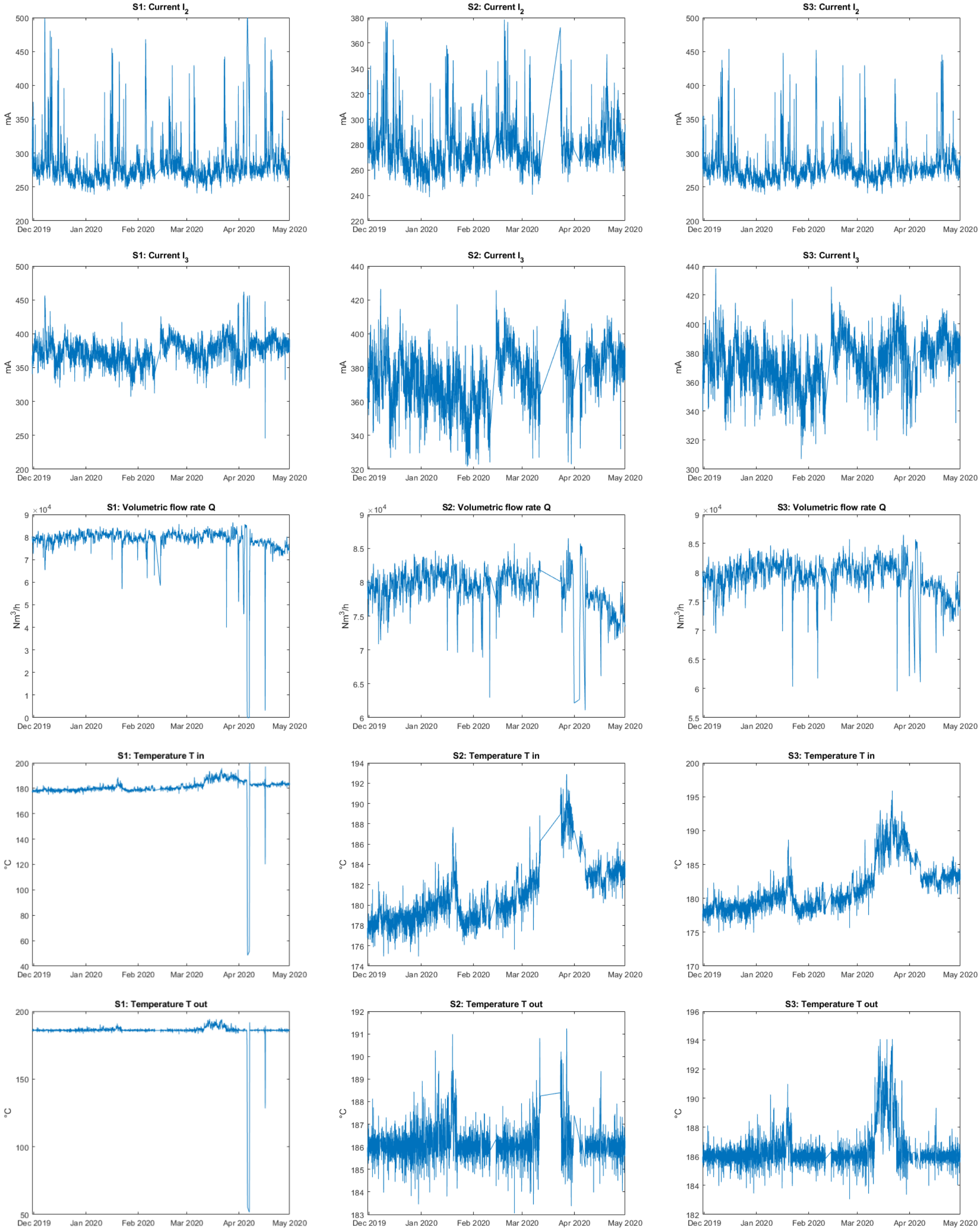


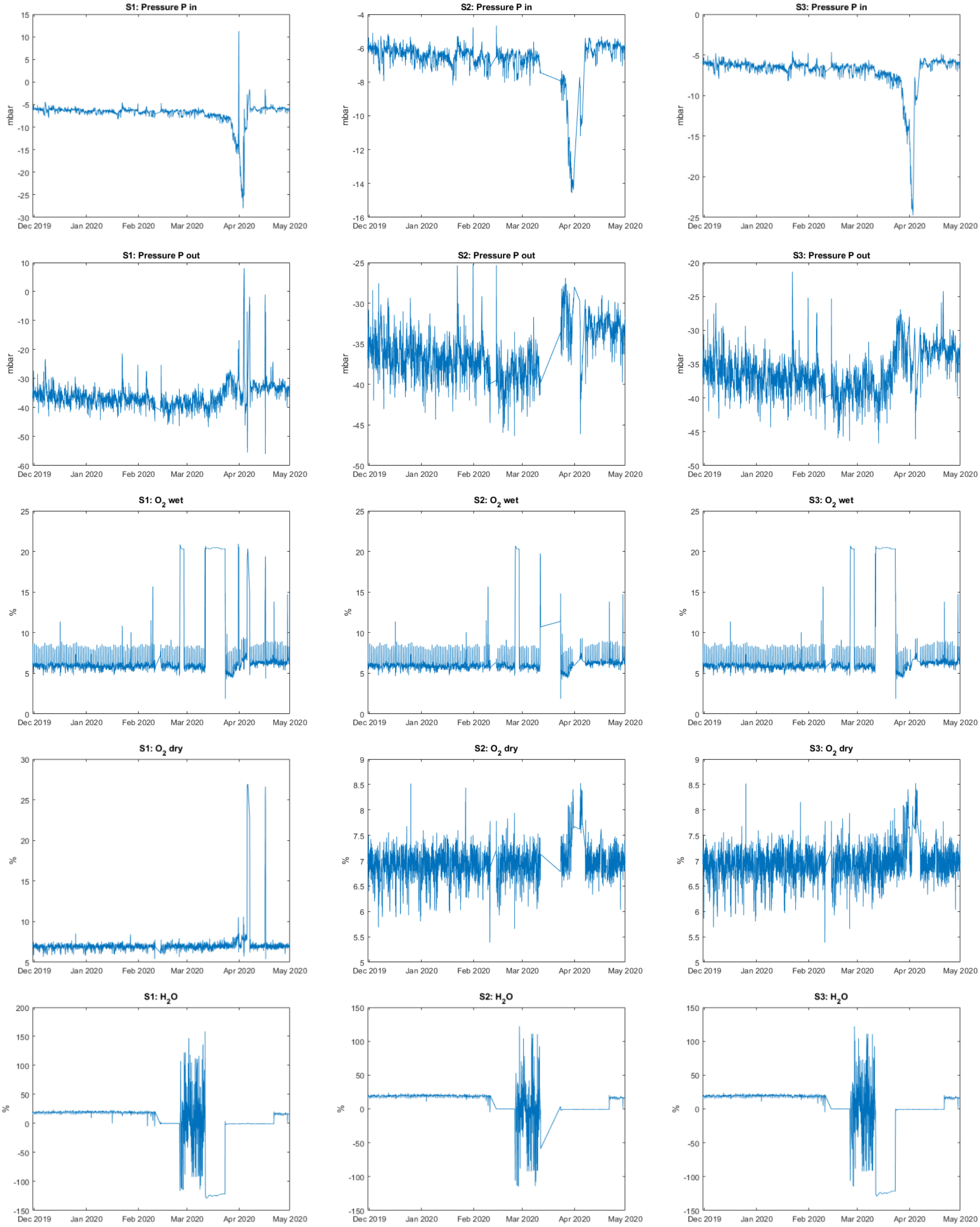







S1: Real steam production
kg/s
0
5
10
15
S2: Real steam production
kg/s
10.5
11
11.5
12
12.5
13
13.5
14
14.5
15
S3: Real steam production
kg/s
10.5
11
11.5
12
12.5
13
13.5
14
14.5
15
S1: Voltage V_1
kV
30
35
40
45
50
55
60
65
70
75
S2: Voltage V_1
kV
48
50
52
54
56
58
60
62
64
66
68
S3: Voltage V_1
kV
40
45
50
55
60
65
70
S1: Voltage V_2
kV
30
35
40
45
50
55
60
65
70
75
S2: Voltage V_2
kV
48
50
52
54
56
58
60
62
64
S3: Voltage V_2
kV
40
45
50
55
60
65
S1: Voltage V_3
kV
30
35
40
45
50
55
60
65
70
75
80
S2: Voltage V_3
kV
52
54
56
58
60
62
64
66
68
S3: Voltage V_3
kV
48
50
52
54
56
58
60
62
64
66
68
S1: Current I_1
mA
150
200
250
300
350
400
450
500
S2: Current I_1
mA
200
220
240
260
280
300
320
340
360
380
400
S3: Current I_1
mA
150
200
250
300
350
400
450
500
Dec 2019
Jan 2020
Feb 2020
Mar 2020
Apr 2020
May 2020


S1: Current I_2
mA
500
450
400
350
300
250
200
S2: Current I_2
380
360
340
320
300
280
260
240
220
S3: Current I_2
S1: Current I_3
S2: Current I_3
440
420
400
380
360
340
320
S3: Current I_3
S1: Volumetric flow rate Q
$\times 10^4$
Nm^3/h
9
8
7
6
5
4
3
2
1
0
S2: Volumetric flow rate Q
8.5
7.5
6.5
S3: Volumetric flow rate Q
5.5
S1: Temperature T in
°C
200
180
160
140
120
100
80
60
40
S2: Temperature T in
194
192
190
188
186
184
182
180
178
176
174
S3: Temperature T in
195
190
185
175
170
S1: Temperature T out
150
100
50
S2: Temperature T out
191
189
187
185
183
S3: Temperature T out
196
194
182
Dec 2019
Jan 2020
Feb 2020
Mar 2020
Apr 2020
May 2020


S1: Pressure P in
S2: Pressure P in
S3: Pressure P in
S1: Pressure P out
S2: Pressure P out
S3: Pressure P out
S1: O_2 wet
S2: O_2 wet
S3: O_2 wet
S1: O_2 dry
S2: O_2 dry
S3: O_2 dry
S1: H_2O
S2: H_2O
S3: H_2O
mbar
%
15
10
5
0
-5
-10
-15
-20
-25
-30
-4
-6
-8
-12
-14
-16
-35
-40
-45
-50
-60
25
20
9
8.5
8
7.5
7
6.5
6
5.5
200
150
100
50
-50
-100
-150
Dec 2019
Jan 2020
Feb 2020
Mar 2020
Apr 2020
May 2020

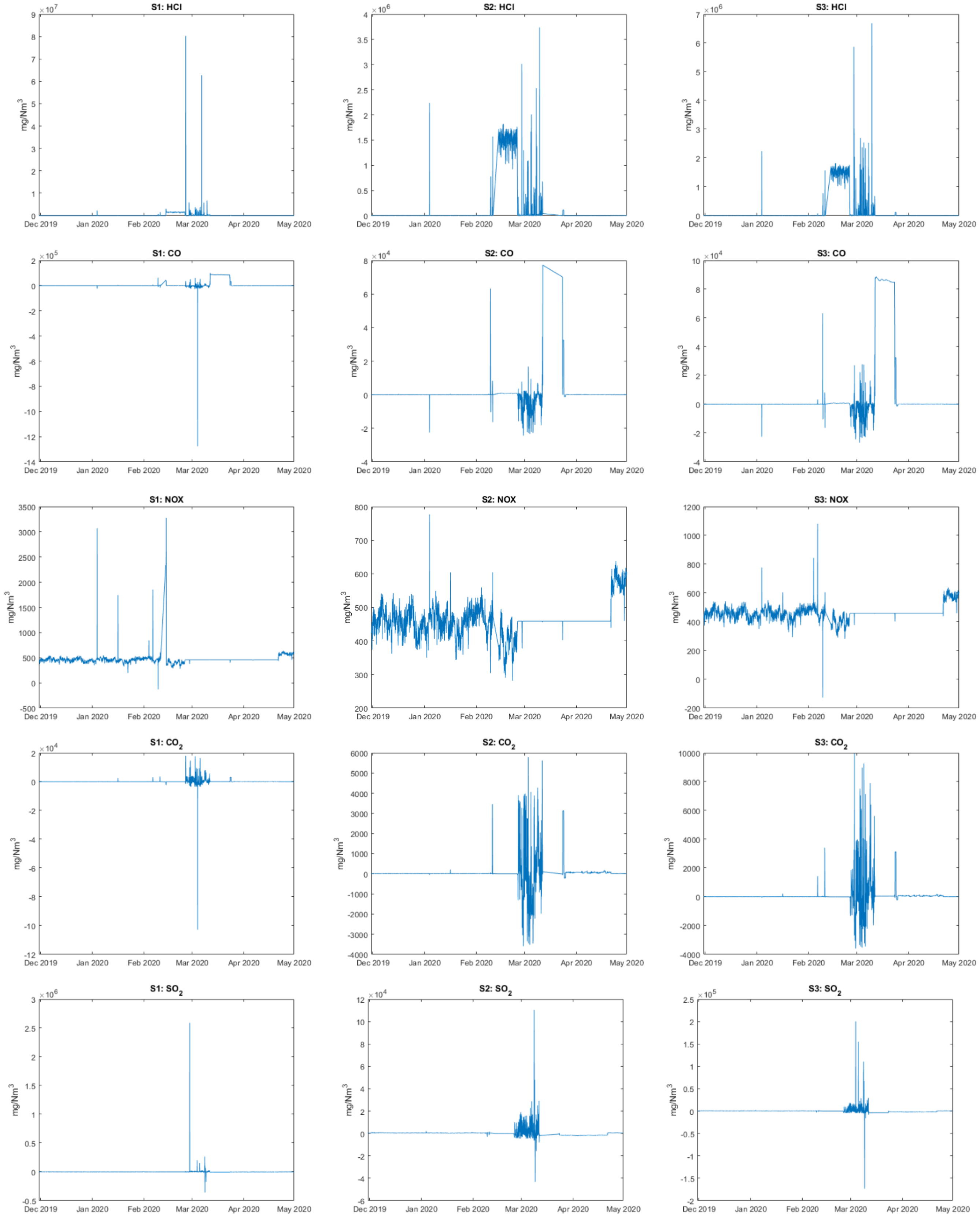


Figure 7.1.1: Comparison of ESP process data variables for each scenario

7.2 Appendix B: Extrapolating prediction performance

This section presents the extrapolating prediction results of the 6 models using all 23 predictors. The results from all 7 splitting cases are shown and compared.

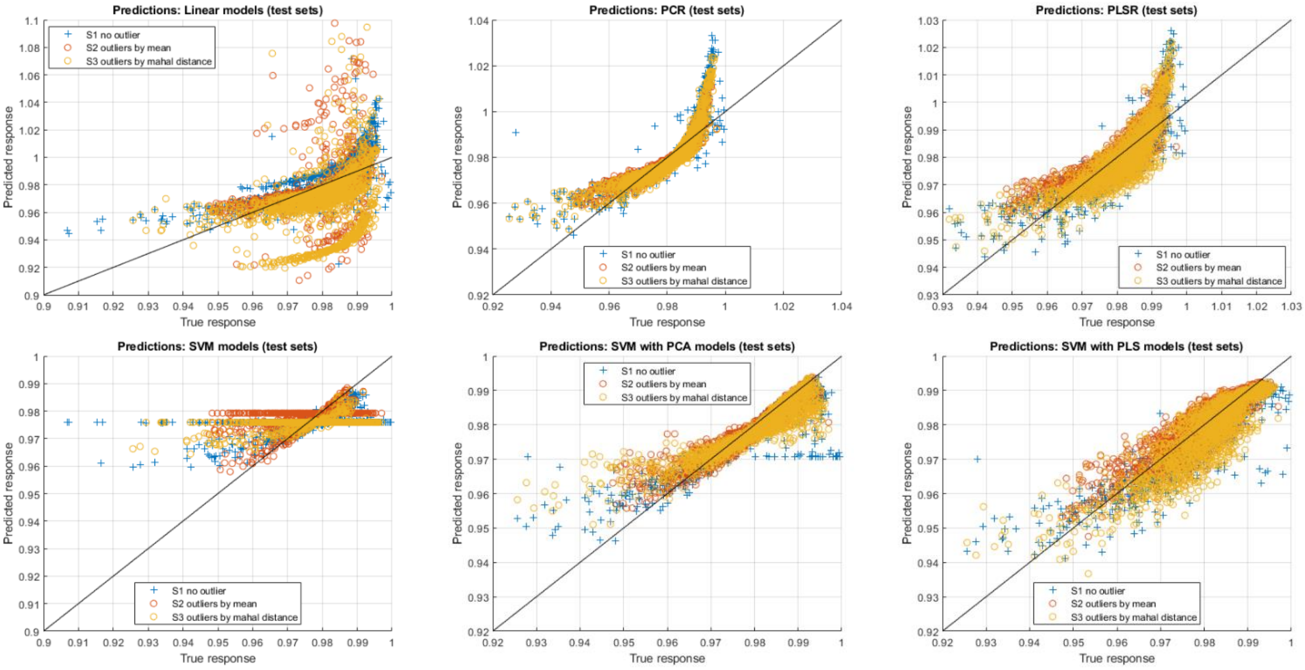


Figure 7.2.1: Case 50-50 model performance (23 predictors)

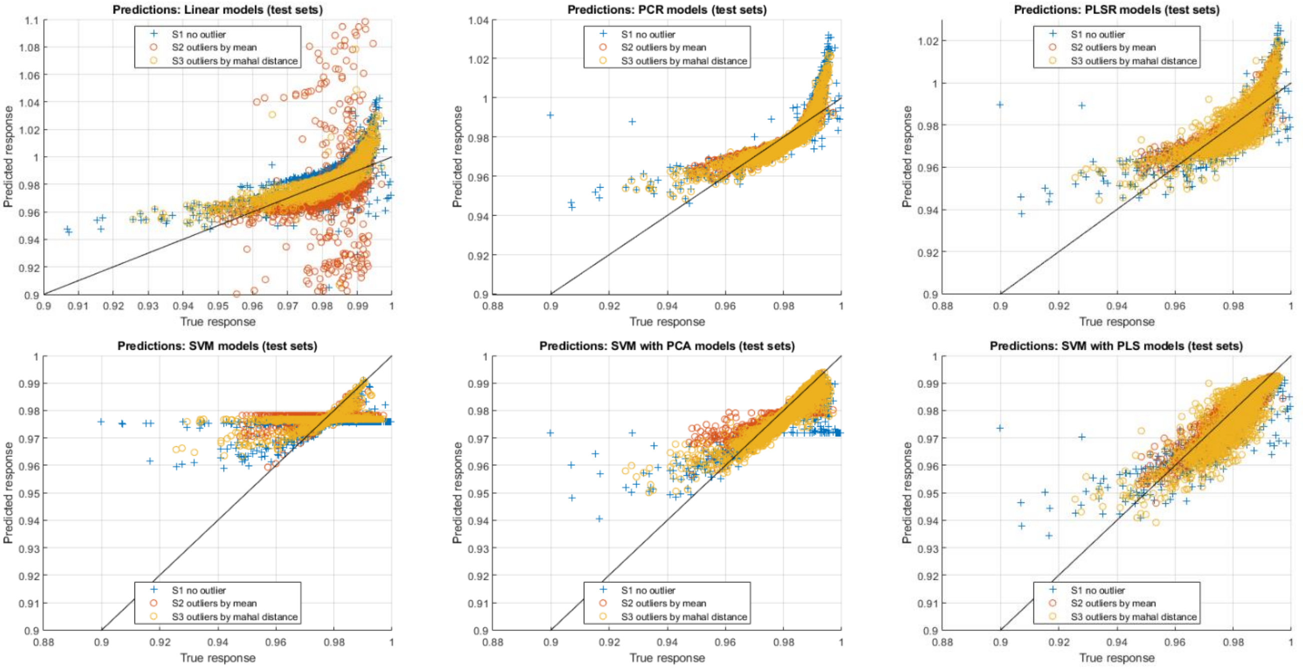


Figure 7.2.2: Case 55-45 model performance (23 predictors)

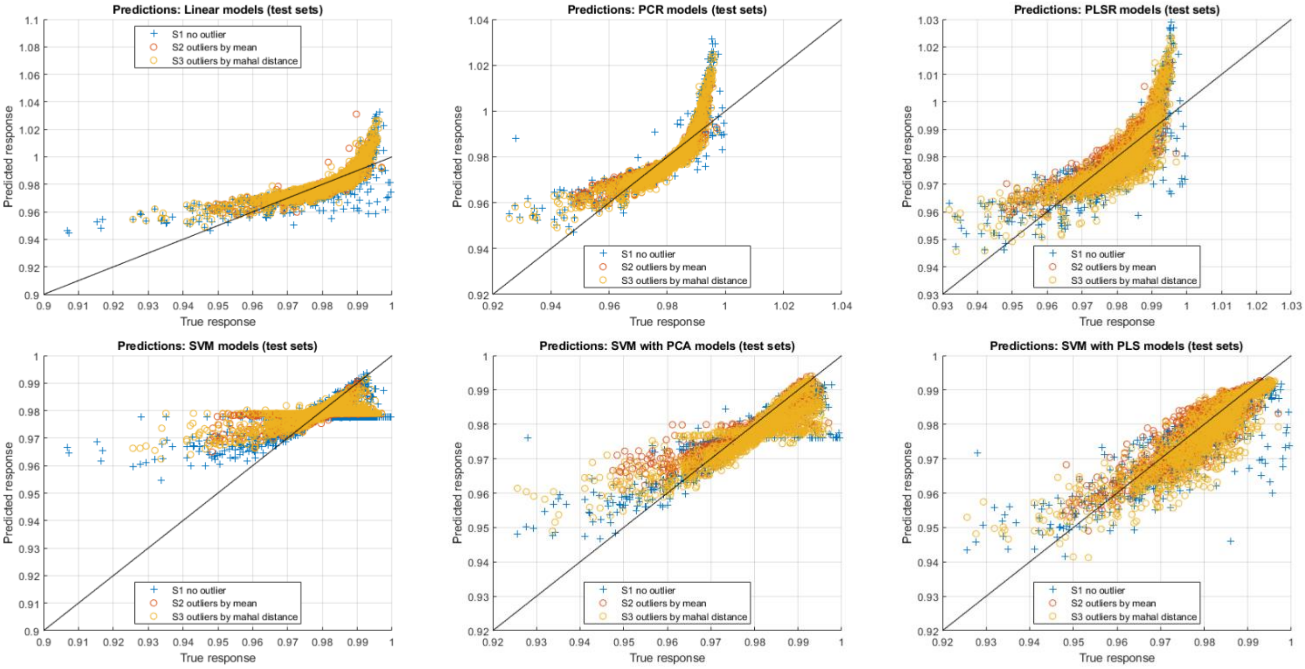


Figure 7.2.3: Case 60-40 model performance (23 predictors)

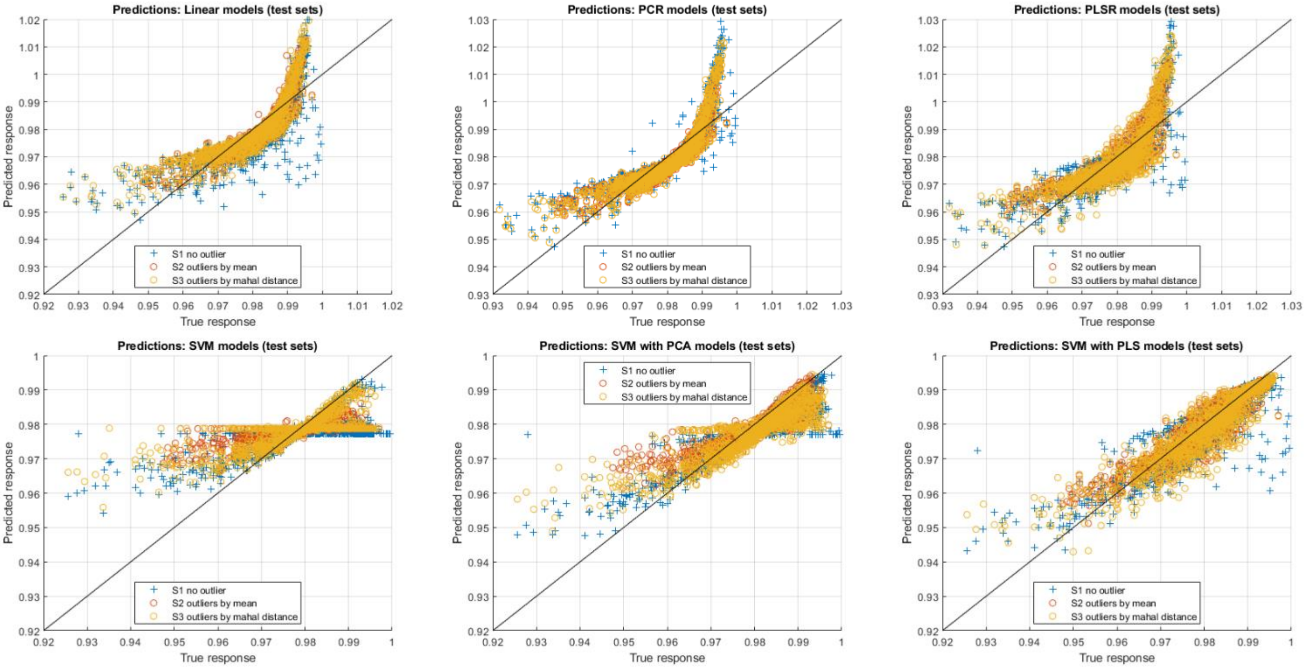


Figure 7.2.4: Case 65-35 model performance (23 predictors)

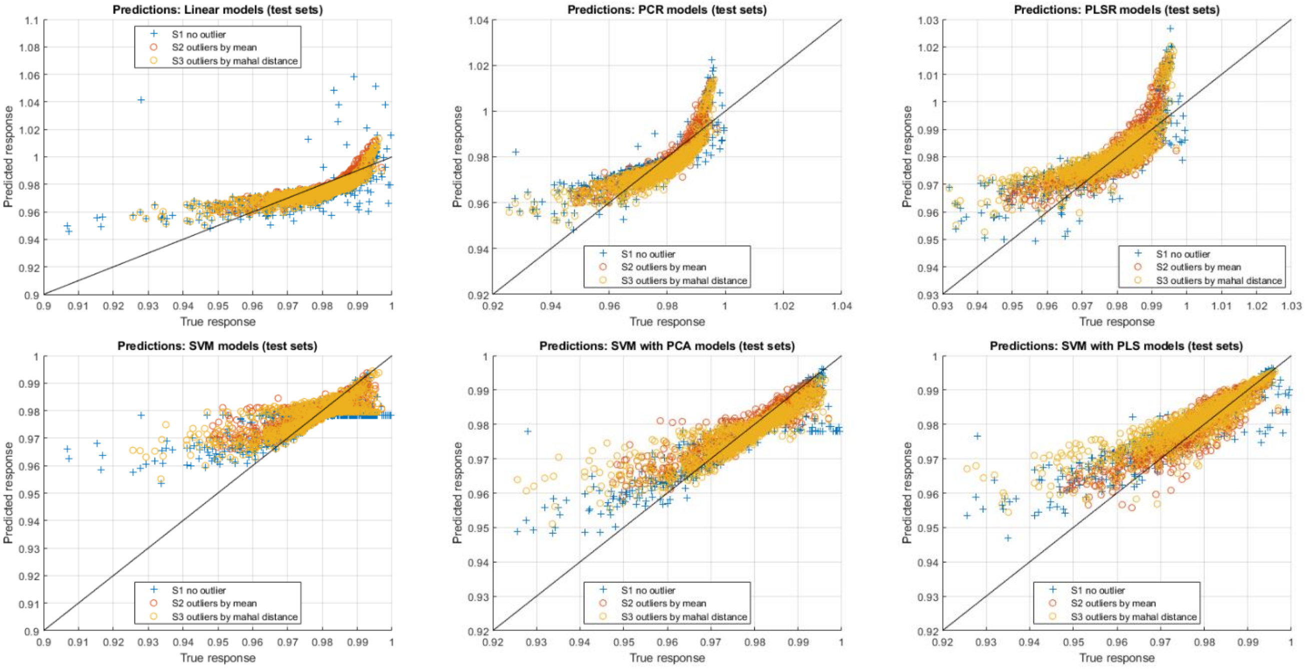


Figure 7.2.5: Case 70-30 model performance (23 predictors)

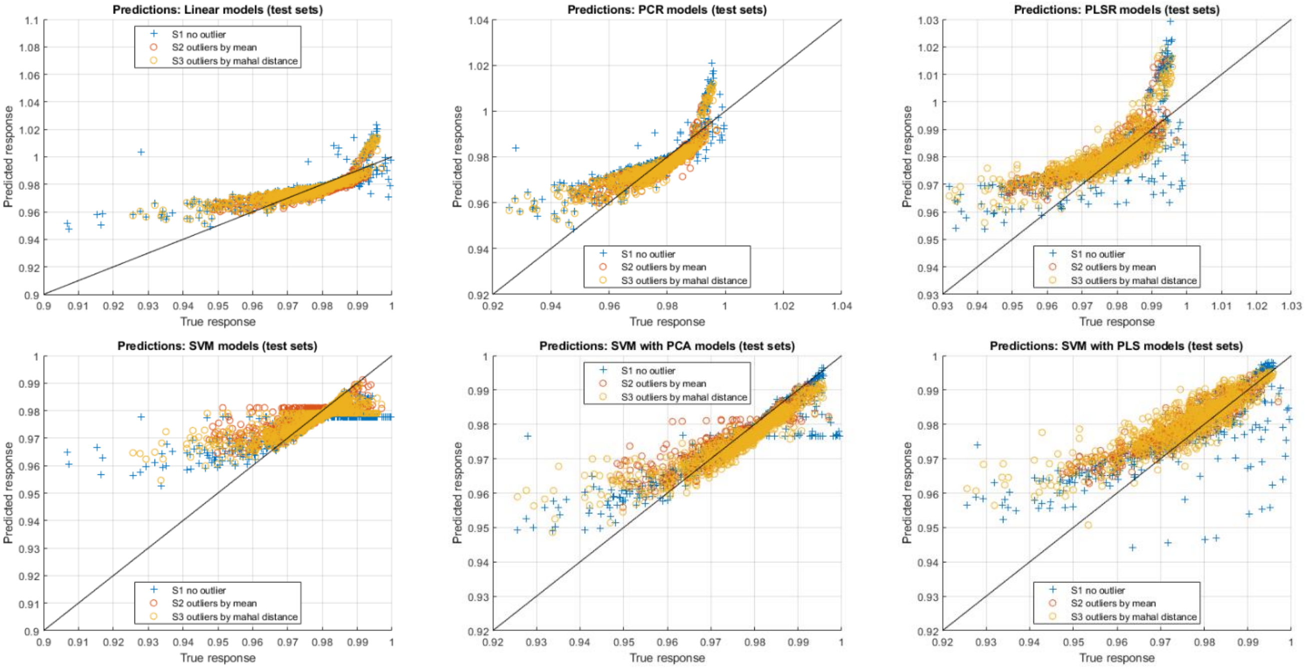


Figure 7.2.6: Case 75-25 model performance (23 predictors)

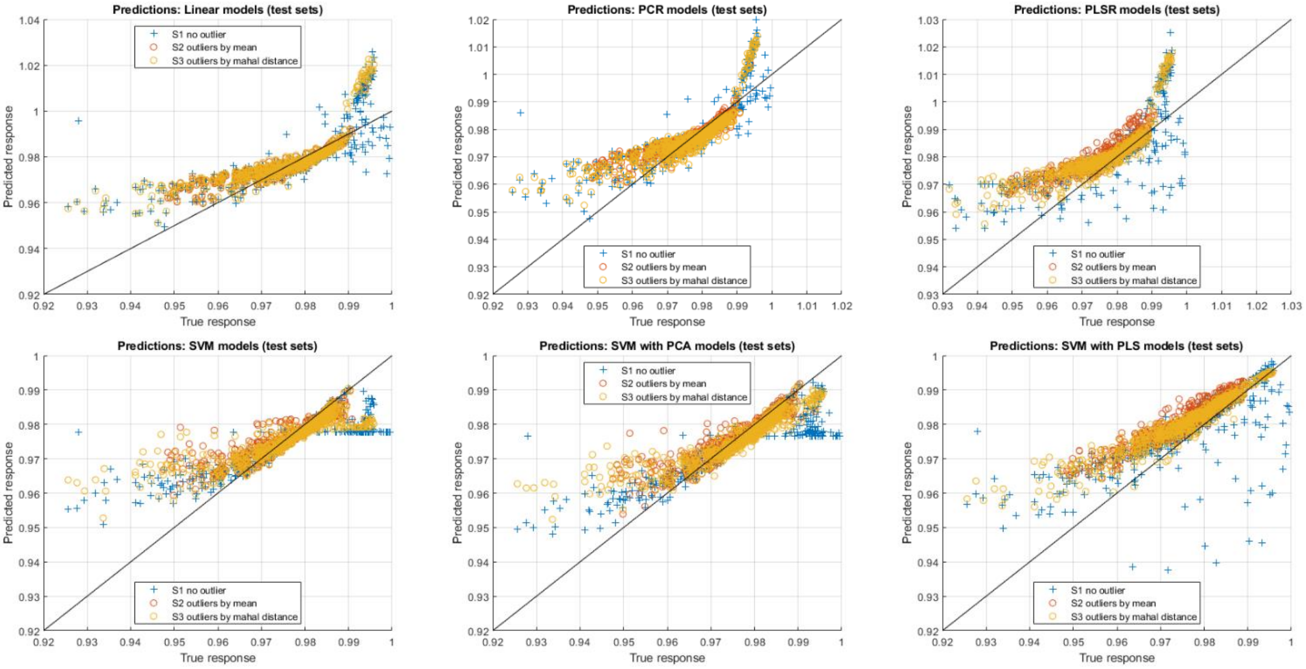


Figure 7.2.7: Case 80-20 model performance (23 predictors)

7.3 Appendix C: Statistical indicators for SVM with PLS

This section presents statistical indicators for SVM with PLS of all splitting cases with 23 predictors as shown in Table 7.3.1 to Table 7.3.7.

Table 7.3.1: Statistical indicators for SVM with PLS (Case 50-50, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0050	0.0058	0.0029	0.0041	0.0037	0.0060
R^2	0.7562	0.8251	0.8627	0.8000	0.8384	0.7643
MARE	0.0020	0.0042	0.0022	0.0032	0.0026	0.0046
MaxARE	0.0865	0.0804	0.0127	0.0169	0.0350	0.0288

Table 7.3.2: Statistical indicators for SVM with PLS (Case 55-45, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0054	0.0059	0.0027	0.0039	0.0041	0.0064
R^2	0.7126	0.8094	0.8949	0.8208	0.7968	0.7311
MARE	0.0025	0.0042	0.0020	0.0031	0.0029	0.0050
MaxARE	0.0904	0.0818	0.0116	0.0155	0.0325	0.0284

Table 7.3.3: Statistical indicators for SVM with PLS (Case 60-40, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0052	0.0065	0.0028	0.0041	0.0036	0.0057
R^2	0.7221	0.7613	0.8868	0.8074	0.8373	0.7888
MARE	0.0026	0.0045	0.0021	0.0032	0.0026	0.0043
MaxARE	0.0887	0.0836	0.0155	0.0209	0.0239	0.0302

Table 7.3.4: Statistical indicators for SVM with PLS (Case 65-35, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0049	0.0063	0.0025	0.0035	0.0034	0.0051
R^2	0.7524	0.7642	0.9023	0.8594	0.8535	0.8092
MARE	0.0024	0.0042	0.0019	0.0026	0.0025	0.0038
MaxARE	0.0841	0.0838	0.0103	0.0145	0.0237	0.0303

Table 7.3.5: Statistical indicators for SVM with PLS (Case 70-30, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0031	0.0069	0.0024	0.0039	0.0027	0.0068
R^2	0.9029	0.8457	0.9115	0.8491	0.9083	0.7864
MARE	0.0016	0.0043	0.0018	0.0029	0.0019	0.0046
MaxARE	0.0857	0.0859	0.0191	0.0167	0.0293	0.0448

Table 7.3.6: Statistical indicators for SVM with PLS (Case 75-25, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0035	0.0082	0.0025	0.0057	0.0041	0.0076
R^2	0.8660	0.6758	0.9088	0.8718	0.7951	0.7658
MARE	0.0021	0.0050	0.0019	0.0045	0.0029	0.0055
MaxARE	0.0797	0.0812	0.0144	0.0204	0.0265	0.0419

Table 7.3.7: Statistical indicators for SVM with PLS (Case 80-20, 23 predictors)

Statistical Indicators	**Data set and Scenario**					
	Train_s1	**Test_s1**	**Train_s2**	**Test_s2**	**Train_s3**	**Test_s3**
RMSE	0.0043	0.0094	0.0026	0.0069	0.0021	0.0072
R^2	0.8024	0.6054	0.8963	0.8937	0.9462	0.9058
MARE	0.0027	0.0057	0.0019	0.0059	0.0014	0.0054
MaxARE	0.0820	0.0869	0.0184	0.0221	0.0198	0.0386

7.4 Appendix D: Model performance (15 predictors)

This section presents the extrapolating prediction results of the 6 models using 15 predictors and scenario 2. The results from all 7 splitting cases are shown.

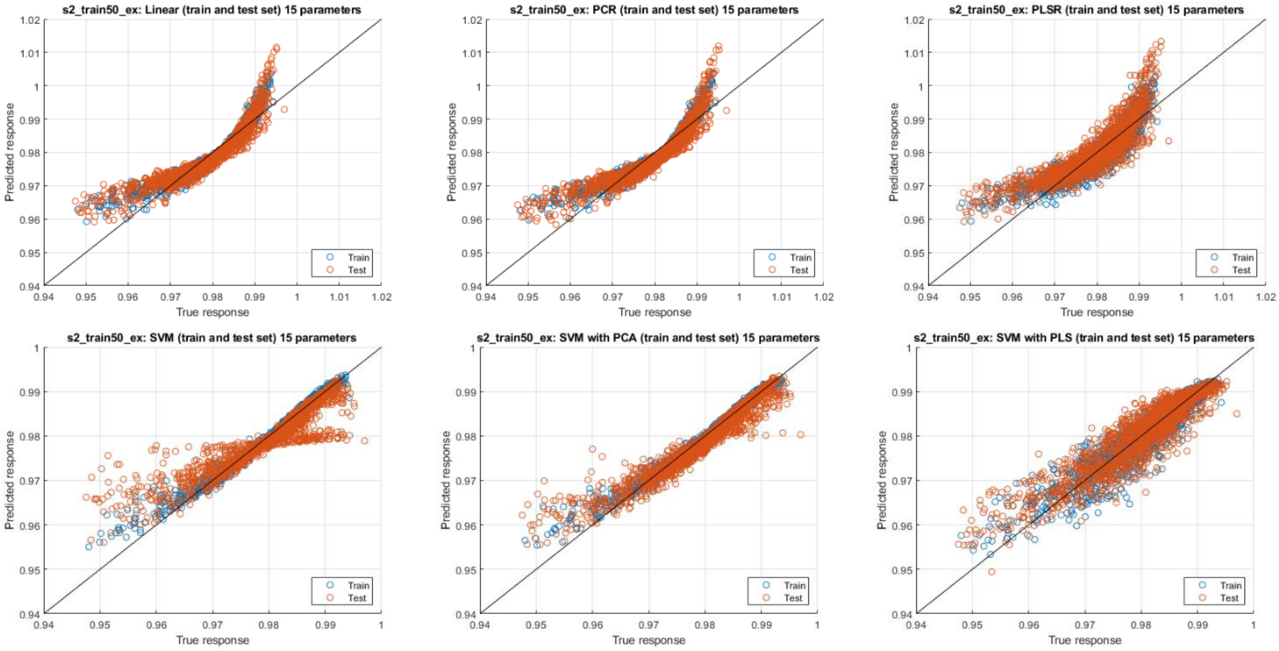


Figure 7.4.1: Case 50-50 model performance (scenario 2, 15 predictors)

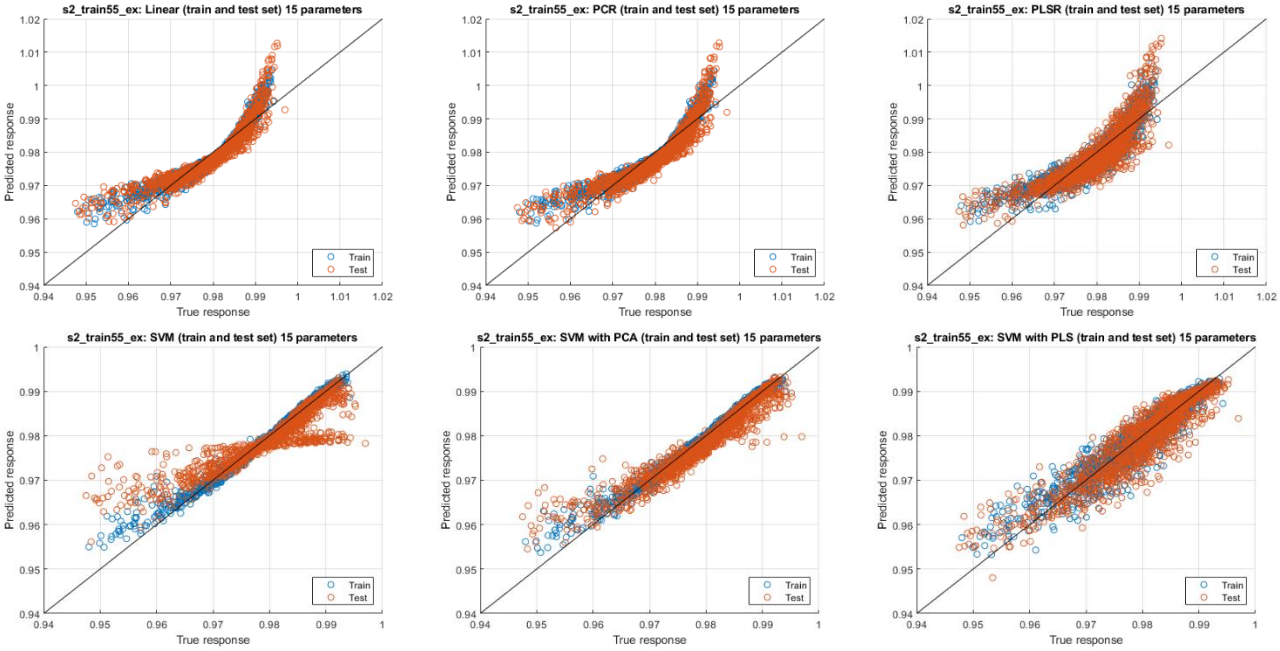


Figure 7.4.2: Case 55-45 model performance (scenario 2, 15 predictors)

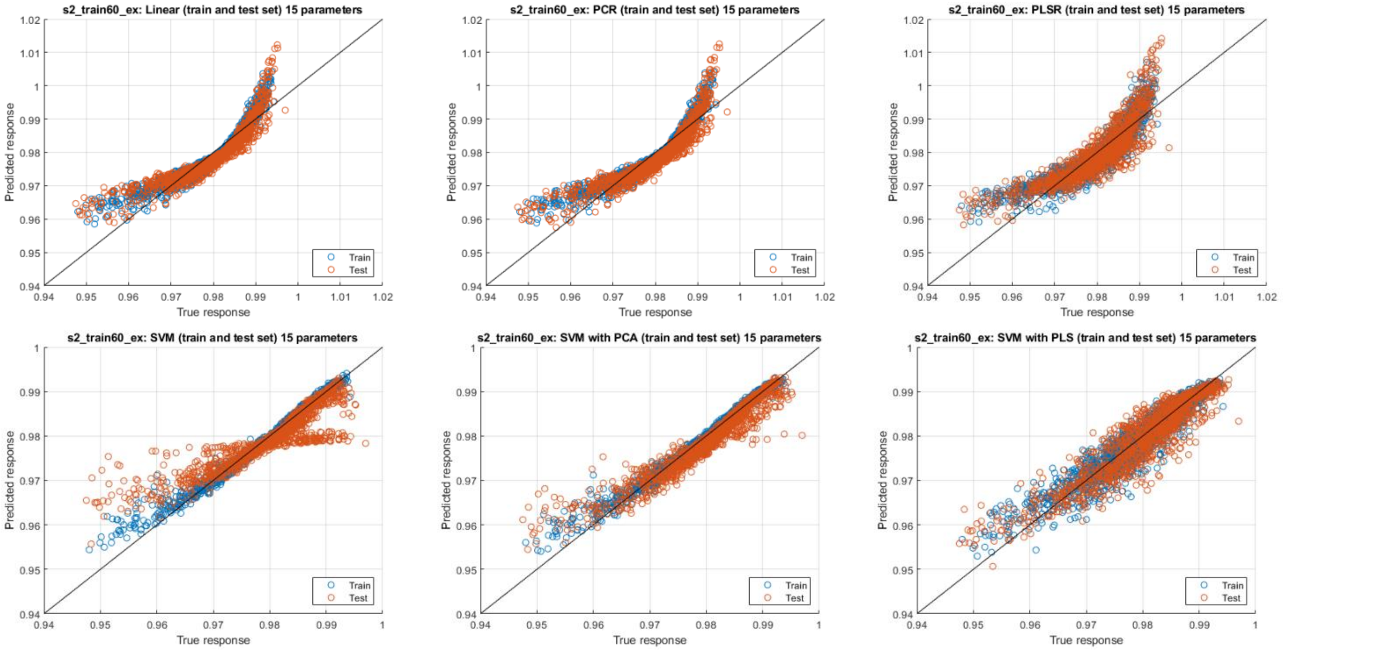


Figure 7.4.3: Case 60-40 model performance (scenario 2, 15 predictors)

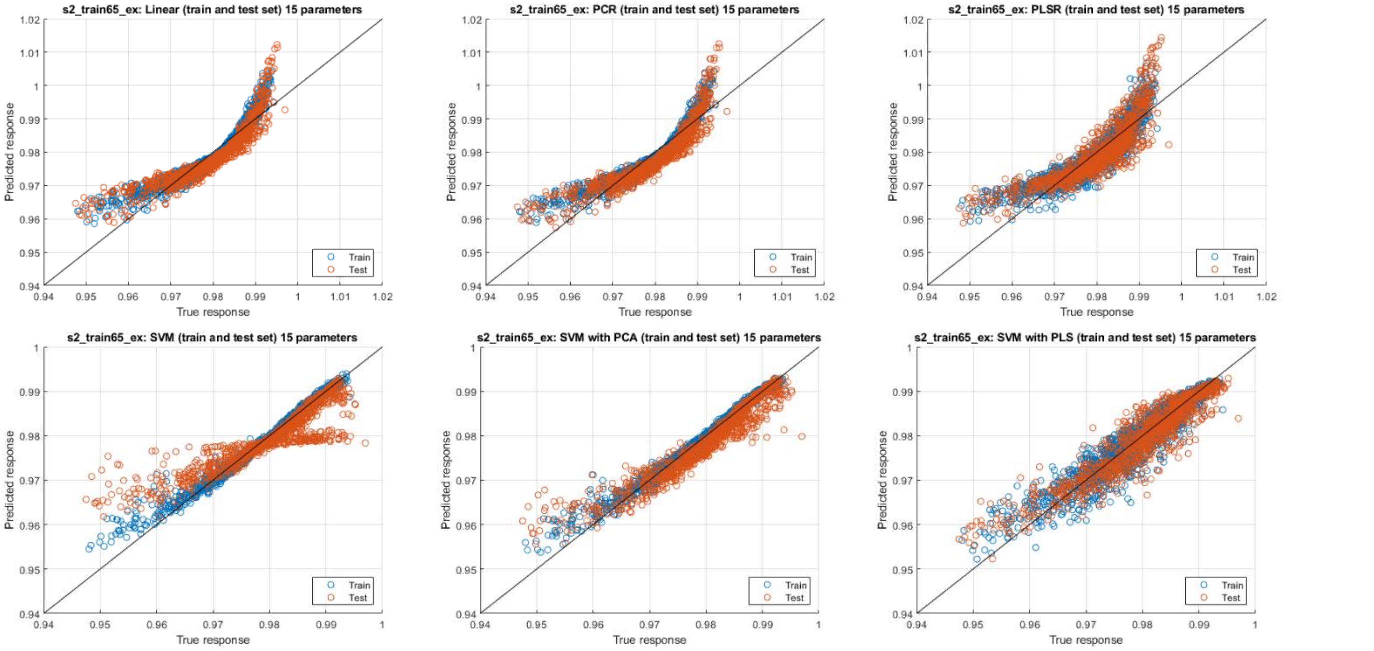


Figure 7.4.4: Case 65-35 model performance (scenario 2, 15 predictors)

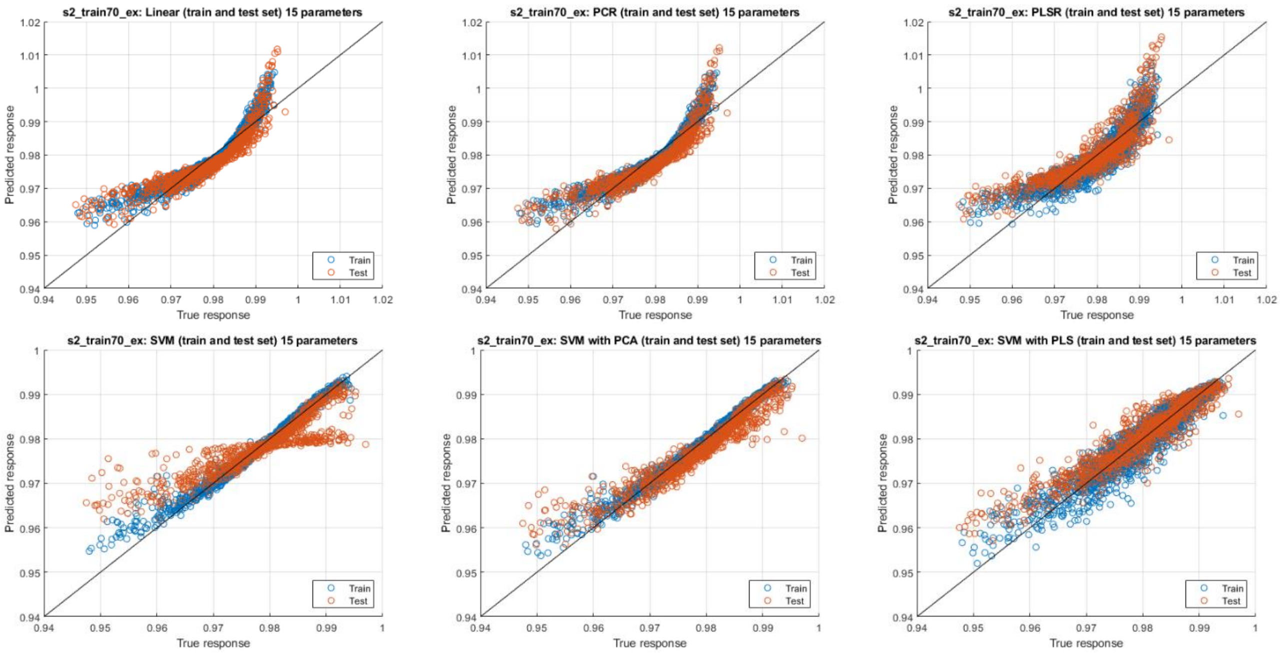


Figure 7.4.5: Case 70-30 model performance (scenario 2, 15 predictors)

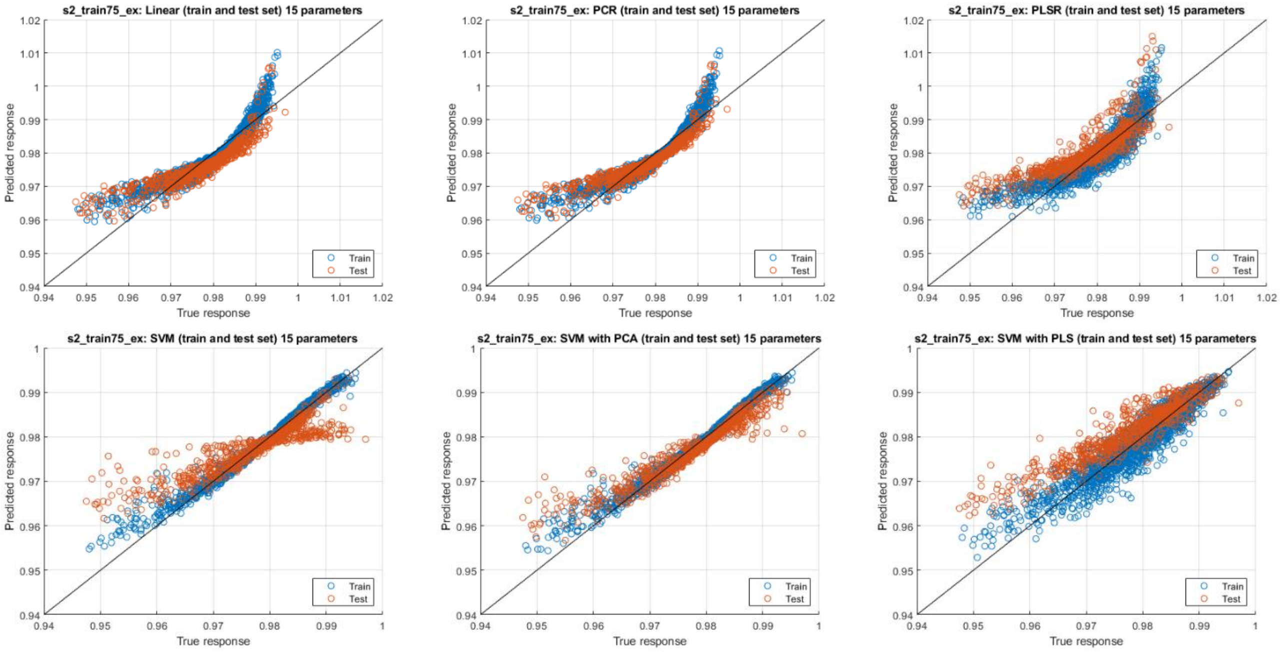


Figure 7.4.6: Case 75-25 model performance (scenario 2, 15 predictors)

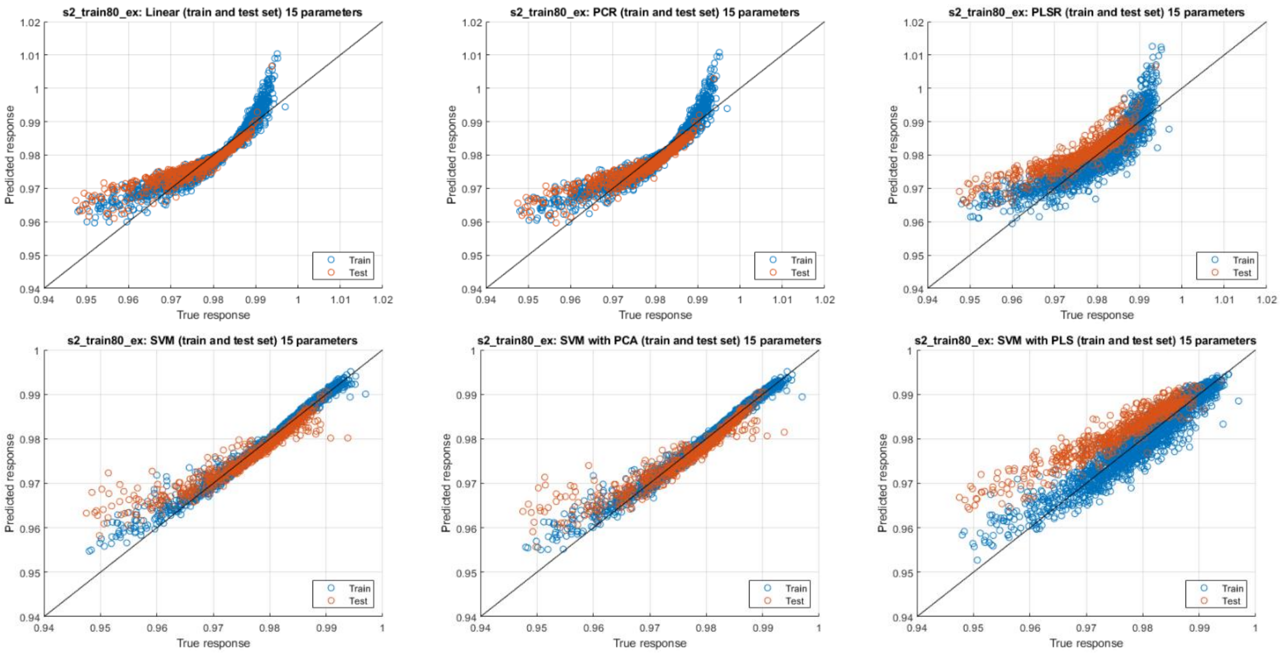


Figure 7.4.7: Case 80-20 model performance (scenario 2, 15 predictors)

7.5 Appendix E: Statistical indicators (15 predictors)

This section presents statistical indicators for all 6 models of all splitting cases with 15 predictors using test set in scenario 2 as shown in Table 7.5.1 to Table 7.5.7.

Table 7.5.1: Statistical indicators for all models (s2, case 50-50, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0039	0.0039	0.0050	0.0041	0.0029	0.0037
R^2	0.8139	0.8167	0.7131	0.8300	0.9161	0.8342
MARE	0.0029	0.0030	0.0039	0.0026	0.0019	0.0029
MaxARE	0.0188	0.0177	0.0206	0.0254	0.0194	0.0157

Table 7.5.2: Statistical indicators for all models (s2, case 55-45, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0039	0.0041	0.0050	0.0043	0.0030	0.0036
R^2	0.8039	0.8006	0.7092	0.8056	0.9070	0.8467
MARE	0.0030	0.0033	0.0039	0.0027	0.0022	0.0027
MaxARE	0.0181	0.0178	0.0193	0.0251	0.0178	0.0142

Table 7.5.3: Statistical indicators for all models (s2, case 60-40, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0041	0.0042	0.0051	0.0044	0.0030	0.0036
R^2	0.7970	0.7947	0.7050	0.8022	0.9062	0.8448
MARE	0.0032	0.0034	0.0040	0.0029	0.0022	0.0028
MaxARE	0.0181	0.0175	0.0201	0.0252	0.0176	0.0159

Table 7.5.4: Statistical indicators for all models (s2, case 65-35, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0043	0.0044	0.0052	0.0047	0.0032	0.0035
R^2	0.7873	0.7860	0.6996	0.7997	0.9055	0.8573
MARE	0.0034	0.0037	0.0041	0.0031	0.0024	0.0027
MaxARE	0.0181	0.0175	0.0193	0.0252	0.0173	0.0146

Table 7.5.5: Statistical indicators for all models (s2, case 70-30, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0044	0.0044	0.0055	0.0049	0.0033	0.0038
R^2	0.7804	0.7807	0.6925	0.7977	0.9039	0.8718
MARE	0.0035	0.0037	0.0040	0.0033	0.0025	0.0028
MaxARE	0.0185	0.0175	0.0203	0.0254	0.0182	0.0173

Table 7.5.6: Statistical indicators for all models (s2, case 75-25, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0046	0.0043	0.0061	0.0050	0.0034	0.0055
R^2	0.8005	0.8197	0.6970	0.8058	0.9097	0.8715
MARE	0.0038	0.0033	0.0044	0.0034	0.0023	0.0043
MaxARE	0.0188	0.0194	0.0221	0.0255	0.0206	0.0216

Table 7.5.7: Statistical indicators for all models (s2, case 80-20, 15 predictors)

Statistical Indicators	**Models**					
	Linear regression	**PCR**	**PLSR**	**SVM**	**SVM PCA**	**SVM PLS**
RMSE	0.0046	0.0044	0.0066	0.0037	0.0036	0.0069
R^2	0.8478	0.8511	0.7906	0.8921	0.8825	0.8922
MARE	0.0032	0.0033	0.0051	0.0023	0.0023	0.0060
MaxARE	0.0201	0.0191	0.0232	0.0220	0.0210	0.0217