A comparison study between different kernel functions in the least square support vector regression model for penicillin fermentation process

Jameson Malang^{*}, Wan Sieng Yeo, Zhen Yang Chua, Jobrun Nandong and Agus Saptoro

Department of Chemical and Energy Engineering, Curtin University Malaysia, CDT 250, 98009 Miri, Sarawak, Malaysia.

*jameson@curtin.edu.my

Abstract. Soft sensors are becoming increasingly important in our world today as tools for inferring difficult-to-measure process variables to achieve good operational performance and economic benefits. Recent advancement in machine learning provides an opportunity to integrate machine learning models for soft sensing applications, such as Least Square Support Vector Regression (LSSVR) which copes well with nonlinear process data. However, the LSSVR model usually uses the radial basis function (RBF) kernel function for prediction, which has demonstrated its usefulness in numerous applications. Thus, this study extends the use of nonconventional kernel functions in the LSSVR model with a comparative study against widely used partial least square (PLS) and principal component regression (PCR) models, measured with root mean square error (RMSE), mean absolute error (MAE) and error of approximation (E_a) as the performance benchmark. Based on the empirical result from the case study of the penicillin fermentation process, the E_a of the multiquadric kernel (MQ) is lowered by 63.44% as compared to the RBF kernel for the prediction of penicillin concentration. Hence, the MQ kernel LSSVR has outperformed the RBF kernel LSSVR. The study serves as empirical evidence of LSSVR performance as a machine learning model in soft sensing applications and as reference material for further development of non-conventional kernels in LSSVR-based models because many other functions can be used as well in the hope to increase the prediction accuracy.

1. Introduction

Industrial processes employ various hardware sensors such as flow rate, pressure and temperature for data delivery to control systems. This is for process monitoring and control to guarantee consistency in product quality. However, some crucial process variables are not easily measurable and require expensive sensors or offline laboratory testing with a significant time delay to ascertain product quality [1]. This coupled with drawbacks of hardware sensors such as sensor faults, low sampling frequency and maintenance requirements [2], as a result, many researchers attempt to construct a predictive model based on easily measured variables to estimate the difficult-to-measure variables, such predictive models are termed as soft sensors [3]. The potential of automatic control using soft sensors and the economic advantages in doing so leads to great interest in soft sensors from both academia and industry. However, there are still issues unresolved in soft sensors development with measurement noises, co-linear features, missing values, varying sampling rates and data outliers being the common issues [4]. Another issue is the dynamic nature of process plants, the mixture of gradual and abrupt changes in the process poses difficulty for soft sensors and often results in the degradation of prediction accuracy [5]. To mitigate these issues, traditional sensors usually tackle these problems with the linearised model, overdesigning

equipment and avoiding complex operating regimes, which these methods usually result in a loss of economic advantages [1].

Vapnik's Support Vector Machine (SVM), first designed for classification [6] and subsequently expanded to regression, is known as Support Vector Regression (SVR), and it has demonstrated considerable potential in applications to high dimensional nonlinear problems. This is because of its excellent generalisation capabilities over PLS-based methods with great performance under limited training data samples [7]. SVR, however, requires complex quadratic programming optimisation, leading to reduced modelling efficiency. Work from [8] incorporated the least-square method with SVR, later termed Least Square Support Vector Regression (LSSVR) has greatly improved modelling efficiency as the complex optimisation is replaced by solving series of linear equations, making LSSVR an ideal algorithm for large scale regression problem while maintaining the great performance of SVR when data are limited. The core idea behind LSSVR is to translate nonlinear samples from low-dimensional space into high-dimensional feature space by employing the kernel function, allowing the nonlinear samples to be partitioned linearly in this space to fulfil the fitting prediction requirement [9].

Due to the lower computational complexity of the optimisation process in LSSVR as compared to the SVR model, therefore, it has been broadly used in the prediction of quality variables in nonlinear processes. For instance, [9] developed an LSSVR model with a Gaussian kernel function that can accurately assess some anomalous observations that may take place in the estimated value and forecast GPS (Global Positioning System) signals with improved precision. Besides that, [10] managed to design a smart LSSVR model by adopting the Gaussian radial basis function (RBF) kernel function to address batch operations' time-varying, multiphase and nonlinear features. Nevertheless, all these studies did not compare the performance of different kernel functions in the LSSVR model in which kernel function is the premise of constructing an LSSVR model. Additionally, different kernel functions may result in varied accuracy levels for LSSVR models. The kernel function type must be carefully chosen in order to construct a highly accurate LSSVR model [11]. Despite that, the selection of kernel function was solely based on the researchers' prior knowledge and experience.

To date, extensive searches in many academic databases for comparison of LSSVR performance with different kernels show limited results. This work seeks to fill this research gap by expanding the use of kernel approaches in LSSVR for the prediction of process variables in the fermentation of penicillin. To evaluate the effectiveness of kernels in LSSVR, the predictive performance of the model with various kernels is assessed and compared with that of other well-established techniques, such as principal component regression (PCR) and partial least square regression (PLSR). By comparing the performance of different kernel functions in an LSSVR model, it is possible to identify the kernel function that yields the best performance for a given dataset. This can help machine learning practitioners to choose the most appropriate kernel function for their specific application, leading to better overall performance of the model.

2. Methodology

In this section, the kernel functions, LSSVR, PLSR, PCR, data splitting and parameter tuning as well as specifications of computer configuration are described.

2.1 Kernel functions

In the kernel functions, the input and output variables are denoted by x and y. Additionally, b is the kernel parameter and σ is the standard deviation (sigma). There are 12 different kernel functions used in this study and the equations are given in **Table 1**.

Table 1. The general form of kernel functions applied in this study [3].

Corresponding kernel	Kernel function
----------------------	-----------------

RBF (RBF)	$K(x, y) = \exp\left(-\frac{1}{2\sigma^2} \ x - y\ ^2\right)$
Linear (LIN)	$\mathbf{K}(x,y) = x^T y + \sigma$
Polynomial (POLY)	$\mathbf{K}(x,y) = (x^T y + 1)^b$
Sigmoidal (SIG)	$K(x, y) = \tanh (bx^T y + 1)$
Laplace (LAPLACE)	$K(x, y) = \exp\left(-\frac{\ x - y\ }{\sigma}\right)$
Rational Quadratic (RQ)	$K(x, y) = 1 - \frac{ x-y ^2}{ x-y ^2 + b}$
Multiquadric (MQ)	$K(x, y) = \sqrt{\ x - y\ ^2 + b^2}$
Inverse Multiquadric (INMQ)	$K(x, y) = \frac{1}{\sqrt{\ x - y\ ^2 + b^2}}$
Cauchy (CAUCHY)	$K(x, y) = \frac{1}{1 + \frac{\ x - y\ ^2}{\sigma^2}}$
Generalized T-student (GTS)	$K(x, y) = \frac{1}{1 + x - y ^{b}}$
Fourier (FOURIER)	$K(x, y) = \frac{1 - b^2}{2(1 - 2bcos(x - y)) + b^2}$
ANOVA (ANOVA)	$K(x,y) = \sum_{k=1}^{n} \exp\left(-\sigma(x^{k} - y^{k})^{2}\right)^{b}$

2.2 Least square support vector regression

The LSSVR algorithm was developed following [11] and it is shown as follows:

Step 1: Considering the SVM model of the form given in Equation (1).

$$Y = w^T \varphi(X) + B \tag{1}$$

where $\varphi(X)$ represents the high-dimensional and infinite-dimensional feature space mapping term, *B* stands for the bias term, and *w* stands for the weight vector with *m* dimension.

Step 2: Formulate the cost function in **Equation (2)**, which forms the optimisation problem in primal space.

$$\begin{cases} \min J(w, e) = \frac{1}{2} \left[w^T w + \gamma \sum_{k=1}^{N} e_k^2 \right] \\ \text{subjected to } Y_k = w^T \varphi(X_k) + B + e_k \ (k = 1, 2, \dots, N) \end{cases}$$
(2)

where e_k and γ stand for the training error for X_k and the regularisation constant, respectively.

Step 3: The optimisation problem in Equation (2) is not solvable when w is potentially infinitedimensional. To solve the problem, the Lagrangian function as expressed in Equation (3) is calculated in which the solutions of w and e are determined by using the Lagrange multiplier optimal programming method. The objective function can be attained by converting the constraint problem into a nonconstraint problem [12].

$$\mathcal{L}(w, b, e, \alpha) = J(w, e) - \sum_{k=1}^{n} \alpha_k \{ w^T \varphi(X_k) + B + e_k - Y_k \}$$
(3)

where α_k represents the Lagrange multiplier.

Step 4: Differentiate the Lagrangian by applying the Karush-Kuhn-Tucker conditions and the optimality conditions are shown in **Equation (4)**:

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow w = \sum_{k=1}^{N} \alpha_{k} \varphi(X_{k}) \\ \frac{\partial \mathcal{L}}{\partial B} = 0 \rightarrow \sum_{k=1}^{N} \alpha_{k} = 0 \\ \frac{\partial \mathcal{L}}{\partial e_{k}} = 0 \rightarrow \alpha_{k} = \gamma e_{k}, \ k = 1, 2, ..., N \\ \frac{\partial \mathcal{L}}{\partial \alpha} = 0 \rightarrow w^{T} \varphi(X_{k}) + B + e_{k} - Y_{k} = 0, \ k = 1, 2, ..., N \end{cases}$$

$$(4)$$

Step 5: The following solution is obtained when w and e are eliminated and is given in Equation (5):

$$\begin{bmatrix} 0 & 1_{\nu}^{T} \\ 1_{\nu} & \Omega + \frac{1_{\nu}}{\gamma} \end{bmatrix} \begin{bmatrix} B \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \gamma \end{bmatrix}$$
(5)

where $1_v = [1, 1, ..., 1]$, $Y = [Y_1, Y_2, ..., Y_N]$ and $\alpha = [\alpha_1, \alpha_2, ..., \alpha_N]$.

Step 6: The kernel trick based on Mercer's condition is applied to Ω as shown in Equation (6):

$$\Omega_{kl} = \varphi(X_k)^T \varphi(X_l)$$

$$\Omega_{kl} = K(X_k, X_l), \quad k, l = 1, 2, ..., N$$
(6)
where $K(X_k, X_l)$ denotes the kernel function.

Step 7: The LSSVR model is then presented in Equation (7):

$$Y = \sum_{k=1}^{N} \alpha_k K(x, x_k) + B \tag{7}$$

2.3 Partial least squares regression

PLSR is a multivariate regression technique that is suitable for analysing changes in a large number of highly correlated input variables *X* and connecting them to a set of output variables *Y*. PLSR deals with the relationship between *X* and *Y* both internally and externally. *X* is decomposed as follows [13]:

$$X = TP^{T} + E \tag{8}$$

where the input matrix is represented by $X \in \mathbb{R}^{m \times n}$, the score matrix is represented by $T \in \mathbb{R}^{n \times a}$, the loading matrix is represented by $P \in \mathbb{R}^{m \times a}$, and the noise matrix is $E \in \mathbb{R}^{m \times n}$ is. $X = \sum_{j=1}^{n} t_j p_j^T + E$ can be obtained by substituting TP^T with the sum of the product of the score vector t_j (the *j*th row of *T*) and the loading vector p_j (the *j*th row of *P*). Similarly, *Y* can be broken down into:

$$Y = UQ^T + F \tag{9}$$

where the output matrix is denoted by $Y \in \mathbb{R}^{n \times m}$, $U \in \mathbb{R}^{n \times a}$ stands for the score matrix, the loading matrix is $Q \in \mathbb{R}^{m \times a}$, and the noise matrix is $F \in \mathbb{R}^{n \times m}$. In the same way, $Y = \sum_{j=1}^{a} u_j q_j^T + F$ can be obtained by replacing UQ^T with the sum of the product of u_j and q_j . Assume that $\tilde{u}_j = b_j t_j$, where b_j stands for the regression coefficient, U = TB, and the regression matrix is represented by $B \in \mathbb{R}^{a \times a}$. The equation $Y = TBQ^T + F$ can be used to represent the relationship between X and Y.

2.4 Principal component regression

The goal of PCR is to identify a subset of all components in order to reduce the number of dimensions by decreasing the effective size of the original space. The derived model structure is [14]:

$$X = TP^{T} + E$$

$$Y = TC^{T} + E$$
(10)

where $X^{n \times m} = [x_1, x_2, ..., x_n]^T$, $Y^{n \times r} = [y_1, y_2, ..., x_n]^T$ are the measuring matrices built from the input and output variables, respectively. The sizes of the input and output variables are represented by *m* and *r*, while *n* represents the size of the data samples. $T^{n \times q}$ represents the principal component matrix where *q* stands for the selected size of latent variables, $P^{m \times q}$ represents the loading matrix and the regression matrix is represented by $C^{r \times q}$.

2.5 Data splitting and parameters tuning

From 1,500 data generated, the training and testing data are divided in a ratio of 75%:25%. [15]. Hence, N_t is 1,500 whereas the number of training and testing data are 1,125 and 375, respectively, which are represented by N_1 and N_2 . The LSSVR model regularisation parameter, γ and kernel parameter, b are tuned using Leave-one-out (LOO) procedure. The appropriate values for the upper and lower boundaries are set to the following values in the grid region: $\gamma \in \{2^{-5}, 2^{-3}, ..., 2^{15}\}$ and $b \in \{2^{-15}, 2^{-13}, ..., 2^{8}\}$ [16]. Searching for the best combination of parameters is crucial in this stage so that the model can accurately make a prediction [17].

2.6 Prediction accuracy measurement

For the measurement of quality prediction, the RMSE and MAE error metrics are used as measurement metrics of error. These two metrics are found in the evaluation of performance for both the machine-learning model [18] and the non-machine learning-based model [19]. Given that the model uses two error metrics, and that the metrics are often not ranked equally, the E_a helps determine the best possible

model. A lower E_a suggests that more models would be more closely related to the dataset's real nature. [20]. The formula of RMSE, MAE and E_a are demonstrated in **Equations (11)** to **(13)** ([3], [18], [21]).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N_t} (\hat{y}_i - y_i)^2}{N_t}}$$
(11)

$$MAE = \frac{\sum_{i=1}^{N_t} |\hat{y}_i - y_i|}{N_t}$$
(12)

$$E_{a} = \left(\frac{N_{1}}{N_{t}}\right)RMSE_{1} + \left(\frac{N_{2}}{N_{t}}\right)RMSE_{2} + |RMSE_{1} - RMSE_{2}|$$
(13)

2.7 Specifications of computer configuration

The computer setup required to execute the LSSVR, PLS and PCR models in this study is described in **Table 2**.

Computer setup	Details		
Central processing unit (CPU)	Intel(R) Core(TM) i5-1135G7 (2.40GHz) 1.38 GHz processor		
Operating system (OS)	Windows 11 (64-bit)		
Random-access memory (RAM)	8.00 GB		
MATLAB version	R2022a		

Table 2. The details of the computer setup employed in this study.

3. Results and discussions

The industrial-scale penicillin fermentation simulation (IndPenSim), whose data are publicly accessible for download at www.industrialpenicillinsimulation.com and perfectly suitable for the development of data analytics, machine learning, or artificial intelligence algorithms, is chosen as the case study. [22]. A detailed description of the fermentation process for penicillin production can be found in [23]. The input variables and one output variable employed in this case study are shown in **Table 3**. The results of the 12 kernels investigated as well as PLSR and PCR models for this case study are tabulated in **Table 4**. The errors are ranked in ascending order of Ea as it indicates the performance of the model.

Table 3. The input variables and output variable employed in this case study.

Input variables	Output variable
Temperature, fermentor temperature, pH, agitator power, aeration rate, substrate feed rate, carbon dioxide concentration, biomass concentration, substrate concentration, dissolved oxygen concentration and culture volume	penicillin concentration

Table 4. Predictive performance of LSSVR models using different kernel functions, PLSR and PCR.

Models	Training data		Testing data				
	RMSE ₁	MAE ₁	RMSE ₂	MAE ₂	Ea		
LSSVR with MQ	1.14×10 ⁻⁵	2.24×10 ⁻⁶	0.0293	0.0266	0.0366		
LSSVR with SIG	0.0661	0.0559	0.0684	0.0579	0.0690		
LSSVR with LIN	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with LAPLACE	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with RQ	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with INMQ	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with GTS	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with ANOVA	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with RBF	0.0433	0.0366	0.0887	0.0822	0.1001		
LSSVR with POLY	0.3575	0.2555	0.0863	0.0848	0.5610		
LSSVR with FOURIER	0.0011	8.21×10-4	1.1112	0.8828	1.3887		
LSSVR with CAUCHY	6.85×10 ⁻⁴	4.84×10 ⁻⁴	1.02×10 ⁻⁶	1.2098	1.5120		
Other regression models							
PLSR	0.2061	0.1748	0.4324	0.4268	0.4889		
PCR	0.2379	0.1973	0.4500	0.4440	0.5031		

Of all the kernels investigated, the MQ kernel function gives the lowest RMSE and MAE for both training and testing datasets as well as the lowest E_a values as bolded in **Table 4** (RMSE₁ = 1.14×10^{-5} , RMSE₂ = 0.0293, MAE₁ = 2.24×10^{-6} , MAE₂ = 0.0266 and $E_a = 0.0366$) when the kernel parameter, *b* is tuned at 2^{-3} . It demonstrates that MQ has the highest ability to forecast the nonlinear penicillin fermentation process as compared to other models. Followed by the SIG kernel with RMSE₁ = 0.06661, RMSE₂ = 0.0684, MAE₁ = 0.0559, MAE₂ = 0.0579 and $E_a = 0.0690$ with *b* tuned at 2^{-15} . By comparing the results of MQ and SIG with RBF kernel in terms of E_a , it is revealed that MQ and SIG have improved the results by 63.44% and 31.07%, respectively.

It is interesting to note that the LIN, LAPLACE, RQ, INMQ, GTS, ANOVA and RBF kernels are found to give similar results for all the error metrics which are $RMSE_1 = 0.0433$, $RMSE_2 = 0.0887$, $MAE_1 = 0.0366$, $MAE_2 = 0.0822$ and $E_a = 0.1001$. The reason most likely has to do with the size of the feature space after kernel transformation in which the transformation is virtually identical [24]. Besides, the similar performance of LIN and RBF kernels shows that the LIN kernel function which is also a popular kernel used by many researchers due to its advantage of small computational requirements ([25],[26]) is comparable with the RBF kernel. Based on the aforementioned, it is also noted that many kernel functions have outperformed or shown similar performances as compared to the commonly used RBF kernel, which reinforces the need for research in LSSVR kernels to allow an appropriate selection of kernels.

Generally, as presented in **Table 4**, the LSSVR models with the appropriate kernels have also outperformed the conventional PLSR and PCR models. However, it is not always true since it can be seen that the LSSVR models equipped with POLY, FOURIER and CAUCHY kernels are found to perform worse than PLSR and PCR with the values of E_a of 0.5610, 1.3887 and 1.5120, respectively. This may be due to the parameters used in this study that minimises the model error [27]. Although it can be utilised in the analysis of multidimensional spaces, the CAUCHY kernel is discovered to be the worst kernel in this application with the highest value of $E_a = 1.5120$ and may not be effective for the prediction of nonlinear processes.

In this case study, PLSR displays a better result than PCR with a prediction accuracy of 2.82% improvement probably due to the adverse effect of systematic variation in the input variable matrix which is not related to the output variable matrix. In other words, there exists systemic variation that is

not part of the joint correlation structure of the input and output variable matrix, and it negatively affects the model performance [28].

To elucidate this matter, the plots of actual and predicted penicillin concentration (output variable) from the LSSVR model with selected kernel functions for both training and testing data are presented in **Figures 1** and **2**. In these two figures, only the selected kernel functions i.e., RBF, MQ, CAU, POLY, FOU and SIG together with PCR as well as PLSR models are plotted since some kernel functions exhibited the same results as each other as evident in **Table 4**. From the indication in **Figure 1**, it is worth noting that most LSSVR models with different kernel functions (except for the POLY kernel) can better predict the penicillin concentration for the training dataset as compared to PLSR and PCR models. This is further justified by the small values of RMSE in **Table 4**. However, the POLY kernel has shown worse performance in the prediction of output variables for both training and testing datasets as can be observed in **Figures 1** and **2** although the E_a for the POLY kernel is not the lowest among all the kernel functions studied. This may be due to the POLY kernel dealing best with the normalised [24].



Figure 1. Graph of actual and predicted output variables of the training dataset for selected kernel functions.



Figure 2. Graph of actual and predicted output variables of the testing dataset for selected kernel functions.

For the testing dataset in **Figure 2**, MQ has the closest output prediction as compared to RBF and this has strengthened the earlier findings from **Table 4** that MQ has outperformed the conventional RBF kernel. POLY kernel once again has shown poor prediction with no particular trend observed and underpredicts the output variable. As for the FOU kernel, the prediction seems fine within the first 35 samples but worsens thereafter. Henceforth, it indicates that these two kernels are not suitable for the prediction of penicillin concentration. Based on this case study, different kernel functions may perform better as evidenced in this penicillin fermentation process even though RBF is frequently used in SVM-based models. This resonates with research suggesting that while RBF kernels are usually applied for data transformation in kernel mapping, there is a certain dataset where RBF is outperformed by other types of kernels [29].

4. Conclusion

The research has extended the LSSVR model with non-conventional kernels beyond typical RBF, linear and polynomial kernels. Comparative studies of the models are tested with one industrially validated penicillin fermentation simulation. Comparison of the LSSVR model against the PLSR and PCR model is also performed as a benchmark of machine learning-based soft sensors against conventional soft sensor models.

A comparison of the LSSVR model with the different kernels is performed, and it is found that whilst MQ and SIG kernels have the potential to outperform the conventional and popular kernels such as LIN, POLY and RBF kernels, but the model performance metrics used in this study which are RMSE and MAE generally have high variance and non-convex in nature. This suggests that while it is possible for MQ and SIG kernels to surpass the widely used kernels, however, it would require careful development of a hyperparameter optimisation algorithm as well. Since the main idea of the kernel function is to optimise the objective function in the LSSVR model, therefore, kernel function with small error is crucial in helping to minimise the cost function. However, the widely used RBF kernel still performs properly although other kernels such as MQ and SIG may perform even better in this case study.

In conclusion, the LSSVR model with the appropriate kernel can outperform conventional PLS and PCR models as well as the LSSVR model with RBF kernel function, with MQ and SIG kernel potentially being superior under certain circumstances based on the case study conducted. The result of LSSVR outperforming the PLS and PCR model may require more case studies in other industrial processes to validate that LSSVR performance generalises well to industrial processes. Besides, the study of the optimal amount of training and testing samples in ensuring the optimum prediction accuracy can also be investigated in choosing a suitable kernel function.

References

- [1] Gargalo, C. L., Udugama, I., Pontius, K., Lopez, P. C., Nielsen, R. F., Hasanzadeh, A., ... & Gernaey, K. V. (2020). Towards smart biomanufacturing: a perspective on recent developments in industrial measurement and monitoring technologies for bio-based production processes. *Journal of Industrial Microbiology & Biotechnology: Official Journal of the Society for Industrial Microbiology and Biotechnology*, 47(11), 947-964.
- [2] Liu, Y., & Xie, M. (2020). Rebooting data-driven soft-sensors in process industries: A review of kernel methods. *Journal of Process Control*, 89, 58-73.
- [3] Ngu, J. C. Y., & Yeo, C. (2022). A Comparative Study of Different Kernel Functions Applied to LW-KPLS Model for Nonlinear Processes. *Biointerface Research in Applied Chemistry*, 13(2).
- [4] Souza, F. A., Araújo, R., & Mendes, J. (2016). Review of soft sensor methods for regression applications. *Chemometrics and Intelligent Laboratory Systems*, 152, 69-79.
- [5] Jin, H., Chen, X., Yang, J., & Wu, L. (2014). Adaptive soft sensor modeling framework based on just-in-time learning and kernel partial least squares regression for nonlinear multiphase batch processes. *Computers & Chemical Engineering*, 71, 77-93.
- [6] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20(3), 273-297.
- [7] Curreri, F., Fiumara, G., & Xibilia, M. G. (2020). Input selection methods for soft sensor design: A survey. *Future Internet*, 12(6), 97.
- [8] Suykens, J. A., De Brabanter, J., Lukas, L., & Vandewalle, J. (2002). Weighted least squares support vector machines: robustness and sparse approximation. *Neurocomputing*, *48*(1-4), 85-105.
- [9] Xiaoming, L., Xinglong, T., & Changsheng, Z. (2019). PSO-LSSVR assisted GPS/INS positioning in occlusion region. *Sensors*, 19(23), 5256.
- [10] Sheng, X., & Xiong, W. (2020). Soft sensor design based on phase partition ensemble of LSSVR models for nonlinear batch processes. *Mathematical Biosciences and Engineering*, 17(2), 1901-1921.
- [11] Adnan, R. M., Liang, Z., Yuan, X., Kisi, O., Akhlaq, M., & Li, B. (2019). Comparison of LSSVR, M5RT, NF-GP, and NF-SC models for predictions of hourly wind speed and wind power based on cross-validation. *Energies*, 12(2), 329.
- [12]Kisi, O., & Parmar, K. S. (2016). Application of least square support vector machine and multivariate adaptive regression spline models in long term prediction of river water pollution. *Journal of Hydrology*, 534, 104-112.
- [13] Bao, L., Yuan, X., & Ge, Z. (2015). Co-training partial least squares model for semi-supervised soft sensor development. *Chemometrics and Intelligent Laboratory Systems*, 147, 75-85.
- [14] Zhu, P., Liu, X., Wang, Y., & Yang, X. (2018). Mixture semisupervised Bayesian principal component regression for soft sensor modeling. *IEEE Access*, 6, 40909-40919.
- [15] Mickel, V. M., Yeo, W. S., & Saptoro, A. (2019). Evaluating the Performance of Newly Integrated Model in Nonlinear Chemical Process Against Missing Measurements. *Chemical Product and Process Modeling*, 14(4).

- [16] Xu, S., An, X., Qiao, X., Zhu, L., & Li, L. (2013). Multi-output least-squares support vector regression machines. *Pattern Recognition Letters*, 34(9), 1078-1084.
- [17] Achirul Nanda, M., Boro Seminar, K., Nandika, D., & Maddu, A. (2018). A comparison study of kernel functions in the support vector machine and its application for termite detection. *Information*, 9(1), 5.
- [18] Thien, T. F., & Yeo, W. S. (2022). A comparative study between PCR, PLSR, and LW-PLS on the predictive performance at different data splitting ratios. *Chemical Engineering Communications*, 209(11), 1439-1456.
- [19] Tange, R. I., Rasmussen, M. A., Taira, E., & Bro, R. (2017). Benchmarking support vector regression against partial least squares regression and artificial neural network: Effect of sample size on model performance. *Journal of Near Infrared Spectroscopy*, 25(6), 381-390.
- [20] Yeo, W. S. (2019). Adaptive Soft Sensors for Non-Gaussian Chemical Process Plant Data Based on Locally Weighted Partial Least Square (Doctoral dissertation, Curtin University).
- [21] Wang, W., & Lu, Y. (2018, March). Analysis of the mean absolute error (MAE) and the root mean square error (RMSE) in assessing rounding model. In *IOP conference series: materials science and engineering* (Vol. 324, No. 1, p. 012049). IOP Publishing.
- [22] Goldrick, S., Ştefan, A., Lovett, D., Montague, G., & Lennox, B. (2015). The development of an industrial-scale fed-batch fermentation simulation. *Journal of biotechnology*, *193*, 70-82.
- [23] Goldrick, S., Duran-Villalobos, C. A., Jankauskas, K., Lovett, D., Farid, S. S., & Lennox, B. (2019). Modern day monitoring and control challenges outlined on an industrial-scale benchmark fermentation process. *Computers & Chemical Engineering*, 130, 106471.
- [24] Drewnik, M., & Pasternak-Winiarski, Z. (2017, June). SVM kernel configuration and optimization for the handwritten digit recognition. In *IFIP International Conference on Computer Information Systems and Industrial Management* (pp. 87-98). Springer, Cham.
- [25] Tang, Q., Qiu, W., & Zhou, Y. (2019). Classification of complex power quality disturbances using optimized S-transform and kernel SVM. *IEEE Transactions on Industrial Electronics*, 67(11), 9715-9723.
- [26] Nguyen, H., Bui, X. N., Choi, Y., Lee, C. W., & Armaghani, D. J. (2021). A novel combination of whale optimization algorithm and support vector machine with different kernel functions for prediction of blasting-induced fly-rock in quarry mines. *Natural Resources Research*, 30(1), 191-207.
- [27] Mehmani, A., Chowdhury, S., Meinrenken, C., & Messac, A. (2018). Concurrent surrogate model selection (COSMOS): optimizing model type, kernel function, and hyper-parameters. *Structural* and Multidisciplinary Optimization, 57(3), 1093-1114.
- [28] Trygg, J., & Lundstedt, T. (2007). Chapter 6—chemometrics techniques for metabonomics. *The handbook of metabonomics and metabolomics*, 171-199.
- [29] Wang, B. W., Tang, W. Z., Song, L. K., & Bai, G. C. (2020, December). PSO-LSSVR: A surrogate modeling approach for probabilistic flutter evaluation of compressor blade. In *Structures* (Vol. 28, pp. 1634-1645). Elsevier.