Applied Soft Computing Journal 164 (2024) 111975

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A comprehensive overview of the applications of kernel functions and data-driven models in regression and classification tasks in the context of software sensors

Joyce Chen Yen Ngu [a], Wan Sieng Yeo [a,b,*], Teck Fu Thien [a], Jobrun Nandong [a]

[a] Department of Chemical and Energy Engineering, Faculty of Engineering and Science, Curtin University Malaysia, CDT 250, 98009 Miri, Sarawak, Malaysia
[b] Curtin Malaysia Research Institute, Curtin University, Malaysia

HIGHLIGHTS

- Progress and gaps in kernel functions and data-driven models are presented.
- Kernel functions for classification and regression are critically reviewed.
- Integrating kernel functions with adaptive data-driven models is recommended.

ARTICLE INFO

Keywords:
Data-driven model
Kernel functions
Classification
Nonlinear
Soft sensor

ABSTRACT

Data-driven models can reduce the number of hardware sensors in a process plant by acting as low-cost substitutes for hardware sensors. Since some data-driven models have difficulty dealing with nonlinear data, kernel functions have been integrated into the data-driven models due to their capability to handle this nonlinear behavior of data. However, the existing review studies on kernel functions and data-driven models for regression and classification are still limited. Moreover, for kernel functions, most research studies have only focused on the radial basis function group, such as gaussian and hyperbolic tangent kernel functions. Considering these research gaps, this review study aims to summarize the most up-to-date cumulative studies on kernel functions, their application categories, and their integration with data-driven models. Different from other existing review studies, this study discussed the characteristics, advantages, and disadvantages of different kernel functions. Additionally, this study also summarizes and critically reviews data-driven models for regression and classification tasks, including their advantages and disadvantages. Moreover, this review discovers the state of the art of different kernel functions that were used in data-driven models for regression and classification. Besides, this review study also found that the existing studies on kernel functions are mostly used for the classification task rather than the regression task. In addition, the gaussian kernel is found to be the most applied kernel function in various applications. Lastly, it is recommended to emphasize integrating different kernel functions with adaptive data-driven models in different industrial applications.

1. Introduction

Machine learning (ML) models use computer systems to develop an algorithm that can forecast the desired output variable(s) of the input variables [1–3]. It has been applied in various areas to accelerate traditional optimization procedures and minimize the time and cost of simulation [4–6]. Based on the models, the computer systems perform a

Abbreviations: CNN, Convolutional neural networks; COA, Cut-off area; G, Graphene; GPR, Gaussian process regression; JIT, Just-in-time; KPCA, Kernel principal component analysis; KPLSR, Kernel partial least squares regression; LSSVR, Least square support vector regression; LW-KPLSR, Locally weighted partial kernel partial least squares regression; LW-PLSR, Locally weighted partial least squares regression; ML, Machine learning; MLR, Multiple linear regression; NIPALS, Nonlinear iterative partial least squares; NIRS, Near-infrared spectroscopy; PC, Principal component; PCA, Principal component analysis; PCR, Principal component regression; PLSR, Partial least squares regression; RBF, Radial basis function; SIMPLS, Statistically inspired modification of partial least squares; SVM, Support vector machine; SVR, Support vector regression; WWTP, Wastewater treatment plant.

* Correspondence to: Department of Chemical and Energy Engineering, Curtin University, CDT 250, Miri 98009, Sarawak, Malaysia.
E-mail address: 256830B@curtin.edu.au (W.S. Yeo).

https://doi.org/10.1016/j.asoc.2024.111975
Received 14 January 2023; Received in revised form 15 June 2024; Accepted 3 July 2024
Available online 10 July 2024

Symbols

$\widehat{X}$	Dominant information
b	Kernel parameter
B	Regression coefficient matrix
C	Constant value or intercept
E	Residual matrices
E_1	Regression residual
f	Target function
k	Number of principal components
m	Number of input variables
n	Number of samples
P	Loading matrix
p1	Loading vector
R	Number of output variables
T	Score matrix
T	Transpose of a vector or matrix
t_1	Score
X	Input
x_i	Historical data
x_q	Query data
X_{train}	Known inputs
Y	Output
$Y_{estimate}$	Estimate outputs
Y_{real}	Known outputs
β	slope coefficient or weight,
β'	Modified terms or weights
λ	Eigenvalue
Φ	Feature matrix
Ω	Similarity matrix
Θ	Size of the dimension
ε	Noise parameter

learning process instead of utilizing human effort that manually identifies the correlation between the input and output variables. Therefore, ML has found increasing applications in complex or big data to give inferences about the required output in response to several inputs. ML algorithms have been widely employed in various applications [7-10]. All algorithms can roughly take the following formula; the only difference among different methods is how the target function, *f* is determined via Eq. (1).

$$Y = f(X) \tag{1}$$

The data goes along with the Central Limit Theorem, in which the distribution of average values for a sufficient sample size will be closely normal. Before developing an ML model using data, the complexity of the learning process can be reduced. As a result, the developed models often follow a theorem that leads to the use of parametric models [11]. For instance, these parametric types of models are linear regression, principal component regression (PCR), and partial least squares regression (PLSR) [12,13]. Meanwhile, non-parametric type models that do not assume any distribution are called non-parametric or distribution-free tests, and they are only applicable to simple designs [14]. Moreover, different from the parametric model, non-parametric type models do not need distribution parameters like mean and standard deviation to get an algorithm [11,14]. Also, they are generally less versatile than parametric ones since they have less supporting evidence while concluding the *f* [14].

Besides assumptions, developing a model also depends on different learning methods, which include supervised, unsupervised, semi-supervised, and reinforcement learning. A supervised learning machine learns the *f* to find a relationship between input and output variables. These learning algorithms are mainly categorized into regression, for which the former predicts continuous values, and classification for categorizing input data, and the selected task is subjected to the targeted objectives [1,15,16]. Unlike the supervised learning type, unsupervised ML uses input only in the training, where the objective is to identify underlying structures or distributions within the data [17]. Unsupervised learning models can fall into one of two groups, either clustering or dimensionality reduction [1].

Researchers have also developed semi-supervised ML methods that use a combination of input and partially labeled data to train a model. This learning method is the most common due to the time and resource-consuming process of labeling data [18]. According to He et al. [18], there are four types of semi-supervised models: generative, disagreement-based, graph-based, and support vector machine (SVM) models. Reinforcement learning involves determining the *f* in Eq. (1) using a series of rewards and penalties [19]. The algorithm has a degree of freedom to explore a predefined learning space [1]. This process is known as the Markov decision process, in which feedback information flows through learning agents and the environment [20].

On the other hand, a soft sensor uses the above-mentioned ML methods for prediction. As observed in various fields of industrial application, soft sensors, or so-called inferential sensors, play a significant role as an important tool. It is a model that functions to measure the output variables, which are hard to acquire in an industrial process [21]. However, when there are changes in the process characteristics, the approximation performance of soft sensors will worsen [22]. In the field of the process industry, the term "soft sensor" is mainly used, but the effectiveness of mathematical modeling in the engineering field is rife. Soft sensors allow reliable and consistent assessment of parameters such as wastewater parameters and have become a prominent approach in the field of ML [23,24]. According to Meng et al. [25], the combination of computer technology and knowledge of industrial processes is achieved by applying soft sensor technology spontaneously in the field of the sugar industry. The technology of utilizing software in the industry is adopted instead of hardware. To predict the dominant variables, instrumental variables are used, which increases the possibility of unpredictable variable measurement.

It can be concluded that the practice of soft sensors is more widespread compared to hardware sensors in industrial process plants. Soft sensors have been widely studied in process monitoring and control for processing industrial plants. In industrial processes, they play a role as a predictive model based on the availability of massive amounts of data [26,27]. Due to the limitations of hardware measuring appliances, which may be expensive or unavailable, they are primarily responsible for the online prediction of certain variables that are critical to quality control and production safety [28]. In addition to reducing operating costs, estimations from soft sensors are delay-free and accommodate complex industrial processes that may not be well-understood [29–31].

The soft sensors can be sorted into two main categories: model-driven and data-driven soft sensors. By referring to the Kalman filter, which is the first principle model, it is considered part of the family of model-driven soft sensors [21]. As industry processes in the real world are made of a complex system consisting of unwell-understood process mechanisms, the system identification techniques developed a data-driven soft sensor model, to obtain the correlation among variables [32]. Thus, the soft sensor of data-driven systems is presented as an alternative method.

Moreover, kernel functions allow the mapping of nonlinear datasets into a high-dimensional feature space [33–35]. This is known as a nonlinear projection that produces simulated linear behavior in the feature space [33–36]. A kernel function is a mathematical tool used to translate input data into a higher-dimensional space. A linear algorithm operating within the higher dimensional space will behave nonlinearly in the original input space [37]. It is a powerful tool, as relationships between nonlinearity, linearity, and algorithms can be established solely through dot products between vectors [37]. Furthermore, kernel functions are beneficial in different appliances, such as supplying an easy bridge between linearity and nonlinearity for algorithms that can exist in the form of dot products [37,38].

When applied appropriately, linear algorithms can be converted into nonlinear algorithms. In simple terms, a dataset forming a nonlinear line within a lower dimension may be converted into a linear 2-dimensional plane within a higher-dimensional space. These nonlinear algorithms exhibit an approximate equivalence to their respective linear originals within the limit space of the feature space [33–35]. On the other hand, the primary ideology of kernel parameter optimization took part in utilizing a soft sensor algorithm to learn a component of the instrumental variable principal under the kernel partial squares, which mainly serves to reduce the deviation of cross-validation to a minimum. Hence, the optimum kernel function parameters are acquired, and the behavior transformation matrix of the instrumental variable kernel function matrix will be kept [25,39].

Particularly, the primary focus of this study is a combination of kernel function and data-driven regression models. The scope and objectives of the review article are to present the concepts, techniques, and applications of kernel methods and data-driven models in regression and classification tasks. The structure of the review is as follows: Section 2 describes the data acquisition and pre-processing before model design and development. Then, the following Sections 3 to 4 explain data-driven model development and design and provide a review of various data-driven models for soft sensors. Next, Section 5 discusses several kernel functions that have found usage in data-driven models. Since nonlinearity is very common in the process industry, the following Sections 6 to 7 show the general design methodology, with an emphasis on basic data-driven regression models for soft sensors to support the use of kernel functions. Finally, Sections 8 to 9 provide future research challenges and directions on wastewater treatment, as well as conclusions and recommendations for future works. The flow chart of the review is summarized in Fig. 1.

During the identification stage in Fig. 1, a total of 58,064 research articles appeared through the search term. At the screening stage, to enhance the transparency of the review process, the article selection process applies three criteria, which are articles or proceeding papers published from the year 2015 onwards, as well as engineering research areas. Applying these criteria reduced the number of articles to 8710. After that, in the eligibility stage, article title and summary verification were conducted by determining the keyword of the "kernel function", which then narrowed the selection to 2233 articles. At the final stage, 2139 papers that were less relevant to the main research area of this study, i.e., not focusing on chemical engineering applications, were excluded. As a result, the corpus includes 94 articles.

2. Data acquisition and pre-processing

Acquired datasets are often rich in quantity but may fail to provide the desired useful information [40]. In this case, the raw data frequently contains impurities such as locked variables, drifted values, and missing

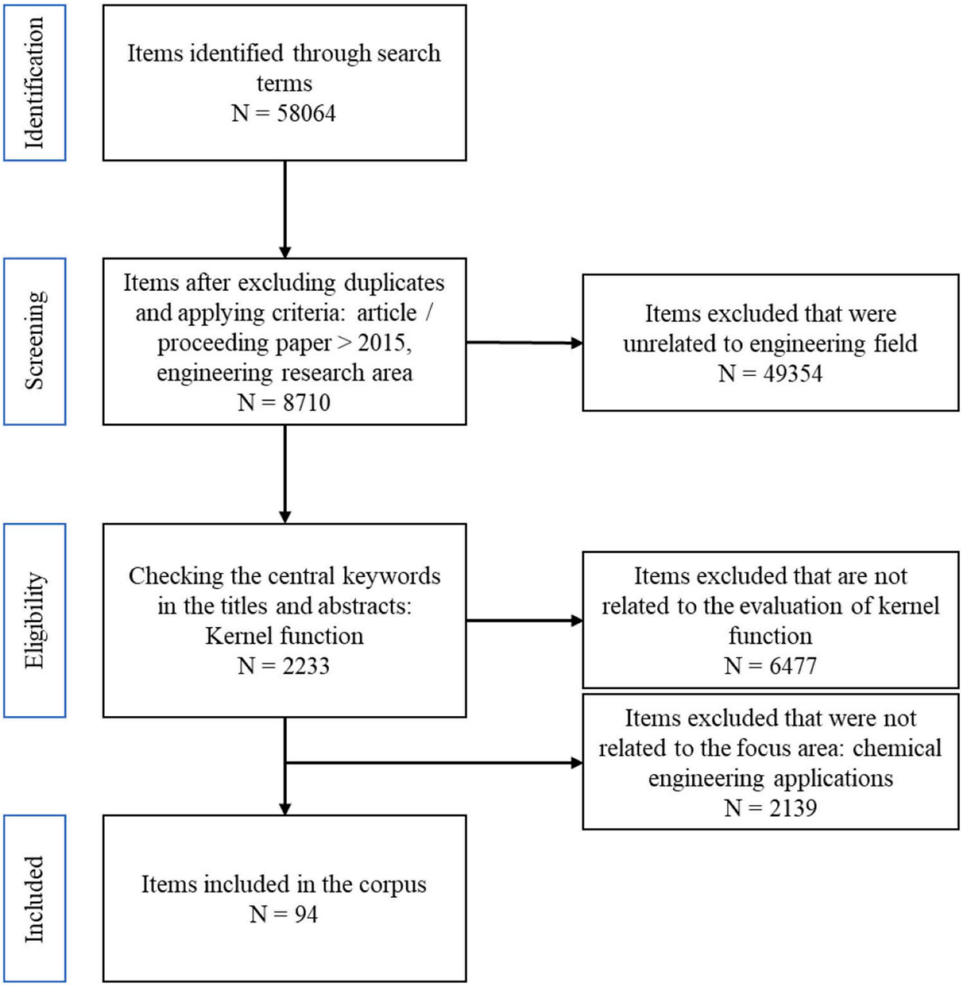
Identification
Items identified through search terms N = 58064
Screening
Items after excluding duplicates and applying criteria: article / proceeding paper > 2015, engineering research area N = 8710
Items excluded that were unrelated to engineering field N = 49354
Eligibility
Checking the central keywords in the titles and abstracts: Kernel function N = 2233
Items excluded that are not related to the evaluation of kernel function N = 6477
Items excluded that were not related to the focus area: chemical engineering applications N = 2139
Included
Items included in the corpus N = 94

Fig. 1. Flow chart of process of reviewed articles selection in the review.

measurements [15,40–43]. Initial inspections of the raw data can provide an understanding of the dataset's main structure. Hence, it helps to identify characteristics such as redundancies, time delays in measurements, and seasonal disturbances [15]. Nevertheless, data inspection requires manual exploration of analytical charts and basic knowledge of the underlying processes [15,40,44].

Collected datasets are usually noisy in addition to possessing redundant and corrupted information. Pre-processing is an essential step before model design, and it requires prior knowledge, as experts can obtain valuable information on the relevance of samples in the dataset [15]. Pre-processing involves optimizing the data to allow the model to learn more effectively [41]. The typical steps in pre-processing are normalization of data so that it has a zero mean and zero-unit variance, rectification of missing data, outlier detection, removal of redundant data, feature selection, correcting drifted data, and detection delays between important variables [5,15,41]. Some popular techniques in pre-processing are sample selection and variable selection [15]. Additionally, in pre-processing, kernel methods are used to project the data onto higher-dimensional spaces, where linear methods are more likely to be applicable. Hence, it can handle the nonlinear patterns in the data, while maintaining the computational elegance of matrix algebra [45–48]. In plant data analysis, it is mostly applied in the feature extraction step in the context of process monitoring.

3. Data-driven model development and design

After pre-processing, the next phase of developing a soft sensor involves designing the required model. This phase is crucial, as the model selection depends on the design requirements and whether the selected model fulfills the intended objectives [15]. So far, a methodological approach for the model design phase does not exist. Thus, one frequently chooses the model parameters and structure on an ad hoc basis (as needed) [15,41]. For this reason, the model design phase often relies on the intuition and experience of the designer, i.e., depending on the personal field of expertise or process-specific knowledge that can assist one's understanding of the model scope and objectives [15,29,41,49].

The design phase consists of a few main tasks: model selection, training, validation, and testing. The widespread practices generally require evaluating simple structures such as multiple linear regression (MLR) and their performance before increasing model complexity until no significant improvement in predictive performance is attainable [15,29,41,49]. Note that the main applications of soft sensors are for the online prediction of process variables with low sampling rates, i.e., long measurement deadtimes [29]. To operate the given soft sensor, easily measured (secondary) variables must be available to predict and provide inferences about the hard-to-measure (primary) variables. Here, the model for the soft sensor must be predictive, for which ML can be vital for the model's training, testing, and validation [50]. In general, supervised ML can help to build the desired predictive models in the form of regression or classification [15].

4. Data-driven regression models for soft sensors

In an industrial plant, chemical processes are usually multivariable. Hence, control systems must deal with multiple inputs and outputs [51]. Due to this multivariable characteristic, the control systems within the plant are interlinked. It means that the responses to control equipment are highly dependent on measured variables obtained from another source. Typically, easy-to-measure variables such as flow rate have quick process control responses because the measured signal will swiftly pass to an actuator for immediate corrective action. On the other hand, difficult-to-measure variables have slow control responses to the long measurement deadtimes where more resources are required to implement the desired control system [30,52]. This property is often seen in quality control-related requirements such as concentration and melt index [30,53].

Chemical processing industries are equipped with monitoring and control systems to meet the requirements of production effectiveness, product quality, and process safety [40]. Quality-related variables are measured using hardware sensors to provide information for process monitoring and control. Nevertheless, control engineers encounter many issues when using hardware sensors in process plants, as direct measurements may be very costly and time-consuming [52]. According to Kano and Fujiwara [54], a previous survey conducted in Japan reported that the three biggest problems with hardware sensors are aging deterioration (15 %), the need for calibration (21 %), and time-consuming maintenance (27 %). These issues increase the cost of production and reduce product quality, as well as process safety [54].

Therefore, soft sensors become a feasible alternative to overcome the issues with hardware sensors. Soft sensors are software-based inferential models that use appropriate input variables to infer the output variables of similar processes. Ookita [55] studied the focus of utilizing soft sensors at Mitsubishi Chemical Corporation. From their study, 37 % of the soft sensors mainly functioned for stabilizing the quality of products through online estimation, 28 % of the soft sensors were to decrease material and energy consumption, and 18 % were to enhance the reliability of online analyzers [55]. Inferential models such as soft sensors supplement this issue by using easy-to-measure (secondary) variables to estimate difficult-to-measure (primary) variables [21,25,30,56,57].

Generally, two main soft sensor types are currently of industrial interest: data-driven and the first principle, or fundamental model-driven models [29,41,58]. The first principle models are also known as white-boxes [29]. The fundamental model-based techniques are preferable for the development and planning stages of process plant design [41]. Developing the first principal models requires extensive time and resources due to reliance on a detailed understanding of the process background (e.g., chemical reactions, mass-energy balance, heat-mass transfer). Thus, the model becomes increasingly difficult to realize with increasing process complexity [29]. Alternatively, data-driven (black-box) models might serve the same purpose but are easier to realize because their development only requires empirical observations. These models involve utilizing a collected dataset to predict the outcomes of desired variables [15,29,41,44]. Hybrid (grey-box) models incorporate model-driven and data-driven techniques and have shown the potential to overcome certain limitations of individual models [15]. An example of a grey-box model application is when a model-driven soft sensor supplements a complicated process background with empirical data [41,52]. Table 1 shows a summary of the concepts, functions, computational loads, advantages, and disadvantages of soft sensors.

Notice in Table 1 that there are three model types, which include data-driven, model-driven, and hybrid of these two models. Model-driven methods have gained recognition as accurate models since they can provide a better understanding of the specified quality parameters. Therefore, the models are helpful in the development and planning stages. However, model-driven techniques are relatively complex and computationally intensive compared to data-driven models. The critical obstacle to model-driven models is the need for a detailed process understanding, in which some parameters are difficult or expensive to obtain. Although hybrid models do not rely entirely on training data and have higher accuracy, they are still more complex than data-driven models. Additionally, hybrid models may partly inherit the disadvantages of data-driven and model-driven models. Hence, in this condition, the data-driven models are still preferable to either of the remaining types.

A more cost-effective approach to modeling would be utilizing data-driven models. However, the reliability of data-driven models critically relies on the quality of historical data and has lower generalization capabilities [64]. High-quality datasets are difficult to acquire and may be costly to procure [40,66]. Because most modern process plants have large control systems (and measurements), the plants provide a mega-large database to train a model [40,66]. Even when the data from normal plant operations is absent, one can still build a database by

Table 1
Concepts, functions, computational loads, advantages and disadvantages of soft sensors.

Model	Model-driven (White-box)	Data-driven (Black box)	Hybrid (Grey box)
Concept	Examine only the internal operations	Examine only the fundamental elements	Partial analysis of internal working and essential elements
Function	Detect implementation issues by analyzing the internal workings and structure of the software	Study only the fundamental aspects of the system and have no understanding of the interior workings	Improve testing coverage by allowing users to focus on all layers of any complicated system by combining all existing white box and black box testing
Computational load	Least exhaustive and time-consuming	Somewhere in between	Potentially most exhaustive and time-consuming
Model complexity	Lower	Higher	Medium
Advantage	• Useful in the development and planning stages • Provides a better understanding of quality parameters	• Easy to develop using model-building software • Knowledge of process background is not required • Reduces the number of hardware sensors in a process plant	• Combines the advantages of black and white-box models • Does not fully rely on training data • Higher accuracy compared to pure black-box models
Disadvantage	• Relatively complex and computationally intensive compared to black-box models • Requires full understanding of process background	• May include redundant data during training • Poor generalization capability • It may not be accurate in new environments	• Increased complexity • May inherit disadvantages of black and white-box models (time consumption etc.)
Reference	[29,41,52,59–62]	[29,52,59–65]	[15,41,52,59–62, 65]

utilizing specially designed experiments that simulate nominal plant operations [40].

Compared with white-box models, data-driven models remain a preferred option as little to no knowledge of process background is required, in addition to reduced resource consumption [29,44]. In the case of sensors, a data-driven soft sensor might serve as an inexpensive replacement for hardware sensors and reduce the number of hardware sensors in a process plant [29,63].

Model training under supervised learning can be succinctly explained by changing Eq. (1) to Eq. **(2)**.

$$Y_{real} = f(X_{train}) \tag{2}$$

Known inputs, X_{train} are used in conjunction with known outputs, Y_{real} during training. Since both the input and output are known, the f can be determined. During testing, f will be used again to estimate outputs using the training dataset, as seen in Eq. (3).

$$Y_{estimate} = f(X_{train}) \tag{3}$$

These equations are not limited to a single input variable but are also applicable to accommodate multiple inputs. Predicting outputs should, in theory, strengthen the conclusion made from the model hypothesis. However, the predictions may become skewed or inaccurate due to the curse of dimensionality [67,68]. One of the effects of this curse would be collinearity, i.e., the existence of a relationship between input variables that may arise due to multiple inputs [68,69]. These correlated inputs will subsequently complicate the learning process. For example, one might predict a person's height based on weight information (a single input). However, given two inputs, e.g., weight in kilograms and weight in grams, either input will be redundant. In this scenario, the conversion rate of kilogram to gram represents collinearity that will cause height prediction to become increasingly poor [68–71].

In some cases, the MLR can suffer from collinearity issues [68,69]. For that reason, one can conduct principal component analysis (PCA) and PLSR using linear regression to minimize dimensionality and avoid collinear input data [40,70]. One can carry out a modified PCA, allowing for a supervised regression method. This approach becomes handy when one uses it according to linear regression; this leads to the PCR model [40,72–74]. Recall that PCA can reduce the dataset dimensionality problem by employing principal components (PCs). Investigating the correlations between multiple inputs using PCA allows the transformation of the correlated inputs into uncorrelated PCs [75,76]. In PCR, the PCs represent the whole dataset and are utilized to displace the inputs while applying linear regression [77]. Another method is called PLSR, which is a supervised learning method that depends on the same theory, in which a set of latent (hidden) variables replaces the original variables to approximately illustrate the correlation in the data [70,73,78]. One can identify the latent variables in PLSR by figuring out the correlation between the inputs, *X,* and output variables, *Y.* Unlike the PLSR, PCA only takes into account input variables *X* to compile the dataset [70,79].

Some researchers have suggested using the nonlinear iterative partial least squares (NIPALS) algorithm to build PLSR models [80]. Another approach is the direct application of a statistically inspired modification to the PLSR method. This second approach becomes known as the statistically inspired modification of the PLSR (i.e., the statistically inspired modification of partial least squares ((SIMPLS)) model [40,81,82]. As previously mentioned, either PCA or PLSR can help resolve the collinear data issue. However, a potential drawback emerges as both methods make the problem simple by assuming the input variables have the same degree of significance for the predictive performance. Back to the previous example of height prediction, considering that genetics and weight were elements for forecasting height, the algorithm would assume both variables to impose the same degree of influence on the model prediction. If the genetic factor contributes more than weight to the height, then the model prediction will become skewed. In this scenario, one would have to devise a way to identify the degree of importance of individual inputs. Concerning this problem, PLSR models have undergone improvement by introducing weights following the degree of importance [83–85]. This improved method is called locally weighted partial least squares regression (LW-PLSR).

As mentioned earlier, PCR, PLSR, and LW-PLSR are linear regression types only applicable to linear relationships between variables. Moreover, a comparative study of these regression models is available in Thien and Yeo [86]. Kernel partial least squares regression (KPLSR) models can be developed by applying kernel functions to PLSR to handle the nonlinearity problem in chemical processes [87,88]. This approach is called the "kernel trick", where the PLS represents a specific case of KPLSR [88]. The KPLSR model utilizes a nonlinear projection function that maps the original process variables into a high-dimensional feature space [33–35]. This results in an approximately linear relationship between the translated and output variables [33–35]. The "kernel trick" may be implemented in LW-PLSR as well. It becomes known as locally weighted kernel partial least squares regression (LW-KPLSR) and allows consideration for weighted and nonlinear datasets. Table S1 in the supplementary materials summarizes the advantages and disadvantages of each discussed regression technique.

4.1. Linear regression and multiple linear regression

The formula described in Eq. (1) is a basic description of simple linear regression, and its true form is as in Eq. (4) from Hope [68] and Ross [89] as follows:

$$Y = C + \beta X + \varepsilon \tag{4}$$

where Y is the response variable or output, C is a constant value or intercept, β is the slope coefficient or weight, X is the predictor (input), and ε is a noise parameter. The constant, C defines the response value when there is zero input, whereas the slope coefficient, β defines the ratio of change in response to input values. Both the constant, C, and the slope coefficient, β are determined by minimizing the squared distances between the data points and the resulting lines [68,89,90]. The equation is not limited to a single input. In the MLR, the formula in Eq. (4) can be expanded to include multiple inputs, as in Eq. (5) [68,71,91].

$$Y = C + \beta 1X1 + \beta 2X2 + \beta 3X3 + \ldots + \beta nXn + \varepsilon \tag{5}$$

The MLR models are parametric if the datasets are of normal distribution [13]. The advantages of MLR models include data visualization, ease of interpretation, and structural simplicity, where the model building is possible without physical knowledge of the phenomena [68,92]. Ayoub [93] used the MLR model to predict the removal percent for one of the most persistent pharmaceuticals in the ecosystem, namely tetracycline, and they found that while MLR can incorporate interactions between variables through interaction terms, it primarily describes linear relationships. However, it is reasonable to think that predictors having little significance for the actual phenomena would cause highly skewed results and prediction performance [94]. A workaround for this problem is to use methods that can somehow recognize the existence of a relationship between input data while providing accurate results. To resolve the issue in the MLR models, one can also use the PCR and PLSR models in conjunction with the former [40,70].

4.2. Principal component analysis and principal component regression

The salient feature of the PCA is the projection of the original input space onto a lower subspace to identify linearly uncorrelated variables. The lower subspace consists of several axes known as PCs [75,76,95]. The PCs emerge due to the orthogonal projections that convert a set of normalized matrix, X of m variables, and n samples into a matrix of independent variables $\widehat{X}$ in the PC subspace, where E denotes a matrix in the residual subspace [40,71]. For more details, they can be found in Zhao et al. [71], and Zhang et al. [96]. The relationship between these variables is as in Eq. (6).

$$X = \widehat{X} + E \tag{6}$$

In Eq. (6), $X \in \Re^{N \times M}$ represents the normalized matrix (original input), $m = (1, 2, 3, \ldots., M)$ represents the number of input variables, $n = (1, 2, 3, \ldots, N)$ represents the number of samples, $\widehat{X}$ represents the dominant information (orthogonal and orthonormal matrices), and E represents the residual matrices [40,71]. Eq. (6) can then be expanded into Eqs. (7) and (8) [71]:

$$\widehat{X} = TP^T \tag{7}$$

$$E = \widetilde{T}\widetilde{P}^T \tag{8}$$

where $T = [t_1, t_2, t_3, \ldots, t_k] \in \Re^{N \times K}$ represents the score matrix (PC matrix or orthogonal matrix or scores), the superscript T denotes the transpose of a vector or matrix, the vector t_1 represents the score (PCs), $P = [p_1, p_2, p_3, \ldots, p_k] \in \Re^{M \times K}$ represents the loading matrix, also known as the principal transform matrix or orthonormal matrix, $k = (1, 2, 3, \ldots, K)$ represents the number of PCs, and the vector p_1 represents the loading vector (principal vector or eigenvector) associated with an eigenvalue λ_1 of the covariance matrix for X in Eq. (6) [40,71].

PCR involves implementing a linear regression model with the output variables and the PCs [40,71,72,74]. A simplified PCR is shown in Eq. (9) as follows [97].

$$Y \approx C + \beta'_1 T_1 + \beta'_2 T_2 + \beta'_3 T_3 + \ldots + \beta'_n T_n + \varepsilon \tag{9}$$

where β' represents the modified terms or weights, and T represents the PCs concerning input variables. The modified β' the term arises due to the reduction of standard errors and bias towards the regression estimates [72].

Without considering output variables, such PCA falls into an unsupervised learning category [17]. After implementing a linear regression model into the PCA, the resulting PCR becomes a supervised method for which the input variables are known [40,72,73]. If the model assumes the given dataset is a gaussian (normal) distribution, then the PCR model is a parametric type. Consequently, it follows the Central Limit Theorem [11,13,90]. The method uses the input dimensions to identify and rank "features" called PCs. Thus, the PCR helps to identify the required number of PCs to sufficiently represent the original dataset. In a way, one can remove the collinearity and reduce dimensionality in the dataset [40,69,76,98,99].

Furthermore, PCA is used in classification to identify and determine the significant variables by breaking down the original inputs into uncorrelated PCs [75,76]. In a practical situation, the PCs do not represent the original inputs 100 % but can only approximately translate a majority of the dataset (i.e., its variance) with a minimal loss of information [76,77]. Note that the larger the number of PCs adopted, the more the PCs can capture the original dataset. However, for data reduction, it is essential to keep the number of PCs required as low as possible, just enough to attain a given threshold variance of the original dataset. Unlike the PCA, the PCR involves the application of the PCs rather than the actual inputs, to perform linear regression to determine the correlation between the input and output [40,71,74]. The method is typically helpful when the number of PCs exceeds the number of actual inputs [94].

Meanwhile, Mouhtady et al. [100] have examined the use of PCA in the field of wastewater from industrial dyes. By utilizing PCA, the adsorption efficiency of graphene (G) oxide-based hydrogels in wastewater dye removal was evaluated. Mouhtady et al. [100] found that PCA can show that the alginate G oxide-based hydrogel with the absence of polyvinyl alcohol gave better tolerance in a basic medium and delivered a higher adsorption capacity. Their study proves that PCA can be effective in describing hydrogel selection along with the influencing factors for pollutant removal from aqueous solutions.

The advantages of using the PCA technique include noise reduction, the removal of collinearity, and data visualization in 2-dimension or 3-dimension [69,74,76]. PCA and its variant PCR can effectively reduce the data dimensionality by determining the main features or critical patterns representing the entire dataset [77]. Still, there is a drawback since the PCR does not consider the output variables when determining the PCs [40,79]. Therefore, another method has evolved to circumvent the drawback, which is the PLSR, which can take into account the output and input variables together [40,83].

4.3. Partial least squares regression

According to Yuan et al. [101], similarities in the input do not correspond to similarities in the output variables. In PCA and PCR, one determines the PCs using only the input variables without considering the outputs, as in Eqs. (6) to (8). One can overcome this drawback by adopting the PLSR method, which uses latent variables to represent the input-output variable datasets. Here, the previously mentioned SIMPLS or NIPALS algorithm can effectively help to identify the latent variables of interest [40,81,82]. Essentially, the PLSR method follows the following equations [70,77,78,102].

$$X = TP^T + E_X \tag{10}$$

$$Y = UQ^T + E_Y \tag{11}$$

whereby superscript T denotes the transpose of a matrix, $X \in \Re^{N \times M}$ represents the normalized input matrix (original input), $m = (1, 2, 3, \ldots, M)$ represents the number of input variables, $n = (1, 2, 3, \ldots, N)$ represents the number of samples, $Y \in \Re^{N \times R}$ represents the normalized output matrix (original output), R represents the number of output variables, $T = [t_1, t_2, t_3, \ldots, t_k] \in \Re^{N \times K}$ represents the score matrix for X (latent variable matrix or orthogonal matrix or scores), $U = [u_1, u_2, u_3, \ldots, u_k] \in \Re^{N \times K}$ represents the score matrix for Y (orthogonal matrix or latent variable matrix or scores), $k = (1, 2, 3, \ldots, K)$ represents the number of latent variables, $P = [p_1, p_2, p_3, \ldots, p_k] \in \Re^{M \times K}$ represents the loading matrix of X (orthonormal matrix), $Q = [q_1, q_2, q_3, \ldots, q_k] \in \Re^{R \times K}$ represents the loading matrix of Y (regression coefficient vector obtained from latent variables concerning output variables), $E_X \in \Re^{N \times M}$ and $E_Y \in \Re^{N \times R}$ represent the residual matrices for X and Y, respectively [83,103–105].

After the decomposition of Eqs. (10) and (11), the linear regression is applied to both T and U terms [106]. The relationship between the latent variables T and U can be depicted using Eq. (12).

$$U = TB + E_1 \tag{12}$$

where $B = diag[b_1, b_2, b_3, \ldots, b_k] \in \Re^{K \times K}$ represents the regression coefficient matrix and E_1 represents the regression residuals [103,105].

Note that all PLSR models fall into the supervised type due to the consideration of outputs during training and the adjustment of some latent variables [73,81,104,107]. PLSR models are parametric because they involve the estimation of parameters from the data. They do not require the dataset to follow a Gaussian or normal distribution, although assuming such distributions can sometimes simplify the analysis. PLSR is robust to various data distributions and is particularly useful in handling multicollinearity and high-dimensional data. The modeling procedure involves the decomposition of the X and Y matrices into loadings, scores, and residual matrices, as in Eqs. (10) and (11) [106]. The method applies regression to determine the score matrices of models [105,106]. The PLSR relies on the same principle as PCR, but the former obtains its latent variables using both the input and output variables [40,70,79]. Like the PCs in PCR, the latent variables can only represent the majority of the variance in the actual inputs. Specifically, the PLSR helps to reduce dimensionality and collinearity. The objective is attainable by maximizing the degree of association (covariance) between Y and the score vector t [70,98,104]. The decomposition process can either use the NIPALS or the SIMPLS algorithm [40,81,82].

In addition, the PLSR model is a multivariate calibration method that can be utilized to evaluate pharmaceutical datasets. According to Naguib and Abdallah [108], the effect of cut-off area (COA) on datasets consists of a training set, a test set, and dosage form sets. In their study, two different sets ranging from 200–325 nm (with COA) and 221–325 nm (without COA) were studied using the PLSR method. Apart from that, PLSR is also known as a linear chemometric method that possesses the finite capability to deal with unexpected noise, and it is recommended to use more elaborate chemometric tools for including COA data, particularly when adding it is a must.

Several advantages of PLSR models include noise reduction, better predictions compared to linear regression models, and the removal of collinearity [40,104,109]. Generally, one would consider PCR to have relatively low prediction capabilities compared to PLSR due to the consideration of both input and output variables of the latter [40,110]. However, since both methods build their foundation on the MLR approach, PCR and PLSR models can, in theory, attain similar prediction capabilities [40,70,111].

4.4. Locally weighted partial least squares regression

LW-PLSR is an extension of PLSR whereby query, x_q and historical, x_i samples are weighted according to importance using a similarity matrix [83,84,104,112]. Then, the LW-PLSR model is built locally based on the weights, hence the name, locally weighted [4,113]. In soft sensors, LW-PLSR is a well-known just-in-time (JIT) modeling technique based on PLSR and is an important aspect of inferential control [83,84,104,112,114]. To do so, the matrices of the input $X \in \Re^{N \times M}$ and output $Y \in \Re^{N \times R}$ are stored in a database [83,84,112,115]. In a dataset $\{x_i, y_i\}_{i=1}^{N}$ of a sample size of $n = (1, 2, 3, .., N)$, the input and output can be denoted by Eqs. (13) and (14).

$$X_i = [x_{i1}, x_{i2}, x_{i3}, \ldots, x_{iM}]^T \tag{13}$$

$$Y_i = [y_{i1}, y_{i2}, y_{i3}, \ldots, y_{iR}]^T \tag{14}$$

whereby $m = (1, 2, 3, \ldots, M)$ is the number of input variables, $R = (1, 2, 3, \ldots, R)$ is the number of output variables, and superscript T is the transpose of a vector or matrix [83,84,104,112]. Whenever an output estimation is required, the similarity of the query x_q and historical x_i data are then compared before constructing the LW-PLSR model [83–85,112]. The weighting process is done with the similarity matrix $\Omega \in \Re^{N \times N}$ shown in Eq. (15) [83–85,112,115].

$$\Omega = diag(\omega_1, \omega_2, \omega_3, \ldots, \omega_N) \tag{15}$$

The JIT methods involve building a series of local models online. Note that the methods are not in the category that uses a regression technique [116]. LW-PLSR is known as a JIT modeling technique due to the integration of a PLSR model into the JIT framework. Therefore, the LW-PLSR models tend to be superior to the PLSR, as the former models can analyze the similarities between the inputs. Also, the former can take advantage of query samples using techniques such as distance-based (JIT approach) or time-based (temporal) methods [83,116–119].

Furthermore, Qi et al. [118] demonstrated a study to estimate the approximation of the analysis sample for near-infrared spectroscopy (NIRS). In their study, the LW-PLSR acquired superior estimation results over conventional methods such as PCR and PLSR. LW-PLSR is examined to estimate the details of volatility, moisture, fixed carbon, and sawdust ash through the utilization of NIRS. LW-PLSR methods assign assorted weights based on the level of equivalence of the predicted specimens as well as the modeling specimens, which brings an obvious enhancement to prediction prototypes in terms of accuracy and robustness.

The LW-PLSR method can deal with time-varying, nonlinear, multiphase, and multimode process characteristics [84,117,119–121]. Hence, the abilities of LW-PLSR are often beneficial for process monitoring and control equipment such as soft sensors [84,117,119,122]. Spatial LW-PLSR models typically use distance to analyze similarities based on the spatial relevance of the samples. So, a smaller spatial distance between samples and predictions indicates a higher significance and is therefore assigned a higher weight [119]. If the LW-PLSR model uses a sampling interval distance, the model fits into a temporal LW-PLSR category [119].

4.5. Kernel partial least squares regression

According to Cover's theorem, a kernel function converts a nonlinear dataset (X, Y) into an approximately linear dataset by utilizing high-dimensional nonlinear mapping [33–35]. A normalized input matrix $X = [x_1, x_2, x_3, \ldots, x_N]^T \in \Re^{N \times M}$ (original input) is mapped into a high-dimensional feature space using a nonlinear projection function ϕ through the following Eq. (16).

$$x_i \in \Re^M \rightarrow \phi(x_i) \in R^\Theta \tag{16}$$

whereby the superscript T denotes the transpose of a matrix, $n = (1, 2, 3, ..., N)$ represents the number of samples, $m = (1, 2, 3, ..., M)$ represents the number of input variables, and the size of the dimension, Θ is arbitrary or possibly infinite [33,34,123]. The original input is then translated into the feature matrix denoted by Φ before being subjected to the mean centring shown in Eqs. (17) to (20):

$$\Phi = [\phi(x_1), \phi(x_2), \phi(x_3), ..., \phi(x_N)]^T \in \Re^{N\times\Theta} \tag{17}$$

$$\overline{\phi}(x_i) = \phi(x_i) - \overline{\phi} \tag{18}$$

$$\overline{\phi} = \frac{1}{N}\sum\nolimits_{i=1}^{N} \phi(x_i) = \frac{1}{N}\Phi^T 1_N \tag{19}$$

$$\overline{\Phi} = [\overline{\phi}(x_1), \overline{\phi}(x_2), \overline{\phi}(x_3), ..., \overline{\phi}(x_N)]^T = \Phi - [\overline{\phi}, \overline{\phi}, \overline{\phi}, ..., \overline{\phi}]^T = \Phi - \frac{1}{N}1_N 1_N^T \Phi \tag{20}$$

where $\overline{\Phi}$ represents the mean of the translated input matrix Φ in the feature space and 1_N represents a vector of ones [34,123]. The approximately linear dataset can be denoted by $(\overline{\Phi}, Y)$ shown in Eqs. (21) and (22):

$$\overline{\Phi} = TP^T + E_\Phi \tag{21}$$

$$Y = UQ^T + E_Y \tag{22}$$

whereby $T = [t_1, t_2, t_3, ..., t_k] \in \Re^{N\times K}$ represents the score matrix for X (score matrix or latent variable matrix or orthogonal matrix), $P = [p_1, p_2, p_3, ..., p_k] \in \Re^{M\times K}$ represents the loading matrix of $\overline{\Phi}$ (orthonormal matrix), $E_\Phi \in \Re^{N\times M}$ represents the residual matrix of $\overline{\Phi}$, $E_Y \in \Re^{N\times R}$ represents the residual matrices for Y, and $k = (1, 2, 3, ..., K)$ represents the number of latent variables [33,34,123].

KPLSR is classified as an extension of PLSR, while PLSR is known as a special case of KPLSR [88,124,125]. This is seen when comparing Eqs. (10) and (11) with Eqs. (21) and (22). Implementing a "kernel trick" on conventional PLSR converts a nonlinear dataset into an approximately linear dataset [88]. PLSR is then conducted on the translated data concerning the output data. Besides, different types of kernel functions have been introduced, and the common kernel functions include the sigmoidal kernel, polynomial kernel, and gaussian kernel [123]. Detailed explanations of the types of kernels available are included in Sections 5 and 6.

Furthermore, KPLS possesses the advantage that it can deal with distinct nonlinearities because of its competency in employing different kernel functions. In the research studied by Zhao et al. [88], a coarse sorting method for discrepancy removal by KPLSR is recommended for registration subjected to image sensing registration, which falls under the kernel principal component analysis (KPCA) view of graph matching as well as the corresponding properties of features subjected to KPLSR. Such a method is certainly without accidental evaluation and can store a lot of qualities that are beneficial to image input. Its legitimacy is to decrease the discrepancy resulting from cognate descriptors, aided by the dimensional relationship within the features of matching. By analyzing the collinearity of KPLSR features, it is possible to determine the frequency of outliers in the current matching as well as a reasonable tolerance.

4.6. *Locally weighted kernel partial least squares regression*

The LW-KPLSR model allows weights in addition to the ability to process nonlinearity [115]. Similar to LW-PLSR, it is also an example of a JIT learning method due to the implementation of a JIT framework [84,116]. For a dataset and query of sample size $n = (1, 2, 3, .., N)$, LW-KPLSR may be conducted using the following procedure from Eqs. (23) to (26) [115].

$$x_i = [x_{i1}, x_{i2}, x_{i3}, ..., x_{iM}]^T \tag{23}$$

$$y_i = [y_{i1}, y_{i2}, y_{i3}, ..., y_{iR}]^T \tag{24}$$

$$x_q = [x_{i1}, x_{i2}, x_{i3}, ..., x_{iM}]^T \tag{25}$$

$$y_q = [y_{i1}, y_{i2}, y_{i3}, ..., y_{iR}]^T \tag{26}$$

Step 1: Obtain the kernel matrices for the input variables z and queries z_q as shown in Eqs. (27) and (28). This results in nonlinear mapping into a higher dimensional feature space.

$$Z = \Phi\Phi^T \in \Re^{N\times N} \tag{27}$$

$$Z_q = \Phi_q {\Phi_q}^T \in \Re^{N\times N} \tag{28}$$

Step 2: Allow the kernel matrices for the input and queries to possess zero means using Eqs. (29) and (30), whereby 1_N represents a vector of ones, and this is known as mean centering.

$$\overline{Z} = \left(I_N - \frac{1}{N}1_N 1_N^T\right) Z \left(I_N - \frac{1}{N}1_N 1_N^T\right) \tag{29}$$

$$\overline{Z}_q = \left(Z_q - \frac{1}{N}1_N 1_N^T Z\right)\left(I_N - \frac{1}{N}1_N 1_N^T\right) \tag{30}$$

Step 3: Conduct dual kernel partial least squares discrimination to obtain the regression coefficient (scaled direction of projection concerning kernel matrix Z), $B = diag[b_1, b_2, b_3, ..., b_k] \in \Re^{K\times K}$ using Eq. (31). Then, Eqs. (32) to (33) are used for the rescaling of the inputs and queries.

$$B = YY'Z\beta\text{whereby}\beta = \frac{\beta}{\|\beta\|} \tag{31}$$

$$X_q = Z_q B \tag{32}$$

$$X = ZB \tag{33}$$

Despite the perceived complexity of the LW-KPLSR technique, it is relatively simple when one understands the basics of LW-PLSR. As seen above, the method only requires three additional steps before proceeding with the steps as in the LW-PLSR. The implementation of vast amounts of existing kernel functions, such as gaussian or polynomial kernels, has led to numerous breakthroughs in variable prediction or predictive performance.

Moreover, Zhang et al. [122] proposed a novel LW-KPLSR according to sparse nonlinear features for nonlinear time-varying processes. It was shown that LW-KPLSR improves the locally weighted framework's ability to handle strong nonlinearity by incorporating the nonlinear feature. Furthermore, LW-KPLSR models are built using sparse kernel feature characterization words that represent nonlinear dependency inside the training samples and queries.

5. Kernel functions

As mentioned earlier, the mapping of nonlinear datasets into a high-dimensional feature space can be carried out by applying kernel functions [33–35]. A kernel function is a mathematical tool known for its nonlinear projection capabilities, which create simulated linear behavior in the feature space [33–35]. It translates input data into a higher-dimensional space, enabling a linear algorithm to exhibit nonlinear behavior in the original input space [37]. This approach is powerful because it shows the relationships between nonlinearity, linearity, and algorithms through simple dot products between vectors [37].

Similarly, with appropriate strategies, linear algorithms might be adaptable to consider nonlinearity, i.e., conversion to nonlinear algorithms. In other words, a set of data that forms a nonlinear line within a lower dimension can be transformed into a linear 2-dimensional plane within a higher-dimensional space. Within the limits of the feature space, these nonlinear algorithms exhibit an approximate equivalence to their respective linear originals [33–35]. Recently, it was also found that several kernel functions have received wider research attention for engineering applications.

6. Reviewing kernel functions and their applications

Pilario et al. [45] revealed that KPCA has been widely used for feature extraction in process monitoring. Meanwhile, Liu and Xie [126] also noted that many researchers focus more on kernel functions for classification than regression techniques. It can be seen in Table S2 in the supplementary materials that there are different types of kernel functions, and many of them possess unique characteristics that allow superior performance over others in specific scenarios. Many research papers adopted kernel functions specified in other works, resulting in a lack of comparative studies on different kernel functions within their works. After the kernel functions map the data or variables into a higher dimensional space, a data-driven or ML model called "soft sensor" implies some input variables (secondary), which serve as predictors to predict desired primary output variables.

Table S3 in the supplementary materials summarizes recent applications of kernel functions in ML and data science. In Table S3, each existing study that utilized the kernel function in the ML model was classifiable into regression and classification techniques to understand their applications. Notice that the gaussian kernel has received wide applications in various areas. Some examples of gaussian kernel applications include wastewater treatment [127], data analysis [128–133], medical diagnosis, and cancer detection [134], heat transfer [135], industrial microbiology and biotechnology [136], hydrology and flood frequency analysis [137], digital imaging [138], construction [139], metamodeling techniques [140], crude oil analysis [141] and gas-sweetening processes [142].

Moreover, it can also be seen that the gaussian kernel has been widely applied with various ML models for data analysis. They are SVM on time series forecasting, gaussian process regression (GPR) on compositional kernel learning, least squares support vector regression (LSSVR) on missing data, recursive PCA and local and global randomized PCA on the Tennessee Eastman process, and PCA on fault diagnosis. For instance, Peng and Nagata [128] applied SVM with the gaussian kernel function to conduct the time series prediction of COVID-19 cases, and this integrated model provided the best performance. Besides, Jin [129] presented compositional kernel learning for GPR. They proved that its performance can be enhanced significantly by applying a proper kernel function with the increment of the training data. Mesquita et al. [130] also reported that LSSVR with the gaussian kernel can estimate the missing components.

Furthermore, Shang et al. [131] proposed a recursive PCA to carry out the monitoring of the Tennessee Eastman process more effectively. They discovered the proposed method has a lower computational cost, fits the nonlinear time-varying characteristics of industrial processes, and is also able to identify three types of process faults, which include step, random variation, and sticking. Moreover, Wu et al. [132] developed a local and global randomized PCA by mapping the input space onto a feature space, and they used random Fourier features to reveal nonlinear patterns, while their proposed method can minimize computational and storage costs.

Also, Wang et al. [133] employed an efficient nonlinear fault diagnosis method by combining the contribution plots with KPCA-based fault diagnosis. Their established method outperforms in handling computational complexity and extensibility. Table S3 in the supplementary materials focused on the kernel functions in areas such as mathematical calculations or electronic signals without incorporating ML models. From Table S3, notice that most recent studies on kernel functions used classification rather than regression. Also, most kernel-based data-driven regression models are support vector regression (SVR) and GPR [45,126].

Moreover, from Table S3, notice that the usage of kernel functions in regression models is mostly for process monitoring, as nonlinear datasets are extremely common in process industries [87,143]. Also, it was found that only a minority of the research work in Table S3 investigated the performance of data-driven models with multiple kernel functions. Consequently, the researchers involved might have ignored some possible improvements in their research due to this practice. This is because the multitude of available kernel functions may allow a designer to improve or produce unique ideas in the field of process monitoring by understanding the advantages, characteristics, and limitations of the kernel functions used. To address this research gap, in this review study, Table S4 in the supplementary materials depicts a summary of the characteristics, advantages, and disadvantages of kernel functions in several applications. Notice that most regression-based works on kernel functions utilized either SVR or GPR rather than the MLR, PCR, PLSR, and LW-PLSR techniques.

7. Kernel functions: characteristics, advantages, and limitations

Notably, the kernel functions possess unique characteristics, advantages, and limitations in several applications, as summarized in Table S4 of the supplementary materials. By utilizing kernel functions in the soft sensor application in chemical processes, nonlinear data can be transformed into a higher-dimensional space where linear algorithms can be effectively applied. This approach allows for a curved decision boundary in a lower-dimensional space, corresponding to a linear boundary in a higher-dimensional space. These nonlinear algorithms with kernel functions are equivalent to their linear counterparts, but no explicit feature space transformation is required [144].

The kernel functions can be grouped into two categories: global kernel functions and local kernel functions. The far-from-one data points have an impact on the performances of global kernel functions, such as linear and polynomial models. [145]. The linear kernel is classified as the most common kind of kernel function, which is nonstationary and relies on the specific locations of the inputs [129,145]. Algorithms that use a linear kernel are often comparable to their non-kernel counterparts. For instance, KPCA with a linear kernel is equivalent to standard PCA [37]. The linear kernel can be combined with periodic kernels to handle periodic structures within the data. This kernel is ideal for problems where linear separation of the data is feasible [146]. Nevertheless, it is unable to handle nonlinear relationships between inputs and outputs effectively [137]. Besides, a polynomial kernel exists as a nonstationary kernel that can adapt to problems involving normalized training data. Different from the linear kernel, the polynomial kernel can handle nonlinearity between inputs and outputs [137]. It offers superior extrapolation capability with lower-order polynomial kernels, though its interpolation ability is limited [146].

In contrast, local kernel functions such as the gaussian kernel, exponential kernel, Laplacian kernel, ANOVA kernel, sigmoid kernel, rational quadratic kernel, multiquadric kernel, inverse multiquadric kernel, circular kernel, spherical kernel, and wave kernel, alter the kernel value based on the data points that are close to each other [145]. The gaussian kernel, exponential kernel, Laplacian kernel, and ANOVA kernel are known as radial basis function (RBF) kernels that are isotropic and stationary [147]. When applied appropriately, these kernel functions exhibit good interpolation ability, and the variables can be mapped onto infinitely large dimensions [148,149]. However, these kernel functions demonstrate limited extrapolation capability and lose interpolation ability for large kernel widths [45].

Apart from that, a sigmoid kernel is a conditionally positive definite kernel. It is also known as a multi-layer perception kernel and is equivalent to a gaussian kernel in a certain scenario [37,146,150]. However, because it is conditionally positive definite, it can lead to incorrect approximations and has a lower interpolation capability than the gaussian kernel [137,145]. Additionally, RBF kernels do not include isotropic stationary kernels such as rational quadratic kernels, multiquadric kernels, inverse multiquadric kernels, circular kernels, spherical kernels, and wave kernels. The rational quadratic kernel has good interpolation ability, can serve as an alternative to the gaussian kernel, and reduces the computational load associated with the RBF kernel [151,152]. Yet, it can be altered by a multiquadric kernel, which, despite being a non-positive definite kernel, is commonly used for scattered data and smoothing [152–154]. Like the rational quadratic kernel, the multiquadric kernel also has good interpolation ability, but it fails to perform well under noisy conditions [155]. Different from the multiquadric kernel, the inverse multiquadric kernel can map the variables onto infinitely large dimensions but fails to detect non-convergence [37,156].

The circular kernel and spherical kernel, which are conditionally positive definite, and compactly supported, are commonly applied in geostatistical analysis. The circular kernel performs better on images with symmetric and round shapes [157,158], while the spherical kernel preserves translation-invariant properties, maintains asymmetry, and preserves permutation invariance [159]. However, both of these kernels are limited to one to three dimensions and provide poor performance for nonlinear regression models [158]. In addition, the wave kernel has a symmetric positive semi-definite characteristic [152]. This characteristic makes it suitable for wave propagation studies and provides good performance in Laplacian matrix studies in graph theory. However, its application is limited due to its complexity [147,152,160]. The characteristics, advantages, and limitations of the remaining kernel functions can be referred to in Table S4 in the supplementary materials.

8. Future research challenges and direction

A desirable and dependable soft-sensing model needs to enable an alert reference for proactive action in advance of any abnormal operation and give clear descriptions of dynamics. For instance, the soft-sensing models have been applied by Farzin et al. [161] to predict the biogas yield from anaerobic digesters at a wastewater treatment plant (WWTP). Such a reliable soft-sensing model requirement necessitates the design of an efficient and accurate model. As reported by Obaideen et al. [162], WWTP also contributes to achieving 11 out of 17 sustainable development goals. A WWTP is a complex system with a combination of interconnected physical, chemical, and biological processes and reactions that are operated to treat industrial wastewater and eliminate pollutants [163]. As the WWTP is challenging to conduct analysis and optimization on its control with the traditional mathematical model due to its nonlinear behavior, the model, along with the application of the kernel functions, has the potential to provide a helpful method for assessing the environmental safety of water [164–170]. For instance, the ammonia in WWTP is one of the main sources of nitrogen pollutants from agricultural, sewage, and industrial discharges, posing a considerable threat to water ecosystems. The substantial release of ammonia into water streams can result in eutrophication, which is harmful to aquatic life .

Based on the cumulative current knowledge, the present review suggests adopting pilot-scale experiment data to develop the model for optimization and regeneration. By conducting optimization and regeneration studies, one can determine the optimum regeneration time that can assist in reducing the regeneration process cost and power savings on the amount of electricity used. Although previous studies have demonstrated that different models have broad applications, there remains a minimal study using least-squares-based regression models in WWTP processes. The process of fault detection, process monitoring of water quality parameters, and improvement of treatment outcomes through real-time chemical process monitoring in WWTP are made possible by these least-squares-based regression models. Thus, least-square regression models can be effective alternatives in WWTP processes.

9. Conclusions

This review study presents the latest studies on kernel functions, their application categories, and their integration with data-driven models. Unlike other review studies that are currently in existence, this study compares the characteristics, limitations, and benefits inherent in many kernel functions. Furthermore, this study also critically reviews data-driven models for regression and classification tasks, including the pros and cons. Likewise, this review establishes the cutting edge of different kernel functions that were used in data-driven models for regression and classification. Despite that, notice that the current studies on kernel functions are often used for the classification task as compared to regression. Moreover, the most commonly applied kernel function in various applications is found to be the gaussian kernel.

In addition, this review study serves as a guide for researchers towards choosing suitable kernel functions in real-world applications, impacting the practical implementation of these techniques. Nonetheless, several researchers have attempted to integrate kernel functions with nonlinear models such as the SVM, LSSVR, and GPR, to demonstrate the efforts to widen the applications beyond classification tasks. By comparison, it was found that linear models like PCA and PLSR have received less attention for kernel functions. This finding impacts the researchers' decisions to disclose the kernel function integration with linear models to study their applicability. Apart from that, one of the identified insights into how those studies on kernel functions did not consider the online or adaptive mode. This finding provides an insight into how future research efforts can contribute to more dynamic and real-time uses of kernel functions. Hence, future works should also emphasize integrating different kernel functions with adaptive data-driven models in different industrial applications such as WWTP, ML, manufacturing automation, and medical usage.

CRediT authorship contribution statement

Joyce Chen Yen Ngu: Writing – original draft, Investigation, Data curation. **Wan Sieng Yeo:** Writing – review & editing, Visualization, Supervision, Funding acquisition. **Teck Fu Thien:** Writing – original draft, Data curation. **Jobrun Nandong:** Writing – review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to thank Curtin Malaysia Research Institute Collaborative Research Scheme (CMRI-CRS) 2021 (CMRI 006032), Curtin University Malaysia, and SEGi University Malaysia for providing financial support for this project. Moreover, the authors also thank the contribution of Dr. Chan Mieow Kee from SEGi University Malaysia in this CMRI-CRS 2021 via vote number SEGiIRF/2022-Q1/FoEBEIT/003.

Authorship statement

Authors make substantial contributions to the development of ideas, their implementation, the conception of experiments, data acquisition, analysis, or interpretation. All authors give their final approval of the manuscript version to be submitted and any revised version of it.

Computer code availability

This is a review study and no computer code for data is generated or analyzed during this study, so no computer code is included in this final manuscript.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.asoc.2024.111975.

References

[1] S. Vieira, W.H.L. Pinaya, A. Mechelli, Introduction to machine learning. in: Machine learning, Elsevier, 2020, pp. 1–20, https://doi.org/10.1016/B978-0-12-815739-8.00001-8.

[2] W.S. Yeo, W.J. Lau, Predicting the whiteness index of cotton fabric with a least squares model, Cellulose 28 (2021) 8841–8854, https://doi.org/10.1007/s10570-021-04096-y.

[3] J.C.Y. Ngu, M.K. Chan, W.S. Yeo, J. Nandong, The application of machine learning in nanoparticle treated water: a review, MATEC Web Conf. 377 (2023) 01009, https://doi.org/10.1051/matecconf/202337701009.

[4] M.N. Pervez, W.S. Yeo, M. Mishu, M. Rahman, M.E. Talukder, H. Roy, M.S. Islam, Y. Zhao, Y. Cai, G.K. Stylios, Electrospun nanofiber membrane diameter prediction using a combined response surface methodology and machine learning approach, Sci. Rep. 13 (2023) 1–14, https://doi.org/10.1038/s41598-023-36431-7.

[5] M.N. Pervez, W.S. Yeo, F. Shafiq, M.M. Jilani, Z. Sarwar, M. Riza, L. Lin, X. Xiong, V. Naddeo, Y. Cai, Sustainable fashion: design of the experiment assisted machine learning for the environmental-friendly resin finishing of cotton fabric, Heliyon 9 (2023), https://doi.org/10.1016/j.heliyon.2023.e12883.

[6] M.N. Pervez, W.S. Yeo, L. Lin, X. Xiong, V. Naddeo, Y. Cai, Optimization and prediction of the cotton fabric dyeing process using Taguchi design-integrated machine learning approach, Sci. Rep. 13 (2023) 12363, https://doi.org/10.1038/s41598-023-39528-1.

[7] W.S. Yeo, M.K. Chan, N.A. Bukhari, Prediction of glucose concentration hydrolysed from oil palm trunks using a PLSR-based model, in: P. Vasant, I. Zelinka, G.-W. Weber (Eds.), Intelligent Computing and Optimization, Springer International Publishing, Cham, 2022, pp. 927–937, https://doi.org/10.1007/978-3-030-93247-3_88.

[8] G. Ye, J. Wan, Z. Deng, Y. Wang, J. Chen, B. Zhu, S. Ji, Prediction of effluent total nitrogen and energy consumption in wastewater treatment plants: Bayesian optimization machine learning methods, Bioresour. Technol. 395 (2024) 130361, https://doi.org/10.1016/j.biortech.2024.130361.

[9] J.C.Y. Ngu, W.S. Yeo, M.K. Chan, J. Nandong, Least squares support vector regression-based modeling of ammonia oxidation using immobilized nanoFeCu, J. Water Process Eng. 64 (2024) 105695, https://doi.org/10.1016/j.jwpe.2024.105695.

[10] M.K. Chan, W.S. Yeo, J.C.Y. Ngu, P.Y. Lee, J. Nandong, N.A. Sharani, A.M. Aris, K. B. Chin, Enhancing sustainability in sewage treatment: A least squares support vector regression-based modeling approach for optimizing regeneration conditions of iFeCu, J. Water Process Eng. 64 (2024) 105694, https://doi.org/10.1016/j.jwpe.2024.105694.

[11] A. King, R. Eckersley, Inferential statistics I: basic concepts, Stat. Biomed. Eng. Sci. (2019) 71–90.

[12] G.M. Beuren, M.J. Anzanello, Variable selection using statistical non-parametric tests for classifying production batches into multiple classes, Chemom. Intell. Lab. Syst. 193 (2019) 103830, https://doi.org/10.1016/j.chemolab.2019.103830.

[13] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, Ind. Eng. Chem. Res. 52 (2013) 3543–3562, https://doi.org/10.1021/ie302069q.

[14] S.W. Scheff, Fundamental statistical principles for the neurobiologist: A survival guide, Academic Press, 2016, https://doi.org/10.1016/B978-0-12-804753-8.00008-7.

[15] H. Haimi, M. Mulas, F. Corona, R. Vahala, Data-derived soft-sensors for biological wastewater treatment plants: An overview, Environ. Model. Softw. 47 (2013) 88–107.

[16] A. Alnuaimi, T. Albaldawi, An overview of machine learning classification techniques, BIO Web Conf. 97 (2024) 00133, https://doi.org/10.1051/bioconf/20249700133.

[17] C. McCue, Identification, Characterization, and Modeling. in: Data Mining and Predictive Analysis, Butterworth-Heinemann, Boston, 2015, pp. 137–155.

[18] H. He, D. Han, J. Dezert, Disagreement based semi-supervised learning approaches with belief functions, Knowl. -Based Syst. 193 (2020) 105426, https://doi.org/10.1016/j.knosys.2019.105426.

[19] J. García, D. Shafie, Teaching a humanoid robot to walk faster through Safe Reinforcement Learning, Eng. Appl. Artif. Intell. 88 (2020) 103360, https://doi.org/10.1016/j.engappai.2019.103360.

[20] C.-H. Pi, K.-C. Hu, S. Cheng, I.-C. Wu, Low-level autonomous control and tracking of quadrotor using reinforcement learning, Control Eng. Pract. 95 (2020) 104222, https://doi.org/10.1016/j.conengprac.2019.104222.

[21] H. Hosseini, M. Shahbazian, M.A. Takassi, The design of robust soft sensor using ANFIS network, Int. J. Instrum. Technol. 2 (2014) 9–16, https://doi.org/10.12691/jit-2-1-3.

[22] K. Fujiwara, M. Kano, S. Hasebe, A. Takinami, Soft-sensor development using correlation-based just-in-time modeling, AIChE J. 55 (2009) 1754–1765, https://doi.org/10.1002/aic.11791.

[23] C. Shang, F. Yang, D. Huang, W. Lyu, Data-driven soft sensor development based on deep learning technique, J. Process Control 24 (2014) 223–233, https://doi.org/10.1016/j.jprocont.2014.01.012.

[24] W.S. Yeo, A. Saptoro, Introduction of LSSVR for the prediction of the yellowness index (-xx), Int. J. Comput. Digit. Syst. 14 (2023) 1, https://doi.org/10.12785/ijcds/150107.

[25] Y. Meng, G. Lu, K. Zheng, Z. Tang, X. Wang, Research of methods for extracting principal components responding to sucrose supersaturation based soft sensors in cane sugar process, Sens. Transducers 162 (2013) 221–226.

[26] D. Liu, Y. Wang, C. Liu, X. Yuan, C. Yang, W. Gui, Data mode related interpretable transformer network for predictive modeling and key sample analysis in industrial processes, IEEE Trans. Ind. Inform. (2022), https://doi.org/10.1109/TII.2022.3227731.

[27] C. Liu, K. Wang, Y. Wang, X. Yuan, Learning deep multimanifold structure feature representation for quality prediction with an industrial application, IEEE Trans. Ind. Inform. 18 (2021) 5849–5858, https://doi.org/10.1109/TII.2021.3130411.

[28] M.T. Tham, G.A. Montague, A.J. Morris, P.A. Lant, Soft-sensors for process estimation and inferential control, J. Process Control 1 (1991) 3–14, https://doi.org/10.1016/0959-1524(91)87002-F.

[29] S. Sedghi, R. Tan, B. Huang, Data analytics approach for online produced fluid flow rate estimation in SAGD process, Comput. Chem. Eng. 136 (2020) 106766, https://doi.org/10.1016/j.compchemeng.2020.106766.

[30] W. Shao, Z. Ge, Z. Song, K. Wang, Nonlinear industrial soft sensor development based on semi-supervised probabilistic mixture of extreme learning machines, Control Eng. Pract. 91 (2019) 104098, https://doi.org/10.1016/j.conengprac.2019.07.016.

[31] R. Srinivasan, D. Wang, J. Liu, Data-driven soft sensor approach for quality prediction in a refining process, IEEE Trans. Ind. Inform. 6 (2010) 11–17, https://doi.org/10.1109/TII.2009.2025124.

[32] D. Wang, J. Liu, R. Srinivasan, Data-driven soft sensor approach for quality prediction in a refining process, IEEE Trans. Ind. Inform. 6 (2009) 11–17, https://doi.org/10.1109/TII.2009.2025124.

[33] Y. Gao, X. Kong, C. Hu, Z. Zhang, H. Li, La Hou, Multivariate data modeling using modified kernel partial least squares, Chem. Eng. Res. Des. 94 (2015) 466–474, https://doi.org/10.1016/j.cherd.2014.09.004.

[34] J. Jiao, N. Zhao, G. Wang, S. Yin, A nonlinear quality-related fault detection approach based on modified kernel partial least squares, ISA Trans. 66 (2017) 275–283, https://doi.org/10.1016/j.isatra.2016.10.015.

[35] R. Rosipal, L. Trejo, Kernel partial least squares regression in reproducing kernel hilbert space, J. Mach. Learn. Res. 2 (2001) 97–123, https://doi.org/10.1162/15324430260185556.

[36] W. Huang, L. Wang, J. Zhu, L. Dong, H. Hu, H. Yao, L. Wang, Z. Lin, Application of machine learning in prediction of Pb2+ adsorption of biochar prepared by tube furnace and fluidized bed, Environ. Sci. Pollut. Res. 31 (2024) 27286–27303, https://doi.org/10.1007/s11356-024-32951-5.

[37] C. Souza, Kernel functions for machine learning applications, Creat. Commons Attrib. -noncommercial-Share alike (2010) 1. -1.

[38] K. Wang, C. Yan, X. Yuan, Y. Wang, C. Liu, A reduced nonstationary discrete convolution kernel for multimode process monitoring, Int. J. Mach. Learn. Cybern. 13 (2022) 3711–3725, https://doi.org/10.1007/s13042-022-01621-8.

[39] Y. Xu, M. Jia, Z. Mao, H. Li, A novel kernel dynamic inner slow feature analysis method for dynamic nonlinear process concurrent monitoring of operating point deviations and process dynamics anomalies, J. Process Control 110 (2022) 59–75, https://doi.org/10.1016/j.jprocont.2021.12.005.

[40] D. Slišković, R. Grbić, Ž. HocenskiD. Slišković, R. Grbić, Ž. Hocenski, Metode modeliranja na pogonskim podacima za razvoj soft-senzora, Automatika: časopis za automatiku, mjerenje, elektroniku, računarstvo i komunikacije, 52 (2011) 306-318. https://doi.org/10.1080/00051144.2011.11828430. .

[41] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Comput. Chem. Eng. 33 (2009) 795–814.

[42] W.S. Yeo, A. Saptoro, P. Kumar, Missing data treatment for locally weighted partial least square-based modelling: A comparative study, Asia-Pac. J. Chem. Eng. 15 (2020) 1–13, https://doi.org/10.1002/apj.2422.

[43] V.M. Mickel, W.S. Yeo, A. Saptoro, Evaluating the performance of newly integrated model in nonlinear chemical process against missing measurements, Chem. Prod. Process Model. 14 (2019) 1–20, https://doi.org/10.1515/cppm-2018-0066.

[44] X. Yuan, C. Ou, Y. Wang, C. Yang, W. Gui, A novel semi-supervised pre-training strategy for deep networks and its application for quality variable prediction in industrial processes, Chem. Eng. Sci. 217 (2020) 115509, https://doi.org/10.1016/j.ces.2020.115509.

[45] K.E. Pilario, M. Shafiee, Y. Cao, L. Lao, S.-H. Yang, A review of kernel methods for feature extraction in nonlinear process monitoring, Processes 8 (2019) 24, https://doi.org/10.3390/pr8010024.

[46] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans. Electron. Comput. (1965) 326–334, https://doi.org/10.1109/PGEC.1965.264137.

[47] X. Dong, Y.-L. Bai, W.-D. Wan, Kernel functions embed into the autoencoder to identify the sparse models of nonlinear dynamics, Commun. Nonlinear Sci. Numer. Simul. 131 (2024) 107869, https://doi.org/10.1016/j.cnsns.2024.107869.

[48] R. Fateh, H. Oualla, E.-s Azougaghe, A. Darif, A. Boumezzough, S. Safi, M. Pouliquen, M. Frikel, Machine learning based system identification with binary output data using kernel methods, J. Telecommun. Inf. Technol. 1 (2024) 17–25, https://doi.org/10.26636/jtit.2024.1.1430.

[49] S. Khatibisepehr, B. Huang, S. Khare, Design of inferential sensors in the process industry: a review of Bayesian methods, J. Process Control 23 (2013) 1575–1596, https://doi.org/10.1016/j.jprocont.2013.05.007.

[50] A. Martynenko, N. Misra, Machine learning in drying, Dry. Technol. 38 (2020) 596–609, https://doi.org/10.1080/07373937.2019.1690502.

[51] T. Melo, N. Brunet, D. Pequeno, J. Silva, J.S. Rocha Neto, Decoupling control and soft sensor design for an experimental platform, : PID Control Ind. Process. (2018) 167–186, https://doi.org/10.5772/intechopen.75708.

[52] C. Abeykoon, Design and applications of soft sensors in polymer processing: a review, IEEE Sens. J. 19 (2019) 2801–2813, https://doi.org/10.1109/JSEN.2018.2885609.

[53] C. Abeykoon, Soft sensing of melt temperature in polymer extrusion, in: 2016. European Control Conference (ECC), IEEE, 2016, pp. 340–345, https://doi.org/10.1109/ECC.2016.7810308.

[54] M. Kano, K. Fujiwara, Virtual sensing technology in process industries: trends and challenges revealed by recent industrial applications, J. Chem. Eng. Jpn. 46 (2013) 1–17, https://doi.org/10.1252/jcej.12we167.

[55] K. Ookita, Operation and quality control for chemical plants by soft-sensors, CICSJ Bull. 24 (2006) 31–33.

[56] W.S. Yeo, A. Saptoro, P. Kumar, Adaptive soft sensor development for non-Gaussian and nonlinear processes, Ind. Eng. Chem. Res. 58 (2019) 20680–20691, https://doi.org/10.1021/acs.iecr.9b03821.

[57] J.C.Y. Ngu, W.S. YeoJ.C.Y. Ngu, W.S. Yeo, Prediction of dissolved oxygen using least square support vector regression model, in: 2022 International Conference on Green Energy, Computing and Sustainable Technology (GECOST), IEEE, Miri, Sarawak, Malaysia, (2022), 70-74. https://doi.org/10.1109/GECOST55694.2022.10010638. 2022.

[58] W.S. Yeo, A. Saptoro, P. Kumar, M. Kano, Just-in-time based soft sensors for process industries: A status report and recommendations, J. Process Control 128 (2023) 103025, https://doi.org/10.1016/j.jprocont.2023.103025.

[59] S.K. Lahiri, Multivariable predictive control: applications in industry, John Wiley & Sons, 2017.

[60] M.E. Khan, F. Khan, A comparative study of white box, black box and grey box testing techniques, Int. J. Adv. Comput. Sci. Appl. 3 (2012).

[61] T. Sheakh, A comparative study of software testing techniques viz. White box testing black box testing and grey box testing, Int. J. Allied Pract., Res. Rev. I (2015) 01–08.

[62] S. Nidhra, J. Dondeti, Black box and white box testing techniques-a literature review, Int. J. Embed. Syst. Appl. 2 (2012) 29–50, https://doi.org/10.5121/ijesa.2012.2204.

[63] I. Omrane, Design of an easy tunable soft sensor for real-time speed and position estimation of PMSM, Int. J. Mod. Commun. Technol. Res. 4 (2016) 265699.

[64] L. Xie, Y. Zhao, D. Aziz, X. Jin, L. Geng, E. Goberdhansingh, F. Qi, B. Huang, Soft sensors for online steam quality measurements of OTSGs, J. Process Control 23 (2013) 990–1000, https://doi.org/10.1016/j.jprocont.2013.05.006.

[65] M.K. Habib, S.A. Ayankoso, F. NagataM.K. Habib, S.A. Ayankoso, F. Nagata, Data-driven modeling: concept, techniques, challenges and a case study, in: 2021 IEEE international conference on mechatronics and automation (ICMA), IEEE, (2021), 1000-1007. https://doi.org/10.1109/ICMA52036.2021.9512658. 2021.

[66] B. Lin, B. Recke, J.K. Knudsen, S.B. Jørgensen, A systematic approach for soft sensor development, Comput. Chem. Eng. 31 (2007) 419–425.

[67] A. Barcaru, Supervised projection pursuit–A dimensionality reduction technique optimized for probabilistic classification, Chemom. Intell. Lab. Syst. 194 (2019) 103867.

[68] T.M. Hope, Linear regression. in: Machine Learning, Elsevier, 2020, pp. 67–81, https://doi.org/10.1016/B978-0-12-815739-8.00004-3.

[69] G. Pastorelli, S. Cao, I.K. Cigić, C. Cucci, A. Elnaggar, M. Strlič, Development of dose-response functions for historic paper degradation using exposure to natural conditions and multivariate regression, Polym. Degrad. Stab. 168 (2019) 108944, https://doi.org/10.1016/j.polymdegradstab.2019.108944.

[70] X. Jiang, X. You, S. Yu, D. Tao, C.P. Chen, Y.-m Cheung, Variance constrained partial least squares, Chemom. Intell. Lab. Syst. 145 (2015) 60–71, https://doi.org/10.1016/j.chemolab.2015.04.014.

[71] Z. Zhao, J. Wang, B. Sun, M. Arowo, L. Shao, Mass transfer study of water deoxygenation in a rotor–stator reactor based on principal component regression method, Chem. Eng. Res. Des. 132 (2018) 677–685, https://doi.org/10.1016/j.cherd.2018.02.007.

[72] L. Begdache, H. Kianmehr, N. Sabounchi, A. Marszalek, N. Dolma, Principal component regression of academic performance, substance use and sleep quality in relation to risk of anxiety and depression in young adults, Trends Neurosci. Educ. 15 (2019) 29–37, https://doi.org/10.1016/j.tine.2019.03.002.

[73] E.V. Thomas, Semi-supervised learning in multivariate calibration, Chemom. Intell. Lab. Syst. 195 (2019) 103868, https://doi.org/10.1016/j.chemolab.2019.103868.

[74] X. Yuan, Z. Ge, Z. Song, Locally weighted kernel principal component regression model for soft sensing of nonlinear time-variant processes, Ind. Eng. Chem. Res. 53 (2014) 13736–13749, https://doi.org/10.1021/ie4041252.

[75] W. Guo, S. Qin, Z. Zhao, Generalized ridge and principal correlation estimator of the regression coefficient in growth curve model, Linear Algebra Appl. 591 (2020) 115–133, https://doi.org/10.1016/j.laa.2020.01.006.

[76] F. Kherif, A. Latypova, Principal component analysis, : Mach. Learn., Elsevier (2020) 209–225, https://doi.org/10.1016/B978-0-12-815739-8.00012-2.

[77] M.R.M. Talabis, R. McPherson, I. Miyamoto, J.L. Martin, D. Kaye, Analytics defined, Inf. Secur. Anal. (2015) 1–12, https://doi.org/10.1016/b978-0-12-800207-0.00001-0.

[78] A. Nakar, Ze Schmilovitch, D. Vaizel-Ohayon, Y. Kroupitski, M. Borisover, S. Sela, Quantification of bacteria in water using PLS analysis of emission spectra of fluorescence and excitation-emission matrices, Water Res. 169 (2020) 115197, https://doi.org/10.1016/j.watres.2019.115197.

[79] F.H. LongF.H. Long, Multivariate analysis for metabolomics and proteomics data, in: Proteomic and metabolomic approaches to biomarker discovery, Elsevier, 2013, pp. 299-311. https://doi.org/10.1016/B978-0-12-394446-7.00019-4. 2013.

[80] W.S. Yeo, Prediction of Yellowness Index Using Partial Least Square Regression Model, : 2021 Int. Conf. Green. Energy, Comput. Sustain. Technol. (GECOST), IEEE (2021) 1–5, https://doi.org/10.1109/GECOST52368.2021.9538723.

[81] R. Chiplunkar, B. Huang, Output relevant slow feature extraction using partial least squares, Chemom. Intell. Lab. Syst. 191 (2019) 148–157, https://doi.org/10.1016/j.chemolab.2019.07.003.

[82] S. de Jong, SIMPLS: an alternative approach squares regression to partial least, Chemom. Intell. Lab. Syst. 18 (1993) 2–263, https://doi.org/10.1016/0169-7439(93)85002-X.

[83] K. Hazama, M. Kano, Covariance-based locally weighted partial least squares for high-performance adaptive modeling, Chemom. Intell. Lab. Syst. 146 (2015) 55–62, https://doi.org/10.1016/j.chemolab.2015.05.007.

[84] H. Jin, B. Pan, X. Chen, B. Qian, Ensemble just-in-time learning framework through evolutionary multi-objective optimization for soft sensor development of nonlinear industrial processes, Chemom. Intell. Lab. Syst. 184 (2019) 153–166, https://doi.org/10.1016/j.chemolab.2018.12.002.

[85] S. Kim, M. Kano, H. Nakagawa, S. Hasebe, Estimation of active pharmaceutical ingredients content using locally weighted partial least squares and statistical wavelength selection, Int. J. Pharm. 421 (2011) 269–274, https://doi.org/10.1016/j.ijpharm.2011.10.007.

[86] T.F. Thien, W.S. Yeo, A comparative study between PCR, PLSR, and LW-PLS on the predictive performance at different data splitting ratios, Chem. Eng. Commun. (2021) 1–18, https://doi.org/10.1080/00986445.2021.1957853.

[87] A. Merino, D. Garcia-Alvarez, G.I. Sainz-Palmero, L. Acebes, M. Fuente, Knowledge based recursive non-linear partial least squares (RNPLS), ISA Trans. 100 (2020) 481–494, https://doi.org/10.1016/j.isatra.2020.01.006.

[88] W. Zhao, Z. Tian, L. Yang, W. Yan, J. Wen, Image registration using a kernel partial least squares based mismatches removal method, AEU - Int. J. Electron. Commun. 70 (2016) 427–435, https://doi.org/10.1016/j.aeue.2016.01.001.

[89] S.M. Ross, Introductory statistics, Elvesier, London, 2017, https://doi.org/10.1016/B978-0-12-804317-2.00012-6.

[90] P. Mehta, M. Bukov, C.-H. Wang, A.G. Day, C. Richardson, C.K. Fisher, D. J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, Phys. Rep. 810 (2019) 1–124, https://doi.org/10.1016/j.physrep.2019.03.001.

[91] G. Díaz, J. Coto, J. Gómez-Aleixandre, Prediction and explanation of the formation of the Spanish day-ahead electricity price through machine learning regression, Appl. Energy 239 (2019) 610–625, https://doi.org/10.1016/j.apenergy.2019.01.213.

[92] G. Ciulla, A. D'Amico, Building energy performance forecasting: A multiple linear regression approach, Appl. Energy 253 (2019) 113500, https://doi.org/10.1016/j.apenergy.2019.113500.

[93] M. Ayoub, Degradation of tetracycline using nanoparticles of zero-valent iron and copper, Water Pract. Technol. 17 (2022) 246–253, https://doi.org/10.2166/wpt.2021.100.

[94] M. Vallejo, C. de la Espriella, J. Gómez-Santamaría, A.F. Ramírez-Barrera, E. Delgado-Trejos, Soft metrology based on machine learning: a review, Meas. Sci. Technol. 31 (2020) 032001, https://doi.org/10.1088/1361-6501/ab4b39.

[95] X. Li, J. Su, H. Wang, G. Boczkaj, J. Mahlknecht, S.V. Singh, C. Wang, Bibliometric analysis of artificial intelligence in wastewater treatment: current status, research progress, and future prospects, J. Environ. Chem. Eng. 12 (2024) 113152, https://doi.org/10.1016/j.jece.2024.113152.

[96] H. Zhang, A. Palazoglu, X. Zhang, W. Zhang, Z. Zhao, W. Sun, S. Liu, Prediction of surface ozone exceedance days using PCA with a non-parametric T^2 control limit, Chemom. Intell. Lab. Syst. 133 (2014) 42–48, https://doi.org/10.1016/j.chemolab.2014.02.005.

[97] N.A. Rashid, N.A.M. Rosely, M.A.M. Noor, A. Shamsuddin, M.K.A. Hamid, K. A. Ibrahim, Forecasting of refined palm oil quality using principal component regression, Energy Procedia 142 (2017) 2977–2982, https://doi.org/10.1016/j.egypro.2017.12.364.

[98] Z. Ge, Z. Song, L. Zhao, F. Gao, Two-level PLS model for quality prediction of multiphase batch processes, Chemom. Intell. Lab. Syst. 130 (2014) 29–36, https://doi.org/10.1016/j.chemolab.2013.09.008.

[99] X. Yuan, H. Zhang, Z. Song, A soft-sensor for estimating copper quality by image analysis technology, : 2013 10th IEEE Int. Conf. Control Autom. (ICCA), IEEE (2013) 991–996, https://doi.org/10.1109/ICCA.2013.6565042.

[100] O. Mouhtady, E. Obeid, M. Abu-Samha, K. Younes, N. Murshid, Evaluation of the Adsorption Efficiency of Graphene Oxide Hydrogels in Wastewater Dye Removal: Application of Principal Component Analysis, Gels 8 (2022) 447, https://doi.org/10.3390/gels8070447.

[101] X. Yuan, B. Huang, Z. Ge, Z. Song, Double locally weighted principal component regression for soft sensor with sample selection under supervised latent structure, Chemom. Intell. Lab. Syst. 153 (2016) 116–125, https://doi.org/10.1016/j.chemolab.2016.02.014.

[102] W.S. Yeo, A comparative study of different artificial intelligence models and response surface methodology for heptachlor removal using Fe/Cu nanoparticles, ASEAN Eng. J. 13 (2023) 157–163, https://doi.org/10.11113/aej.v13.20623.

[103] W. Cang, H. Yang, Adaptive soft sensor method based on online selective ensemble of partial least squares for quality prediction of chemical process, Asia-Pac. J. Chem. Eng. 14 (2019) e2346, https://doi.org/10.1002/apj.2346.

[104] H. Kaneko, K. Funatsu, Ensemble locally weighted partial least squares as a just-in-time modeling method, AIChE J. 62 (2016) 717–725, https://doi.org/10.1002/aic.15090.

[105] Y. Wang, H. Cao, Y. Zhou, Y. Zhang, Nonlinear partial least squares regressions for spectral quantitative analysis, Chemom. Intell. Lab. Syst. 148 (2015) 32–50, https://doi.org/10.1016/j.chemolab.2015.08.024.

[106] D.R. Rodrigues, A.C. Olivieri, W.D. Fragoso, S.G. Lemos, Complex numbers-partial least-squares applied to the treatment of electrochemical impedance spectroscopy data, Anal. Chim. Acta 1080 (2019) 1–11, https://doi.org/10.1016/j.aca.2019.07.047.

[107] R.S. Ibrahim, A. Khairy, H.H. Zaatout, H.M. Hammoda, A.M. Metwally, A. M. Salman, Chemometric evaluation of alfalfa sprouting impact on its metabolic profile using HPTLC fingerprint-efficacy relationship analysis modelled with partial least squares regression, J. Pharm. Biomed. Anal. 179 (2020) 112990, https://doi.org/10.1016/j.jpba.2019.112990.

[108] I.A. Naguib, F.F. Abdallah, Ultraviolet cutoff area and predictive ability of partial least squares regression method: a pharmaceutical case study, Spectrochim. Acta - A: Mol. Biomol. Spectrosc. 231 (2020) 118116, https://doi.org/10.1016/j.saa.2020.118116.

[109] C. Mei, Y. Ding, X. Chen, Y. Chen, J. Cai, Soft sensor modelling based on just-in-time learning and bagging-PLS for fermentation processes, Chem. Eng. Trans. 70 (2018) 1435–1440, https://doi.org/10.3303/CET1870240.

[110] Y.-W. Lin, B.-C. Deng, Q.-S. Xu, Y.-H. Yun, Y.-Z. Liang, The equivalence of partial least squares and principal component regression in the sufficient dimension reduction framework, Chemom. Intell. Lab. Syst. 150 (2016) 58–64, https://doi.org/10.1016/j.chemolab.2015.11.003.

[111] M.N. Pervez, W.S. Yeo, M.R. Mishu, A. Buonerba, Y. Zhao, Y. Cai, L. Lin, G. K. Stylios, V. Naddeo, Prediction of the diameter of biodegradable electrospun nanofiber membranes: an integrated framework of taguchi design and machine learning, J. Polym. Environ. 31 (2023) 4080–4096, https://doi.org/10.1007/s10924-023-02837-7.

[112] X. Zhang, M. Kano, S. Matsuzaki, A comparative study of deep and shallow predictive techniques for hot metal temperature prediction in blast furnace ironmaking, Comput. Chem. Eng. 130 (2019) 106575, https://doi.org/10.1016/j.compchemeng.2019.106575.

[113] J.C.Y. Ngu, W.S. Yeo, A comparative study of different kernel functions applied to LW-KPLS model for nonlinear processes, Biointerface Res. Appl. Chem. 13 (2023) 1–16, https://doi.org/10.33263/BRIAC132.184.

[114] W.S. Yeo, Yellowness index prediction using locally weighted kernel partial least square regression model, AIP Conf. Proc., AIP Publ. (2023), https://doi.org/10.1063/5.0165220.

[115] W.S. Yeo, A. Saptoro, P. Kumar, Development of adaptive soft sensor using locally weighted Kernel partial least square model, Chem. Prod. Process Model. 12 (2017), https://doi.org/10.1515/cppm-2017-0022.

[116] M. Chen, S. Khare, B. Huang, A unified recursive just-in-time approach with industrial near infrared spectroscopy application, Chemom. Intell. Lab. Syst. 135 (2014) 133–140, https://doi.org/10.1016/j.chemolab.2014.04.007.

[117] B. Pan, H. Jin, L. Wang, B. Qian, X. Chen, S. Huang, J. Li, Just-in-time learning based soft sensor with variable selection and weighting optimized by evolutionary optimization for quality prediction of nonlinear processes, Chem. Eng. Res. Des. 144 (2019) 285–299, https://doi.org/10.1016/j.cherd.2019.02.004.

[118] S. Qi, Y. Yan, C. Jinhui, Z. Yingying, Proximate analysis of sawdust using Near Infrared Spectroscopy and locally weighted partial least squares, Energy Procedia 88 (2016) 600–607, https://doi.org/10.1016/j.egypro.2016.06.085.

[119] X. Yuan, J. Zhou, Y. Wang, A spatial-temporal LWPLS for adaptive soft sensor modeling and its application for an industrial hydrocracking process, Chemom. Intell. Lab. Syst. 197 (2020) 103921, https://doi.org/10.1016/j.chemolab.2019.103921.

[120] M. Ren, Y. Song, W. Chu, An improved locally weighted PLS based on particle swarm optimization for industrial soft sensor modeling, Sensors 19 (2019) 4099.

[121] A. Saptoro, State of the art in the development of adaptive soft sensors based on just-in-time models, Procedia Chem. 9 (2014) 226–234, https://doi.org/10.1016/j.proche.2014.05.027.

[122] X. Zhang, M. Kano, Y. Li, Locally weighted kernel partial least squares regression based on sparse nonlinear features for virtual sensing of nonlinear time-varying processes, Comput. Chem. Eng. 104 (2017) 164–171, https://doi.org/10.1016/j.compchemeng.2017.04.014.

[123] K. Peng, K. Zhang, G. Li, Quality-related process monitoring based on total kernel PLS model and its industrial application, Math. Probl. Eng. 2013 (2013), https://doi.org/10.1155/2013/707953.

[124] Y. Zhang, Y. Fan, P. Zhang, Combining kernel partial least-squares modeling and iterative learning control for the batch-to-batch optimization of constrained nonlinear processes, Ind. Eng. Chem. Res. 49 (2010) 7470–7477, https://doi.org/10.1021/ie1004702.

[125] X. Liu, S. Zhou, Quality-related fault detection based on approximate kernel partial least squares method, J. Grid Comput. 21 (2023) 29, https://doi.org/10.1007/s10723-023-09670-1.

[126] Y. Liu, M. Xie, Rebooting data-driven soft-sensors in process industries: a review of kernel methods, J. Process Control 89 (2020) 58–73, https://doi.org/10.1016/j.jprocont.2020.03.012.

[127] H. Liu, Y. Zhang, H. Zhang, Prediction of effluent quality in papermaking wastewater treatment processes using dynamic kernel-based extreme learning machine, Process Biochem 97 (2020) 72–79, https://doi.org/10.1016/j.procbio.2020.06.020.

[128] Y. Peng, M.H. Nagata, An empirical overview of nonlinearity and overfitting in machine learning using COVID-19 data, Chaos, Solitons Fractals 139 (2020) 110055, https://doi.org/10.1016/j.chaos.2020.110055.

[129] S.-S. Jin, Compositional kernel learning using tree-based genetic programming for Gaussian process regression, Struct. Multidiscip. Optim. 62 (2020) 1313–1351, https://doi.org/10.1007/s00158-020-02559-7.

[130] D.P. Mesquita, J.P. Gomes, F. Corona, A.H.S. Junior, J.S. Nobre, Gaussian kernels for incomplete data, Appl. Soft Comput. 77 (2019) 356–365, https://doi.org/10.1016/j.asoc.2019.01.022.

[131] L. Shang, Z. Yan, A. Qiu, F. Li, X. Zhou, Efficient recursive kernel principal component analysis for nonlinear time-varying processes monitoring, : 2019 Chin. Control Decis. Conf. (CCDC), IEEE (2019) 3057–3062, https://doi.org/10.1109/CCDC.2019.8832617.

[132] P. Wu, L. Guo, S. Lou, J. Gao, Local and global randomized principal component analysis for nonlinear process monitoring, IEEE Access 7 (2019) 25547–25562, https://doi.org/10.1109/ACCESS.2019.2901128.

[133] G. Wang, J. Jiao, S. Yin, Efficient nonlinear fault diagnosis based on kernel sample equivalent replacement, IEEE Trans. Ind. Inform. 15 (2018) 2682–2690, https://doi.org/10.1109/TII.2018.2871515.

[134] B.P. Vrigazova, Detection of malignant and benign breast cancer using the Anova-Bootstrap-SVM, J. Data Inf. Sci. 5 (2020) 62–75, https://doi.org/10.2478/jdis-2020-0012.

[135] P.M. Kumar, R. Kavitha, Regression analysis for thermal properties of Al_2O_3/H_2O nanofluid using machine learning techniques, Heliyon 6 (2020) e03966, https://doi.org/10.1016/j.heliyon.2020.e03966.

[136] M. Zou, Z.-W. Zhou, L. Fan, W.-J. Zhang, L. Zhao, X.-P. Liu, H.-B. Wang, W.-S. Tan, A novel method based on nonparametric regression with a Gaussian kernel algorithm identifies the critical components in CHO media and feed optimization, J. Ind. Microbiol. Biotechnol. 47 (2020) 63–72, https://doi.org/10.1007/s10295-019-02248-5.

[137] K. Haddad, A. Rahman, Regional flood frequency analysis: evaluation of regions in cluster space using support vector regression, Nat. Hazards 102 (2020) 489–517, https://doi.org/10.1007/s11069-020-03935-8.

[138] S.D. Mehta, R. Sebro, Computer-aided detection of incidental lumbar spine fractures from routine dual-energy X-ray absorptiometry (DEXA) studies using a support vector machine (SVM) classifier, J. Digit. Imaging 33 (2020) 204–210, https://doi.org/10.1007/s10278-019-00224-0.

[139] M. Azimi-Pour, H. Eskandari-Naddaf, A. Pakzad, Linear and non-linear SVM prediction for fresh properties and compressive strength of high volume fly ash self-compacting concrete, Constr. Build. Mater. 230 (2020) 117021, https://doi.org/10.1016/j.conbuildmat.2019.117021.

[140] M.R. Kianifar, F. Campean, Performance evaluation of metamodelling methods for engineering problems: towards a practitioner guide, Struct. Multidiscip. Optim. 61 (2020) 159–186, https://doi.org/10.1007/s00158-019-02352-1.

[141] K. Bi, T. Qiu, An intelligent SVM modeling process for crude oil properties prediction based on a hybrid GA-PSO method, Chin. J. Chem. Eng. 27 (2019) 1888–1894, https://doi.org/10.1016/j.cjche.2018.12.015.

[142] A. Baghban, J. Sasanipour, S. Habibzadeh, Estimating solubility of supercritical H_2S in ionic liquids through a hybrid LSSVM chemical structure model, Chin. J. Chem. Eng. 27 (2019) 620–627, https://doi.org/10.1016/j.cjche.2018.08.026.

[143] J. Malang, W.S. Yeo, Z.Y. Chua, J. Nandong, A. Saptoro, A comparison study between different kernel functions in the least square support vector regression model for penicillin fermentation process, : MATEC Web Conf., EDP Sci. (2023) 01025, https://doi.org/10.1051/matecconf/202337701025.

[144] P. Joshi, Kernel Functions For Machine Learning, in, (2012).

[145] T. Kari, W. Gao, A. Tuluhong, Y. Yaermaimaiti, Z. Zhang, Mixed kernel function support vector regression with genetic algorithm for forecasting dissolved gas content in power transformers, Energies 11 (2018) 2437, https://doi.org/10.3390/en11092437.

[146] D. Jahed Armaghani, P.G. Asteris, B. Askarian, M. Hasanipanah, R. Tarinejad, V. V. Huynh, Examining hybrid and single SVM models with different kernels to predict rock brittleness, Sustainability 12 (2020) 2229, https://doi.org/10.3390/su12062229.

[147] M.G. Genton, Classes of kernels for machine learning: a statistics perspective, J. Mach. Learn. Res. 2 (2001) 299–312, https://doi.org/10.1162/15324430260185646.

[148] E. Dong, K. Zhou, J. Tong, S. Du, A novel hybrid kernel function relevance vector machine for multi-task motor imagery EEG classification, Biomed. Signal Process. Control 60 (2020) 101991, https://doi.org/10.1016/j.bspc.2020.101991.

[149] D. Duvenaud, Automatic model construction with Gaussian processes, in, University of Cambridge, (2014). https://doi.org/10.17863/CAM.14087.

[150] D. Jahed Armaghani, Y.Y. Ming, A. Salih Mohammed, E. Momeni, H. Maizir, Effect of SVM kernel functions on bearing capacity assessment of deep

foundations, J. Soft Comput. Civ. Eng. 7 (2023) 111–128, https://doi.org/10.22115/scce.2023.356959.1510.

[151] F. Lubbe, J. Maritz, T. Harms, Evaluating the potential of gaussian process regression for solar radiation forecasting: a case study, Energies 13 (2020) 5509, https://doi.org/10.3390/en13205509.

[152] X. Zhang, Q. Song, A multi-label learning based kernel automatic recommendation method for support vector machine, PLoS One 10 (2015) e0120455, https://doi.org/10.1371/journal.pone.0120455.

[153] S. Kazem, A. Hatam, Scattered data interpolation: Strictly positive definite radial basis/cardinal functions, J. Comput. Appl. Math. 394 (2021) 113580, https://doi.org/10.14569/IJACSA.2016.070160.

[154] R.L. Hardy, Least squares prediction, Photogramm. Eng. Remote Sens. 43 (1977) 475–492.

[155] S. Fadel, S. Ghoniemy, M. Abdallah, H.A. Sorra, A. Ashour, A. Ansary, Investigating the effect of different kernel functions on the performance of SVM for recognizing Arabic characters, Int. J. Adv. Comput. Sci. Appl. 7 (2016), https://doi.org/10.14569/IJACSA.2016.070160.

[156] J. Gorham, L. Mackey, Measuring sample quality with kernels, : Int. Conf. Mach. Learn., PMLR (2017) 1292–1301, https://doi.org/10.48550/arXiv.1703.01717.

[157] K. He, C. Li, Y. Yang, G. Huang, J.E. Hopcroft, Integrating large circular kernels into cnns through neural architecture search, in: arXiv preprint arXiv: 2107.02451, (2022). https://doi.org/10.48550/arXiv.2107.02451.

[158] S. Boughorbel, J.-P. Tarel, F. Fleuret, N. Boujemaa, The GCS kernel for SVM-based image recognition, in: W. Duch, J. Kacprzyk, E. Oja, S. Zadrożny (Eds.), Artificial Neural Networks: Formal Models and Their Applications – ICANN 2005, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 595–600, https://doi.org/10.1007/11550907_94.

[159] H. Lei, N. Akhtar, A. Mian, Spherical kernel for efficient graph convolution on 3d point clouds, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2020) 3664–3680, https://doi.org/10.1109/TPAMI.2020.2983410.

[160] H. ElGhawalby, E.R. Hancock, Graph embedding using an edge-based wave kernel, in: E.R. Hancock, R.C. Wilson, T. Windeatt, I. Ulusoy, F. Escolano (Eds.), Structural, Syntactic, and Statistical Pattern Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 60–69, https://doi.org/10.1007/978-3-642-14980-1_5.

[161] F. Farzin, S.S. Moghaddam, M. Ehteshami, Auto-tuning data-driven model for biogas yield prediction from anaerobic digestion of sewage sludge at the south-tehran wastewater treatment plant: Feature selection and hyperparameter population-based optimization, Renew. Energy 227 (2024) 120554, https://doi.org/10.1016/j.renene.2024.120554.

[162] K. Obaideen, N. Shehata, E.T. Sayed, M.A. Abdelkareem, M.S. Mahmoud, A. Olabi, The role of wastewater treatment in achieving sustainable development goals (SDGs) and sustainability guideline, Energy Nexus 7 (2022) 100112, https://doi.org/10.1016/j.nexus.2022.100112.

[163] M. Nasr, K. Mohamed, M. Attia, M.G. Ibrahim, Sustainable management of wastewater treatment plants using artificial intelligence techniques. in: Soft Computing Techniques in Solid Waste and Wastewater Management, Elsevier, 2021, pp. 171–185, https://doi.org/10.1016/B978-0-12-824463-0.00009-4.

[164] M. Najafzadeh, M. Zeinolabedini, Prognostication of waste water treatment plant performance using efficient soft computing models: an environmental evaluation, Measurement 138 (2019) 690–701, https://doi.org/10.1016/j.measurement.2019.02.014.

[165] M. Najafzadeh, M. Zeinolabedini, Derivation of optimal equations for prediction of sewage sludge quantity using wavelet conjunction models: an environmental assessment, Environ. Sci. Pollut. Res. 25 (2018) 22931–22943, https://doi.org/10.1007/s11356-018-1975-5.

[166] E. Buonocore, S. Mellino, G. De Angelis, G. Liu, S. Ulgiati, Life cycle assessment indicators of urban wastewater and sewage sludge treatment, Ecol. Indic. 94 (2018) 13–23, https://doi.org/10.1016/j.ecolind.2016.04.047.

[167] M. Mauricio-Iglesias, I. Montero-Castro, A.L. Mollerup, G. Sin, A generic methodology for the optimisation of sewer systems using stochastic programming and self-optimizing control, J. Environ. Manag. 155 (2015) 193–203, https://doi.org/10.1016/j.jenvman.2015.03.034.

[168] S. Li, Y. Li, Q. Lu, J. Zhu, Y. Yao, S. Bao, Integrated drying and incineration of wet sewage sludge in combined bubbling and circulating fluidized bed units, Waste Manag. (Oxf.) 34 (2014) 2561–2566, https://doi.org/10.1016/j.wasman.2014.08.018.

[169] F.S. Mjalli, S. Al-Asheh, H. Alfadala, Use of artificial neural network black-box modeling for the prediction of wastewater treatment plants performance, J. Environ. Manag. 83 (2007) 329–338, https://doi.org/10.1016/j.jenvman.2006.03.004.

[170] H. Moral, A. Aksoy, C.F. Gokcay, Modeling of the activated sludge process by using artificial neural networks with automated architecture screening, Comput. Chem. Eng. 32 (2008) 2471–2478, https://doi.org/10.1016/j.compchemeng.2008.01.008.