Nonlinear dynamic transfer partial least squares for domain adaptive regression

Zhao, Zhijun; Yan, Gaowei; Ren, Mifeng; Cheng, Lan; Li, Rong; Pang, Yusong

DOI
10.1016/j.isatra.2024.08.002

Publication date
2024

Document Version
Final published version

Published in
ISA Transactions

ISA Transactions 153 (2024) 262–275

Contents lists available at ScienceDirect

ISA Transactions

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

Research article

Nonlinear dynamic transfer partial least squares for domain adaptive regression

Zhijun Zhao [a], Gaowei Yan [a,b,*], Mifeng Ren [a], Lan Cheng [a], Rong Li [a], Yusong Pang [c]

[a] College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, 030024, Shanxi, China
[b] Shanxi Research Institute of Huairou Laboratory, Taiyuan, 030032, Shanxi, China
[c] Faculty of Mechanical Engineering, Delft University of Technology, Delft, 2628CD, Netherlands

ARTICLE INFO

Keywords:
Domain adaptive regression
Dynamic partial least squares
Soft sensor
Transfer learning

ABSTRACT

Aiming to address soft sensing model degradation under changing working conditions, and to accommodate dynamic, nonlinear, and multimodal data characteristics, this paper proposes a nonlinear dynamic transfer soft sensor algorithm. The approach leverages time-delay data augmentation to capture dynamics and projects the augmented data into a latent space for constructing a nonlinear regression model. Two regular terms, distribution alignment regularity and first-order difference regularity, are introduced during data projection to address data distribution disparities. Laplace regularity is incorporated into the nonlinear regression model to ensure geometric structure preservation. The final optimization objective is formulated within the framework of partial least squares, and hyperparameters are determined using Bayesian optimization. The effectiveness of the proposed algorithm is demonstrated through experiments on three public datasets.

1. Introduction

The advancement of big data and artificial intelligence (AI) technology provides unprecedented opportunities and challenges for the development of industrial AI [1]. Data-driven soft sensing [2–4] is a typical application of AI technology in the industrial field. It utilizes regression models built from easy-to-measure and unmeasurable variables to achieve rapid prediction of unmeasurable variables. Easily measurable variables refer to the data measured by conventional sensors, such as pressure, temperature, flow, level, and other signals. Unmeasurable variables refer to the variables that cannot be monitored online due to the limited installation environment, or the lag is too large to meet the real-time requirements, or the instruments are too expensive and the maintenance costs are too high and cannot meet the economic requirements. For example, in the high-temperature environment of the gasification melting furnace, the melting index cannot be measured online [5]; The ore grade in the beneficiation production process [6] and the harmful components in waste incineration residues [7] need to be tested by instruments in a laboratory environment, with serious lags; The monitoring of flue gas components in the coal-fired power generation process requires online analytical instruments [8], which are costly to operate and maintain.

Traditional data-driven soft sensing methods comprise statistical and machine learning methods. Representative methods include Partial Least Squares (PLS) [9] and Extreme Learning Machine (ELM) [10]. As a supervised method, the PLS has achieved many successful applications in the fields of soft sensing and process monitoring. However, the original PLS is a static and linear modeling method and needs to follow the basic Gauss–Markov assumption [11], which is difficult to satisfy in the actual industrial process.

With the rapid development of industry and the intensification of market competition, the demand for product diversification is increasing. To meet the market's diverse needs and reduce costs, the production process has been continuously optimized, and advanced control strategies such as optimal and boundary control have been gradually applied to the real production process. Meanwhile, accompanied by equipment reorganization, and the changes in internal and external operating conditions, the actual production process is frequently switched between different working conditions, making the data present characteristics such as dynamic [12], nonlinear [13], non-stationary [14], and multi-modal [15].

In response to the dynamic characteristics of data, a series of dynamic modeling methods have been proposed, which can be divided into three categories: dynamic extension methods, dynamic feature extraction methods, and state space methods [16]. The dynamic expansion methods expand the original data utilizing direct data augmentation and then apply traditional multivariate statistical methods

* Corresponding author at: College of Electrical and Power Engineering, Taiyuan University of Technology, Taiyuan, 030024, Shanxi, China.
E-mail addresses: 273335011@qq.com (Z. Zhao), yangaowei@tyut.edu.cn (G. Yan), renmifeng@126.com (M. Ren), taolan_1983@126.com (L. Cheng), lirong@tyut.edu.cn (R. Li), Y.Pang@tudelft.nl (Y. Pang).

https://doi.org/10.1016/j.isatra.2024.08.002
Received 3 March 2023; Received in revised form 2 August 2024; Accepted 2 August 2024
Available online 13 August 2024

to the augmented data. For instance, Ricker [17] extended the PLS method based on the Finite Impulse Response (FIR) and proposed the dynamic PLS (DPLS) model. The direct data extension methods are simple and convenient to implement, so they are broadly used in process monitoring and soft sensing. Whereas, the ways of direct data augmentation cause the increase of variable dimension, which are computationally inefficient. Based on this, dynamic feature extraction methods are proposed, for example, Dong and Qin [18] proposed the dynamic inner model (DiPLS), by constructing the autocorrelation relationship among latent variables, explicitly expressing the dynamic properties, and making the model explanatory. State-space models are another approach to modeling dynamical systems, the most commonly used state-space model is Canonical Variable Analysis (CVA) [19].

Process nonlinearity is another important characteristic of modern industry. There are often existing nonlinear relationships between process variables [20]. The original PLS model and the above-mentioned dynamically extended DPLS and DiPLS models are entirely linear models, which cannot reveal the nonlinear relationships. The kernel method is a common nonlinear modeling method, which maps the low-dimensional process variables to the high-dimensional feature space through a nonlinear function, and then constructs a linear regression model in the high-dimensional feature space. Applying the above principles, Rosipal and Trejo [21] proposed the KPLS algorithm, and Bennett and Embrechts [22] proposed the DK-PLS algorithm. These methods based on kernel expansion linearize nonlinear relationships between variables, improving the model's ability to explain nonlinear data. Another common nonlinear modeling method is the extension method based on the inner model, which maps the process variable into the latent space, and constructs a nonlinear model in the latent space. Qin and McAvoy [23] used the neural network to establish an inner model between latent variables and proposed the NNPLS algorithm. Lv et al. [24] used the LSSVM method to construct the inner model and proposed the LSSVM-PLS method. Yang et al. [25] and Liu et al. [26] proposed the D-RVM-PLS and D-GPR-PLS algorithms, which use direct matrix augmentation to expand input features and apply RVM and GPR on the augmented data to build an inner model between latent variables, those algorithms improve the fitting ability to nonlinear data while obtaining the dynamic characteristics.

The dynamic, nonlinear, non-stationary, and multi-modal characteristics of data under multiple working conditions require that the soft sensor model has sufficient expressive ability. At the same time, the samples used for training can cover these characteristics, so that the model can learn these characteristics, which is difficult to meet in actual industrial applications. This is because the samples used for training are always limited and it is difficult to cover all production conditions. When a new working condition appears, the existing model is difficult to adapt to the variation of the new working condition, which may easily cause a mismatch of the model. In response to this problem, Wang et al. [27] introduced transfer learning into the process of linear dynamic system modeling to solve the cold start problem of new working conditions in process monitoring. Nikzad-Langerodi et al. [28] combined transfer learning with the nonlinear iterative partial least squares method, and provided a transfer regression model with domain invariant expression. Zhao et al. [29] added domain adaptation regularization in the dynamic system modeling process, and proposed a dynamic transfer soft sensor algorithm, which reduced the impact of data distribution differences on the soft sensor model. Gao et al. [30] applied the meta-learning method to the multi-modal soft sensing process, which significantly improved the prediction accuracy of the model.

Transfer learning [31] can apply the knowledge learned in a certain task or field to a different but related task or field, so that it can be used to address the issue of difficult data label acquisition. At the same time, transfer learning breaks the independent and identical distribution (iid) hypothesis that traditional machine learning requires so that the fields or tasks involved in learning can obey different marginal or conditional probability distributions. For the problem of soft sensor modeling, using plenty of existing operating condition data to train the model and transfer it to the unknown operating condition can effectively handle the problems of model mismatch and degradation attributable to the data distribution shift under variable operating conditions.

Aiming at the dynamic, nonlinear, and multi-modal characteristics of data in the soft sensor modeling process of the modern process industry, so as to handle the problems of model mismatch and model degradation caused by data distribution differences under variable operating conditions, this paper proposes a nonlinear dynamic transfer partial least squares algorithm (NDTPLS). Firstly, sample data are augmented with time delay to obtain the dynamic characteristics. Secondly, the augmented sample data which are high-dimensional are projected into the latent space, and a nonlinear regression model among label data and latent variables is constructed. In the process of data projection, two regular terms named distribution alignment and first order difference regularization are added to deal with the distribution difference of the data. In the nonlinear regression model, the Laplace regular term is added to achieve geometric structure preservation. The final optimization objective is solved under the framework of partial least squares. The hyperparameters are obtained by the Bayesian optimization method.

The main contribution of this article can be summarized as follows:

(a) A brand new adaptive soft sensor modeling method has been proposed by combining unsupervised domain adaptation and dynamic Partial Least Squares under variable operation conditions.
(b) The proposed dynamic transfer outer model based on matrix augmentation realizes adaptive alignment of data distribution differences.
(c) The proposed nonlinear Laplacian inner model realizes the nonlinear mapping of the data through the kernel method, and the Laplacian regularization guarantees the manifold structure of the mapped data. The Laplacian regularization is constructed through transductive learning. To the best of our knowledge, this is the first time that a nonlinear inner model has been constructed using transductive learning, in contrast to previous nonlinear inner models that have been constructed using inductive learning.

The remaining sections of this article are organized as follows: Section 2 introduces the baseline methods which include linear PLS, nonlinear PLS, dynamic PLS, and dynamic nonlinear PLS. Section 3 provides a problem statement about transfer soft sensors and describes our proposed method. Section 4 presents the experimental setup, and hyperparameter analysis and discusses the results. Section 5 concludes this paper.

2. Preliminaries

In this section, the linear PLS, nonlinear PLS, dynamic PLS, and dynamic nonlinear PLS methods are described in mathematical language.

2.1. Linear PLS and nonlinear PLS

Given feature input matrix $\boldsymbol{X} \in \mathbb{R}^{n\times m}$, label output matrix $\boldsymbol{Y} \in \mathbb{R}^{n\times d}$, where the index n is the number of samples, m and d represent the feature dimension. The PLS algorithm projects the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ into the hidden spaces to obtain the first pair of principal components $\boldsymbol{t}$ and $\boldsymbol{u}$, so that on the one hand $\boldsymbol{t}$ and $\boldsymbol{u}$ carry as much variation information as possible, on the other hand, the correlation between $\boldsymbol{t}$ and $\boldsymbol{u}$ is maximum. The formal mathematical expression should be formulated as an optimization task,

$$\begin{aligned} &max\ cov\,(\boldsymbol{u},\boldsymbol{t}) = \boldsymbol{c}^{\mathrm{T}}\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w} \\ &s.t. \quad \|\boldsymbol{c}\| = 1, \|\boldsymbol{w}\| = 1 \end{aligned} \tag{1}$$

where $\boldsymbol{w}$ and $\boldsymbol{c}$ represent the input and output weighting vectors, which can be solved by eigenvalue decomposition or nonlinear iterative method (NIPALS) [32].

Remark 1. $cov(\boldsymbol{u},\boldsymbol{t}) = \sqrt{var(\boldsymbol{u})\,var(\boldsymbol{t})}\,r(\boldsymbol{u},\boldsymbol{t})$. Maximizing the covariance of $\boldsymbol{u}$ and $\boldsymbol{t}$ implies that the variance of $\boldsymbol{u}$ and $\boldsymbol{t}$ should be maximized, while their correlation coefficient $r(\boldsymbol{u},\boldsymbol{t})$ should be maximized.

After obtaining $\boldsymbol{w}$ and $\boldsymbol{c}$, the latent variables can be computed as follows,

$$\begin{aligned}\boldsymbol{t} &= \boldsymbol{X}\boldsymbol{w}\\ \boldsymbol{u} &= \boldsymbol{Y}\boldsymbol{c}\end{aligned} \tag{2}$$

establish the regression equation of $\boldsymbol{X}$, $\boldsymbol{Y}$ to $\boldsymbol{t}$, $\boldsymbol{u}$,

$$\begin{aligned}\boldsymbol{X} &= \boldsymbol{t}\boldsymbol{p}^{\mathrm{T}} + \boldsymbol{E}\\ \boldsymbol{Y} &= \boldsymbol{u}\boldsymbol{q}^{\mathrm{T}} + \boldsymbol{F}\end{aligned} \tag{3}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ represent loading vectors, and $\boldsymbol{E}$ and $\boldsymbol{F}$ are the residual matrices. For linear PLS, the intrinsic relationship between $\boldsymbol{u}$ and $\boldsymbol{t}$ is obtained by simple linear regression,

$$\boldsymbol{u} = b\boldsymbol{t} + \boldsymbol{r} \tag{4}$$

where b is the regression coefficient, $\boldsymbol{r}$ is the residual. For nonlinear PLS, the intrinsic relationship between $\boldsymbol{u}$ and $\boldsymbol{t}$ is obtained through the nonlinear mapping function $f(\bullet)$,

$$\boldsymbol{u} = f(\boldsymbol{t}) + \boldsymbol{r} \tag{5}$$

when $\boldsymbol{t}$ and $\boldsymbol{u}$ do not extract enough information, the original data matrix is deflated through the regression relationship established above to further extract the second pair of principal components, for linear PLS,

$$\begin{aligned}\boldsymbol{E} &= \boldsymbol{X} - \boldsymbol{t}\boldsymbol{p}^{\mathrm{T}}\\ \boldsymbol{F} &= \boldsymbol{Y} - b\boldsymbol{t}\boldsymbol{q}^{\mathrm{T}}\end{aligned} \tag{6}$$

for nonlinear PLS,

$$\begin{aligned}\boldsymbol{E} &= \boldsymbol{X} - \boldsymbol{t}\boldsymbol{p}^{\mathrm{T}}\\ \boldsymbol{F} &= \boldsymbol{Y} - f(\boldsymbol{t})\,\boldsymbol{q}^{\mathrm{T}}\end{aligned} \tag{7}$$

replace $\boldsymbol{X}$ and $\boldsymbol{Y}$ in the formula (1) with $\boldsymbol{E}$ and $\boldsymbol{F}$ respectively, so as to obtain the second pair of principal components. Repeat the above process until the required A pair principal components are obtained. Since the formulas (2) and (3) express the relationship between external variables and latent variables, it is customarily called an outer model. The formulas (4) and (5) express the relationship among internal latent variables, so they are customarily called inner models.

2.2. Dynamic PLS and dynamic nonlinear PLS

The linear (formula (4) and (6)) and nonlinear (formula (5) and (7)) PLS models are static models and fail to represent the dynamic characteristics of the data. The method of direct matrix augmentation converts the dynamic modeling problem of time series into a static modeling problem in space by expanding the serialized historical samples into features, which can effectually obtain the dynamic characteristics of the data. In the field of soft sensing, the Finite Impulse Response (FIR) and the Auto-regressive with Exogenous Inputs (ARX) are two commonly used data augmentation methods.

For the input matrix $\boldsymbol{X} = \left[\boldsymbol{x}_0\,\boldsymbol{x}_1\,\cdots\,\boldsymbol{x}_{n-1}\right]^{\mathrm{T}} \in \mathbb{R}^{n\times m}$ and the output matrix $\boldsymbol{Y} = \left[\boldsymbol{y}_0\,\boldsymbol{y}_1\,\cdots\,\boldsymbol{y}_{n-1}\right]^{\mathrm{T}} \in \mathbb{R}^{n\times d}$, the matrix augmented by FIR can be expressed as,

$$\begin{aligned}\boldsymbol{Z}_\tau &= \left[\boldsymbol{X}_\tau\,\boldsymbol{X}_{\tau-1}\,\cdots\,\boldsymbol{X}_0\right]\\ \boldsymbol{Y}_\tau &= \left[\boldsymbol{y}_\tau\,\boldsymbol{y}_{\tau+1}\,\cdots\,\boldsymbol{y}_{n-1}\right]^{\mathrm{T}}\end{aligned} \tag{8}$$

where τ is the delay coefficient, $\boldsymbol{X}_i = \left[\boldsymbol{x}_i\,\boldsymbol{x}_{i+1}\,\cdots\,\boldsymbol{x}_{i+n-\tau-1}\right]^{\mathrm{T}}$, for $i = 0,1,2,\ldots,\tau$.

Applying the linear and nonlinear PLS to the augmented data matrix, the optimization objective formula of the dynamic linear and dynamic nonlinear partial least squares algorithm can be obtained as,

$$\begin{aligned}&max\,cov\left(\boldsymbol{u}_i,\boldsymbol{t}_i\right) = \boldsymbol{c}_i^{\mathrm{T}}\boldsymbol{Y}_\tau^{\mathrm{T}}\boldsymbol{Z}_\tau\boldsymbol{w}_i\\ &s.t.\quad \|\boldsymbol{c}_i\| = 1, \|\boldsymbol{w}_i\| = 1\end{aligned} \tag{9}$$

the inner and outer models are as follows,

$$\begin{aligned}\boldsymbol{Z}_\tau &= \sum_{i=1}^{A}\boldsymbol{t}_i\boldsymbol{p}_i^{\mathrm{T}} + \boldsymbol{E} = \boldsymbol{T}\boldsymbol{P}^{\mathrm{T}} + \boldsymbol{E}\\ \boldsymbol{Y}_\tau &= \sum_{i=1}^{A}\boldsymbol{u}_i\boldsymbol{q}_i^{\mathrm{T}} + \boldsymbol{F} = \boldsymbol{U}\boldsymbol{Q}^{\mathrm{T}} + \boldsymbol{F}\\ \boldsymbol{u}_i &= b\boldsymbol{t}_i + \boldsymbol{r}_i \quad or \quad \boldsymbol{u}_i = f\left(\boldsymbol{t}_i\right) + \boldsymbol{r}_i\end{aligned} \tag{10}$$

In summary, the nonlinear PLS method can be obtained by changing the relationship between the principal components $\boldsymbol{u}$ and $\boldsymbol{t}$ from linear to nonlinear. QPLS, NNPLS, and LSSVMPLS, introduced in Section 1, all belong to the nonlinear expansion method. At the same time, the dynamics of the system can be obtained through direct data augmentation (such as FIR), thereby extending the linear PLS and nonlinear PLS methods into dynamic PLS and dynamic nonlinear PLS methods. D-GPR-PLS and D-RVM-PLS in the literature both belong to this dynamic expansion method. These different nonlinear modeling and dynamic expansion methods are compared and summarized in Table 1.

3. Methodology

Under variable working conditions, the non-stationary and multimodal characteristics of the data make the distribution of the data to be predicted different from that used for modeling. This section provides a detailed description of the unsupervised transfer soft sensor modeling method, focusing on the perspective of data distribution.

3.1. Problem statement

The labeled historical working condition data $\boldsymbol{X}^{\mathrm{s}}, \boldsymbol{Y}^{\mathrm{s}}$ and the unlabeled working condition data $\boldsymbol{X}^{\mathrm{t}}$ are respectively augmented to obtain the dynamics of the data. The augmented historical operating data is recorded as $(\boldsymbol{Z}_\tau^{\mathrm{s}}, \boldsymbol{Y}_\tau^{\mathrm{s}})$, which is defined as the source domain $\mathbb{D}_{\mathrm{S}}$. The augmented target operating condition data is recorded as $\boldsymbol{Z}_\tau^{\mathrm{t}}$, which is defined as the target domain $\mathbb{D}_{\mathrm{T}}$. Assume that $\mathbb{D}_{\mathrm{S}}$ and $\mathbb{D}_{\mathrm{T}}$ share the same feature space ($\boldsymbol{Z}_\tau^{\mathrm{s}}, \boldsymbol{Z}_\tau^{\mathrm{t}} \in \mathbb{Z}$) and label space ($\boldsymbol{Y}_\tau^{\mathrm{s}}, \boldsymbol{Y}_\tau^{\mathrm{t}} \in \mathbb{Y}$), but the data distribution is different, that is $P_{\mathrm{s}}(\boldsymbol{Z}_\tau^{\mathrm{s}}) \neq P_{\mathrm{t}}(\boldsymbol{Z}_\tau^{\mathrm{t}})$, $Q_{\mathrm{s}}(\boldsymbol{Y}_\tau^{\mathrm{s}}|\boldsymbol{Z}_\tau^{\mathrm{s}}) \neq Q_{\mathrm{t}}(\boldsymbol{Y}_\tau^{\mathrm{t}}|\boldsymbol{Z}_\tau^{\mathrm{t}})$. Define $h : \mathbb{Z} \to \mathbb{Y}$ as the labeling function. Domain adaptation transfer soft sensor aims to find out an empirical mapping $\hat{h} : \mathbb{Z} \to \mathbb{Y}$ through the knowledge of the source domain to minimize the expected error under the target domain, that is,

$$min\,\epsilon_{\mathrm{T}}\left(\hat{h}, h\right) = \mathbb{E}_{\mathbb{D}_{\mathrm{T}}}\left[|\hat{h} - h|\right] \tag{11}$$

By the theory of Structural Risk Minimization (SRM) [35], empirical labeling function $\hat{h} = argmin\,L(h(x), y) + R(h)$, where $L(h(x), y)$ is the empirical loss, and $R(h)$ is a regular term representing the complexity of the model. For the domain adaptation soft sensor modeling task, the empirical loss is obtained through the source domain since the target domain has no labels. However, due to the different distribution and non-stationary nature of the data, it is not enough to learn the label mapping function only through the empirical loss. Therefore, the distribution alignment regularization term and the first-order difference regularization term are introduced to learn the domain invariant representation so as to minimize the expected error on the target domain.

Table 1
The comparison of different kind of PLS methods.

Year	Method	Type	Pros and Cons
1975	NIPALS [32]	Linear	The first PLS method, which solves the multicollinearity problem, but it is a linear model not applicable to nonlinear problems
1988	DPLS [17]	Dynamic	Extending the PLS method using FIR, makes it feasible to solve dynamic problems
1989	QPLS [33]	Nonlinear	Nonlinear extension of PLS methods using quadratic polynomials
1992	NNPLS [23]	Nonlinear	Nonlinear extension of PLS methods using neural networks
2001	KPLS [21]	Nonlinear	Kernel extensions of the PLS method, inefficient when the data size is large
2003	DKPLS [22]	Nonlinear	Direct kernel extension of PLS methods, making it feasible for large-scale problem
2012	LSSVMPLS [24]	Nonlinear	Nonlinear model with internal and external consistency by using inner model error to update outer model weights
2018	DiPLS [18]	Dynamic and linear	Dynamic models that are internally and externally consistent, more explanatory, but still linear models
2019	D-GPR-PLS [26]	Dynamic and nonlinear	The dynamic problem is considered as well as the nonlinear problem
2020	DIPALS [28]	Linear	Introducing domain-invariant expressions to reduce the effect of differences in data distribution, but still linear model
2021	D-RVM-PLS [25]	Dynamic and nonlinear	Adaptive dynamic expansion mechanism
2022	DTPLS [29]	Dynamic and linear	Internally and externally consistent dynamic models that take into account differences in data distribution
2022	GRU-PLS [34]	Dynamic and nonlinear	Using GRU to express internal dynamics
2023	TDLVR [12]	Dynamic and linear	Introduction of co-dynamic variations and error compensation mechanisms, but still linear model

3.2. Reconstruct error minimization

Based on the above SRM theory, a source domain Empirical Risk Minimization (ERM) function is established on the FIR augmented data, which is,

$$\arg\min_{\|\boldsymbol{w}\|=1,\|c\|=1} \mathcal{L} = \|\boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w} - \boldsymbol{Y}_\tau^{\mathrm{s}}\boldsymbol{c}\|_2^2 \tag{12}$$

the objective is to find the optimal weighting vectors w and c so that the distance between the hidden variables $\boldsymbol{t}$ ($\boldsymbol{t} = \boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w}$) and $\boldsymbol{u}$ ($\boldsymbol{u} = \boldsymbol{Y}_\tau^{\mathrm{s}}\boldsymbol{c}$) can be as small as possible. The above objective function has two unknown vectors and can be solved by some numerical optimization algorithm, but a closed-form solution cannot be obtained. To make it easy to solve, it is assumed that $\boldsymbol{Y}_\tau^{\mathrm{s}}$ is univariate, and thus $\boldsymbol{c} = 1$. To prevent symbol abuse, the vector $\boldsymbol{y}_\tau^{\mathrm{s}}$ is used in place of the matrix $\boldsymbol{Y}_\tau^{\mathrm{s}}$. According to the compatibility relationship between vector norm and Frobenius norm,

$$\|\boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w} - \boldsymbol{y}_\tau^{\mathrm{s}}\|_2^2 \le \|\boldsymbol{Z}_\tau^{\mathrm{s}} - \boldsymbol{y}_\tau^{\mathrm{s}}\boldsymbol{w}^{\mathrm{T}}\|_{\mathrm{F}}^2\|\boldsymbol{w}\|_2^2 \tag{13}$$

under the constraint $\|\boldsymbol{w}\| = 1$, an upper bound of empirical error can be obtained, which represents the reconstruction error of the source domain. This upper bound serves as a loss function, which is,

$$\arg\min_{\boldsymbol{w}} \mathcal{L}_{\mathrm{ERM}} = \|\boldsymbol{Z}_\tau^{\mathrm{s}} - \boldsymbol{y}_\tau^{\mathrm{s}}\boldsymbol{w}^{\mathrm{T}}\|_{\mathrm{F}}^2 \tag{14}$$

Besides, according to the properties of Frobenius norm, it can be inferred that,

$$\begin{aligned}\|\boldsymbol{Z}_\tau^{\mathrm{s}} - \boldsymbol{y}_\tau^{\mathrm{s}}\boldsymbol{w}^{\mathrm{T}}\|_{\mathrm{F}}^2 &= tr\left(\boldsymbol{Z}_\tau^{\mathrm{s}}\left(\boldsymbol{Z}_\tau^{\mathrm{s}}\right)^{\mathrm{T}}\right) + tr\left(\boldsymbol{y}_\tau^{\mathrm{s}}\left(\boldsymbol{y}_\tau^{\mathrm{s}}\right)^{\mathrm{T}}\right) + tr\left(-2\boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w}\left(\boldsymbol{y}_\tau^{\mathrm{s}}\right)^{\mathrm{T}}\right) \\ &= -2\,cov\left(\boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w}, \boldsymbol{y}_\tau^{\mathrm{s}}\right) + \text{constant}\end{aligned} \tag{15}$$

comparing the formulas (9) and (15), it can be seen that for a single output system, minimizing the source domain reconstruction error is equivalent to maximizing the hidden variable covariance.

3.3. Dynamic transfer outer model

Based on the source domain reconstruction error, and introducing distribution alignment regularization and first-order difference regularization at the same time, a dynamic transfer outer model is established as follows,

$$\begin{aligned}&\arg\min_{\boldsymbol{w}} \mathcal{L}_{\mathrm{ERM}} + \lambda\mathcal{L}_{\mathrm{DDA}} + \rho\mathcal{L}_{\mathrm{FOD}} \\ &= \arg\min_{\boldsymbol{w}} \|\boldsymbol{Z}_\tau^{\mathrm{s}} - \boldsymbol{y}_\tau^{\mathrm{s}}\boldsymbol{w}^{\mathrm{T}}\|_{\mathrm{F}} + \lambda\left|\frac{1}{n_{\mathrm{S}}}\left(\boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w}\right)^{\mathrm{T}}\boldsymbol{Z}_\tau^{\mathrm{s}}\boldsymbol{w} - \frac{1}{n_{\mathrm{T}}}\left(\boldsymbol{Z}_\tau^{\mathrm{t}}\boldsymbol{w}\right)^{\mathrm{T}}\boldsymbol{Z}_\tau^{\mathrm{t}}\boldsymbol{w}\right| \\ &\quad + \rho\left(\|\dot{\boldsymbol{Z}}_\tau^{\mathrm{s}}\boldsymbol{w}\|_2^2 + \|\dot{\boldsymbol{Z}}_\tau^{\mathrm{t}}\boldsymbol{w}\|_2^2\right)\end{aligned} \tag{16}$$

in the formula (16), the second item is the distribution difference alignment item, which is defined as $\mathcal{L}_{\mathrm{DDA}}$. The third term is the first-order difference regular term, which is defined as $\mathcal{L}_{\mathrm{FOD}}$. λ and ρ are regularization coefficients, n_{S} and n_{T} are the number of samples after matrix augmentation. $\dot{\boldsymbol{Z}}_\tau^{\mathrm{s}}$ and $\dot{\boldsymbol{Z}}_\tau^{\mathrm{t}}$ represent the first difference of data. The architecture of proposed dynamic transfer outer model is shown in Fig. 1.

Sample statistics are an effective description of data distribution, and many of the most commonly used statistics can be constructed from sample moments. When the data distribution is unknown, the difference between data distributions can be judged by comparing the moments of each order of the samples. $\mathcal{L}_{\mathrm{DDA}}$ realizes the distribution alignment by minimizing the sample moments of the latent space pivot, in which the sample first-order moment is implicitly realized utilizing data centering.

$\mathcal{L}_{\mathrm{DDA}}$ regularization mainly considers the impact of difference in feature distribution between the source and target domain. We hope that the trained model will be insensitive to changes in feature distribution. The assumption here is that changes in data features have no impact on labels or that the distribution of labels does not change. This assumption widely exists in real applications. For example, if different sensors are used to measure the same material composition, the characteristics of the sensors are different, but the composition of the material remains unchanged.

Data difference is an effective means to achieve the stabilization of non-stationary data. $\mathcal{L}_{\mathrm{FOD}}$ minimizes the first-order difference of pivots of the source and target domain, which can effectively deal with the data redundancy caused by oversampling, and at the same time suppress the adverse effects caused by data mutations, improving the robustness of the model. The idea of data difference can be originated from the ARIMA model (Box–Jenkins method) [36]. This model first determines whether the data is stationary, and then differentiates the non-stationary data one or more times to achieve stationarity. The difference of data as a regularization can be found in the "Smoothing regularization" section of Stephen Boyd's famous book "Convex

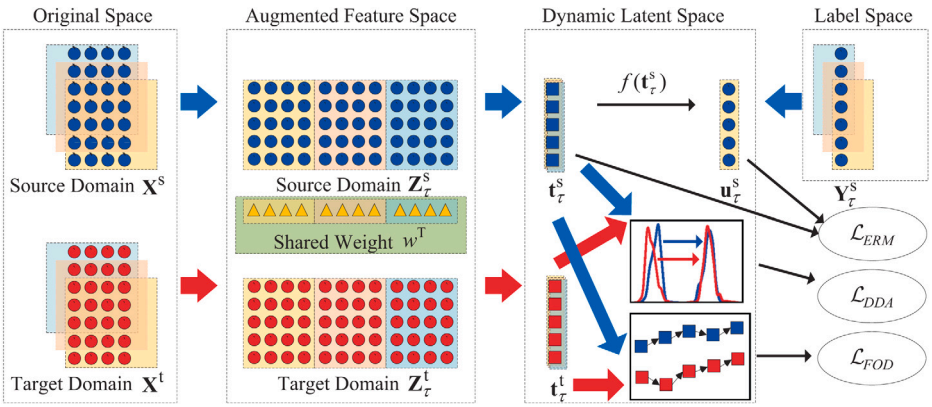

Fig. 1. The framework of dynamic transfer outer model. Transform the original time series data into the feature space through FIR matrix expansion, and find the optimal transformation axis $\boldsymbol{w}$, after projection transformation, the empirical error of the source domain sample in the latent space is minimized, and the distribution difference between the source domain and the target is minimized too.

Optimization" [37]. In Dr. Huang Biao's paper "Output relevant slow feature extraction using partial least squares" [38], the difference of data is also used as a regularization. The above literature provides theoretical support that data difference can stabilize non-stationary data.

The formula (16) defines a convex optimization problem, but the second term is not differentiable at the zero point, so a closed-form solution cannot be obtained. Note that $\frac{1}{n_S}\left(\boldsymbol{Z}_\tau^s\right)^T\boldsymbol{Z}_\tau^s-\frac{1}{n_T}\left(\boldsymbol{Z}_\tau^t\right)^T\boldsymbol{Z}_\tau^t$ is a symmetric matrix, scale it by eigenvalue decomposition and absolute value triangle inequality to obtain a semi-positive definite matrix $\boldsymbol{D}$, as follows,

$$\begin{aligned}\mathcal{L}_{\mathrm{DDA}}&=\left|\frac{1}{n_S}\left(\boldsymbol{Z}_\tau^s\boldsymbol{w}\right)^T\boldsymbol{Z}_\tau^s\boldsymbol{w}-\frac{1}{n_T}\left(\boldsymbol{Z}_\tau^t\boldsymbol{w}\right)^T\boldsymbol{Z}_\tau^t\boldsymbol{w}\right|\\&=\left|\boldsymbol{w}^T\boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{H}^T\boldsymbol{w}\right|\\&\le v_1^2|\lambda_1|+v_2^2|\lambda_2|+\cdots+v_k^2|\lambda_k|\\&=\boldsymbol{w}^T\boldsymbol{D}\boldsymbol{w}\end{aligned}\tag{17}$$

where $\frac{1}{n_S}\left(\boldsymbol{Z}_\tau^s\right)^T\boldsymbol{Z}_\tau^s-\frac{1}{n_T}\left(\boldsymbol{Z}_\tau^t\right)^T\boldsymbol{Z}_\tau^t=\boldsymbol{H}\boldsymbol{\Lambda}\boldsymbol{H}^T$, $\boldsymbol{\Lambda}=diag\left(\lambda_1,\lambda_2,\ldots,\lambda_k\right)$, $\boldsymbol{w}^T\boldsymbol{H}=[v_1,v_1,\ldots,v_k]$, $\boldsymbol{D}=\boldsymbol{H}|\boldsymbol{\Lambda}|\boldsymbol{H}^T$. Let $\mathcal{L}'_{\mathrm{DDA}}=\boldsymbol{w}^T\boldsymbol{D}\boldsymbol{w}$, $\boldsymbol{G}=\left(\dot{\boldsymbol{Z}}_\tau^s\right)^T\dot{\boldsymbol{Z}}_\tau^s+\left(\dot{\boldsymbol{Z}}_\tau^t\right)^T\dot{\boldsymbol{Z}}_\tau^t$. From formula (16) and formula (17), a structural risk upper bound can been achieved,

$$\begin{aligned}&\arg\min_{\boldsymbol{w}}\mathcal{L}_{\mathrm{ERM}}+\lambda\mathcal{L}'_{\mathrm{DDA}}+\rho\mathcal{L}_{\mathrm{FOD}}=\arg\min_{\boldsymbol{w}}\left\|\boldsymbol{Z}_\tau^s-\boldsymbol{y}_\tau^s\boldsymbol{w}^T\right\|_F\\&+\boldsymbol{w}^T\left(\lambda\boldsymbol{D}+\rho\boldsymbol{G}\right)\boldsymbol{w}\end{aligned}\tag{18}$$

deriving the above formula, and the finally obtained common optimal transformation axis is,

$$\boldsymbol{w}=\left(\boldsymbol{I}+\frac{\lambda\boldsymbol{D}+\rho\boldsymbol{G}}{\left(\boldsymbol{y}_\tau^s\right)^T\boldsymbol{y}_\tau^s}\right)^{-1}\frac{\left(\boldsymbol{Z}_\tau^s\right)^T\boldsymbol{y}_\tau^s}{\left(\boldsymbol{y}_\tau^s\right)^T\boldsymbol{y}_\tau^s}\tag{19}$$

after obtaining the common weighting vector $\boldsymbol{w}$, the input score vectors can be computed as,

$$\begin{aligned}\boldsymbol{t}_\tau^s&=\boldsymbol{Z}_\tau^s\boldsymbol{w}\\\boldsymbol{t}_\tau^t&=\boldsymbol{Z}_\tau^t\boldsymbol{w}\end{aligned}\tag{20}$$

for univariate prediction $c=1$, the source domain output score vector $\boldsymbol{u}_\tau^s$ is calculated as follows,

$$\boldsymbol{u}_\tau^s=\boldsymbol{Y}_\tau^sc=\boldsymbol{y}_\tau^s\tag{21}$$

Next, how to construct the nonlinear inner model $\boldsymbol{u}_\tau^s=f\left(\boldsymbol{t}_\tau^s\right)$ is described in detail.

3.4. Nonlinear Laplace inner model

The nonlinear inner model between input and output scores is built via kernel methods. The source and target domain score vectors $\boldsymbol{t}_\tau^s$ and $\boldsymbol{t}_\tau^t$ are mapped into the high-dimensional feature space to obtain the data $\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)$ and $\boldsymbol{\Phi}(\boldsymbol{t}_\tau^t)$. Find an optimal transformation axis $\boldsymbol{\theta}$ in the feature space to minimize the empirical error between the transformed data and output score, which is described in mathematical language as,

$$\arg\min_f\left\|f(\boldsymbol{t}_\tau^s)-\boldsymbol{u}_\tau^s\right\|_2^2=\arg\min_{\boldsymbol{\theta}}\left\|\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\boldsymbol{\theta}-\boldsymbol{u}_\tau^s\right\|_2^2\tag{22}$$

According to the reproducing kernel theory [39], the optimal transformation axis $\boldsymbol{\theta}$ in the sense of minimum mean square error must be a linear combination of all samples in the feature space, which means that $\boldsymbol{\theta}$ is located in the subspace spanned by the sample of the feature space. The labeled source domain samples constitute the feature space, and its optimal transformation axis can be expressed by the following formula,

$$\boldsymbol{\theta}=\sum_{i=1}^{n_S}\alpha_i\boldsymbol{\phi}\left(t_{\tau i}^s\right)=\left[\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\right]^T\boldsymbol{\alpha}\tag{23}$$

where $\boldsymbol{\alpha}=\left[\alpha_1,\alpha_2,\ldots,\alpha_{n_S}\right]^T$, $\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)=\left[\boldsymbol{\phi}\left(t_{\tau1}^s\right),\boldsymbol{\phi}\left(t_{\tau2}^s\right),\ldots,\boldsymbol{\phi}\left(t_{\tau n_S}^s\right)\right]^T$. According to the regularization theory, imposing certain constraints on $\boldsymbol{\theta}$ can effectively prevent the model from overfitting. Substituting the formula (23) into the formula (22) and introducing the vector norm regularization, using the kernel method, the following optimization objective formula is obtained,

$$\arg\min_{\boldsymbol{\theta}}\left\|\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\boldsymbol{\theta}-\boldsymbol{u}_\tau^s\right\|_2^2+\zeta\|\boldsymbol{\theta}\|_2^2=\arg\min_{\boldsymbol{\alpha}}\left\|\boldsymbol{K}^s\boldsymbol{\alpha}-\boldsymbol{u}_\tau^s\right\|_2^2+\zeta\boldsymbol{\alpha}^T\boldsymbol{K}^s\boldsymbol{\alpha}\tag{24}$$

where $\boldsymbol{K}^s$ is the kernel matrix constructed by source domain input score vector, $\boldsymbol{K}^s=\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\left(\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\right)^T\in\mathbb{R}^{n_S\times n_S}$, ζ is the regularization coefficient.

In addition, the above kernel-based nonlinear mapping process does not consider the sequential structure of the latent variable space after matrix augmentation. In order to further utilize the similar geometric structure of neighboring points, this paper introduces Laplacian regularization terms to retain the geometric structure. The Laplace regularization on the source domain is constructed by the following formula,

$$\begin{aligned}\mathcal{L}_{\mathrm{LAP}}^s&=\sum_{i,j=1}^{n_S}\boldsymbol{A}_{ij}^s\left(f(t_{\tau i}^s)-f(t_{\tau j}^s)\right)^2\\&=\sum_{i,j=1}^{n_S}\boldsymbol{A}_{ij}^s\left(\boldsymbol{\phi}^T(t_{\tau i}^s)\boldsymbol{\theta}-\boldsymbol{\phi}^T(t_{\tau j}^s)\boldsymbol{\theta}\right)^2\\&=\boldsymbol{\theta}^T\left(\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\right)^T\boldsymbol{L}^s\boldsymbol{\Phi}(\boldsymbol{t}_\tau^s)\boldsymbol{\theta}\\&=\boldsymbol{\alpha}^T(\boldsymbol{K}^s)^T\boldsymbol{L}^s\boldsymbol{K}^s\boldsymbol{\alpha}\end{aligned}\tag{25}$$

in the formula (25), the source domain Laplacian matrix $\boldsymbol{L}^s=\boldsymbol{D}^s-\boldsymbol{A}^s$, $\boldsymbol{A}^s$ is the source domain affinity matrix, $\boldsymbol{D}^s$ is the source domain diagonal degree matrix, its diagonal elements $\boldsymbol{D}_{ii}^s=\sum_i^{n_S}\boldsymbol{A}_{ij}^s$. The

neighbor matrix is constructed by the following temporal neighbor method,

$$\boldsymbol{A}_{ij}^{\mathrm{s}}=\begin{cases} exp\left(-\left\|t_{\tau i}^{\mathrm{s}}-t_{\tau j}^{\mathrm{s}}\right\|\right) & 0\leq|i-j|\leq\kappa, \\ 0 & otherwise. \end{cases} \tag{26}$$

where κ is a positive integer greater than zero.

Similarly, the Laplace regularization on the target domain can be constructed,

$$\mathcal{L}_{\mathrm{LAP}}^{\mathrm{t}}=\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{K}^{\mathrm{t}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{t}}\boldsymbol{K}^{\mathrm{t}}\boldsymbol{\alpha} \tag{27}$$

where $\boldsymbol{K}^{\mathrm{t}}$ is the kernel matrix constructed by the input score vector of the target domain, $\boldsymbol{K}^{\mathrm{t}}=\boldsymbol{\Phi}(\boldsymbol{t}_{\tau}^{\mathrm{t}})\left(\boldsymbol{\Phi}(\boldsymbol{t}_{\tau}^{\mathrm{s}})\right)^{\mathrm{T}}\in\mathbb{R}^{n_{\mathrm{T}}\times n_{\mathrm{S}}}$, $\boldsymbol{L}^{\mathrm{t}}\in\mathbb{R}^{n_{\mathrm{T}}\times n_{\mathrm{T}}}$ is the target domain Laplacian matrix.

On the basis of the formula (24), adding Laplace regularization (25) and (27), the final nonlinear Laplace inner model is obtained as follows,

$$\arg\min_{\boldsymbol{\alpha}}\|\boldsymbol{K}^{\mathrm{s}}\boldsymbol{\alpha}-\boldsymbol{u}_{\tau}^{\mathrm{s}}\|_2^2+\zeta\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{K}^{\mathrm{s}}\boldsymbol{\alpha}+\eta\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{K}^{\mathrm{s}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{s}}\boldsymbol{K}^{\mathrm{s}}\boldsymbol{\alpha}+\xi\boldsymbol{\alpha}^{\mathrm{T}}(\boldsymbol{K}^{\mathrm{t}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{t}}\boldsymbol{K}^{\mathrm{t}}\boldsymbol{\alpha} \tag{28}$$

where η and ξ is the regularization coefficient. The final optimal analytical solution can be solved by deriving the above formula,

$$\boldsymbol{\alpha}=\left(\boldsymbol{K}^{\mathrm{s}}+\zeta\boldsymbol{I}+\eta\boldsymbol{L}^{\mathrm{s}}\boldsymbol{K}^{\mathrm{s}}+\xi(\boldsymbol{K}^{\mathrm{s}})^{-1}(\boldsymbol{K}^{\mathrm{t}})^{\mathrm{T}}\boldsymbol{L}^{\mathrm{t}}\boldsymbol{K}^{\mathrm{t}}\right)^{-1}\boldsymbol{u}_{\tau}^{\mathrm{s}} \tag{29}$$

after obtaining the inner model regression coefficient vector $\boldsymbol{\alpha}$, the estimate of the source and target domain output scores $\hat{\boldsymbol{u}}_{\tau}^{\mathrm{s}}$ and $\hat{\boldsymbol{u}}_{\tau}^{\mathrm{t}}$ can be calculated,

$$\begin{aligned}\hat{\boldsymbol{u}}_{\tau}^{\mathrm{s}}&=\boldsymbol{K}^{\mathrm{s}}\boldsymbol{\alpha}\\ \hat{\boldsymbol{u}}_{\tau}^{\mathrm{t}}&=\boldsymbol{K}^{\mathrm{t}}\boldsymbol{\alpha}\end{aligned} \tag{30}$$

the loading vectors are,

$$\begin{aligned}\boldsymbol{p}_{\tau}^{\mathrm{s}}&=\left(\boldsymbol{Z}_{\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{s}}/\left(\boldsymbol{t}_{\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{s}}\\ \boldsymbol{p}_{\tau}^{\mathrm{t}}&=\left(\boldsymbol{Z}_{\tau}^{\mathrm{t}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{t}}/\left(\boldsymbol{t}_{\tau}^{\mathrm{t}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{t}}\end{aligned} \tag{31}$$

the residual matrixes are,

$$\begin{aligned}\boldsymbol{E}^{\mathrm{s}}&=\boldsymbol{Z}_{\tau}^{\mathrm{s}}-\boldsymbol{t}_{\tau}^{\mathrm{s}}\left(\boldsymbol{p}_{\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\\ \boldsymbol{E}^{\mathrm{t}}&=\boldsymbol{Z}_{\tau}^{\mathrm{t}}-\boldsymbol{t}_{\tau}^{\mathrm{t}}\left(\boldsymbol{p}_{\tau}^{\mathrm{t}}\right)^{\mathrm{T}}\\ \boldsymbol{F}^{\mathrm{s}}&=\boldsymbol{y}_{\tau}^{\mathrm{s}}-\hat{\boldsymbol{u}}_{\tau}^{\mathrm{s}}\end{aligned} \tag{32}$$

further calculations are performed on the residual matrix until the required A hidden variables are obtained. The final target domain prediction value can be obtained by the following formula,

$$\boldsymbol{y}_{\tau}^{\mathrm{t}}=\sum_{i=1}^{A}\hat{\boldsymbol{u}}_{\tau i}^{\mathrm{t}} \tag{33}$$

Fig. 2 displays the iterative flow chart of the NDTPLS algorithm. It demonstrates that the outer model efficiently extracts dynamic features and aligns data distribution during the input–output mapping process. This is achieved by projecting onto a common weighting vector in both the source and target domains. The inner model realizes nonlinear mapping of projection data and maintains the mapping structure, which is achieved through the construction of kernel matrix and Laplacian matrix. Loading vector enables reconstruction of the projected data to calculate the residual matrix. The whole procedure of the NDTPLS algorithm is summarized in Algorithm 1.

3.5. Hyperparameter optimization

The nonlinear dynamic transfer partial least squares method (NDT-PLS) proposed in this paper contains many hyperparameters, and different model structures can be obtained by selecting different hyperparameters. The highest model complexity can be obtained when all hyperparameters are non-zero. The algorithm retreats to the D-LSSVM-PLS algorithm when the regularization coefficients λ, ρ of the outer model and the regularization coefficients ζ, η, ξ of the inner model are all set to zero. The algorithm retreats to the DPLS algorithm when the kernel function of the inner model is selected as a linear kernel and all the above regularization coefficients are set to zero. Furthermore, the algorithm retreats to the ordinary PLS algorithm, when the delay coefficient τ is also set to zero. Therefore, although too many hyperparameters increase the difficulty of model selection and model optimization, the introduction of these hyperparameters also improves the expressive ability of the model, enabling the model to learn complex data patterns. For the process under multi-working conditions, the sample data are likely to be mixed data of various modes, the model needs to have a strong expressive ability to match the complexity of the data.

In practice, many hyperparameter tuning methods can be chosen, among which grid search, random search, heuristic algorithm, Bayesian optimization, etc. are the most widely used. Grid search tries every combination of hyperparameters in the parameter space. When the number of parameters is large, the calculation amount increases exponentially, and the calculation efficiency is low. Random search samples hyperparameters in the parameter space according to a certain probability distribution, and the calculation efficiency is improved, but each parameter search is independent, and the search capabilities depend on the number of iterations. The heuristic calculation uses a swarm intelligence algorithm to give feasible solutions to hyperparameters within an acceptable time range. The disadvantage of this type of algorithm is that it is easy to be limited to local optimal solutions. The Bayesian optimization method uses the results obtained by each step of parameter sampling (prior knowledge) to calculate the optimization direction of hyperparameters (posterior distribution), which has high computational efficiency and is not easy to fall into local optimum, so it is widely used in the field of AutoML.

Bayesian optimization is a method that uses surrogate models to optimize unknown black-box functions. First, by introducing prior knowledge of the objective function, a probabilistic surrogate model that can reflect the possible behavior of the objective function is established. Secondly, based on the obtained observation data, Bayes' theorem is run to update the model to the posterior distribution. This process not only reflects the new understanding of the objective function, but also quantifies the uncertainty of the prediction. On this basis, an acquisition function (such as expected improvement EI) is used to guide the next step of exploration. This function aims to balance exploring unknown areas to discover potential better solutions and using existing

Algorithm 1 NDTPLS algorithm.

Input: Source domain data $\boldsymbol{X}^{\mathrm{s}},\boldsymbol{y}^{\mathrm{s}}$, target domain data $\boldsymbol{X}^{\mathrm{t}}$, number of principal components A, delay factor τ, outer model tradeoff coefficient λ and ρ, inner model tradeoff coefficient ζ, η, ξ and κ
Output: target domain label $\boldsymbol{y}_{\tau}^{\mathrm{t}}$
(**Augmentation**):Construct the augmented matrix $\boldsymbol{Z}_{\tau}^{\mathrm{s}},\boldsymbol{Z}_{\tau}^{\mathrm{t}},\boldsymbol{y}_{\tau}^{\mathrm{s}}$ from the raw data $\boldsymbol{X}^{\mathrm{s}},\boldsymbol{X}^{\mathrm{t}},\boldsymbol{y}^{\mathrm{s}}$
(**Normalization**):Data normalization to zero mean and unit variance
for i *in* $[1,A]$ **do**
  (**Projection**):
    Calculate the input weight vector $\boldsymbol{w}$ by formula (19)
  (**Regression**):
    Calculate the score vectors for $\boldsymbol{Z}_{\tau}^{\mathrm{s}}$ and $\boldsymbol{Z}_{\tau}^{\mathrm{t}}$: $\boldsymbol{t}_{\tau}^{\mathrm{s}}=\boldsymbol{Z}_{\tau}^{\mathrm{s}}\boldsymbol{w}$, $\boldsymbol{t}_{\tau}^{\mathrm{t}}=\boldsymbol{Z}_{\tau}^{\mathrm{t}}\boldsymbol{w}$
    Construct kernel $\boldsymbol{K}^{\mathrm{s}}$ and $\boldsymbol{K}^{\mathrm{t}}$ using the score vectors $\boldsymbol{t}_{\tau}^{\mathrm{s}}$ and $\boldsymbol{t}_{\tau}^{\mathrm{s}}$
    Calculate inner model regression coefficient vector $\boldsymbol{\alpha}$ by formula (29)
  (**Deflation**):
    Calculate inner model estimates $\hat{\boldsymbol{u}}_{\tau}^{\mathrm{s}}=f\left(\boldsymbol{t}_{\tau}^{\mathrm{s}}\right)=\boldsymbol{K}^{\mathrm{s}}\boldsymbol{\alpha}$
    Calculate inner model estimates $\hat{\boldsymbol{u}}_{\tau}^{\mathrm{t}}=f\left(\boldsymbol{t}_{\tau}^{\mathrm{t}}\right)=\boldsymbol{K}^{\mathrm{t}}\boldsymbol{\alpha}$
    Calculate the loading vectors for $\boldsymbol{Z}_{\tau}^{\mathrm{s}}$ and $\boldsymbol{Z}_{\tau}^{\mathrm{t}}$: $\boldsymbol{p}_{\tau}^{\mathrm{s}}=\left(\boldsymbol{Z}_{\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{s}}/\left(\boldsymbol{t}_{\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{s}}$, $\boldsymbol{p}_{\tau}^{\mathrm{t}}=\left(\boldsymbol{Z}_{\tau}^{\mathrm{t}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{t}}/\left(\boldsymbol{t}_{\tau}^{\mathrm{t}}\right)^{\mathrm{T}}\boldsymbol{t}_{\tau}^{\mathrm{t}}$
    Calculate the residual matrices $\boldsymbol{E}^{\mathrm{s}}=\boldsymbol{Z}_{\tau}^{\mathrm{s}}-\boldsymbol{t}_{\tau}^{\mathrm{s}}\left(\boldsymbol{p}_{\tau}^{\mathrm{s}}\right)^{\mathrm{T}}\Rightarrow\boldsymbol{Z}_{\tau}^{\mathrm{s}}$, $\boldsymbol{E}^{\mathrm{t}}=\boldsymbol{Z}_{\tau}^{\mathrm{t}}-\boldsymbol{t}_{\tau}^{\mathrm{t}}\left(\boldsymbol{p}_{\tau}^{\mathrm{t}}\right)^{\mathrm{T}}\Rightarrow\boldsymbol{Z}_{\tau}^{\mathrm{t}}$, $\boldsymbol{F}^{\mathrm{s}}=\boldsymbol{y}_{\tau}^{\mathrm{s}}-\hat{\boldsymbol{u}}_{\tau}^{\mathrm{s}}\Rightarrow\boldsymbol{y}_{\tau}^{\mathrm{s}}$
end for
Calculate target domain label $\boldsymbol{y}_{\tau}^{\mathrm{t}}=\sum_{i=1}^{A}\hat{\boldsymbol{u}}_{\tau i}^{\mathrm{t}}$

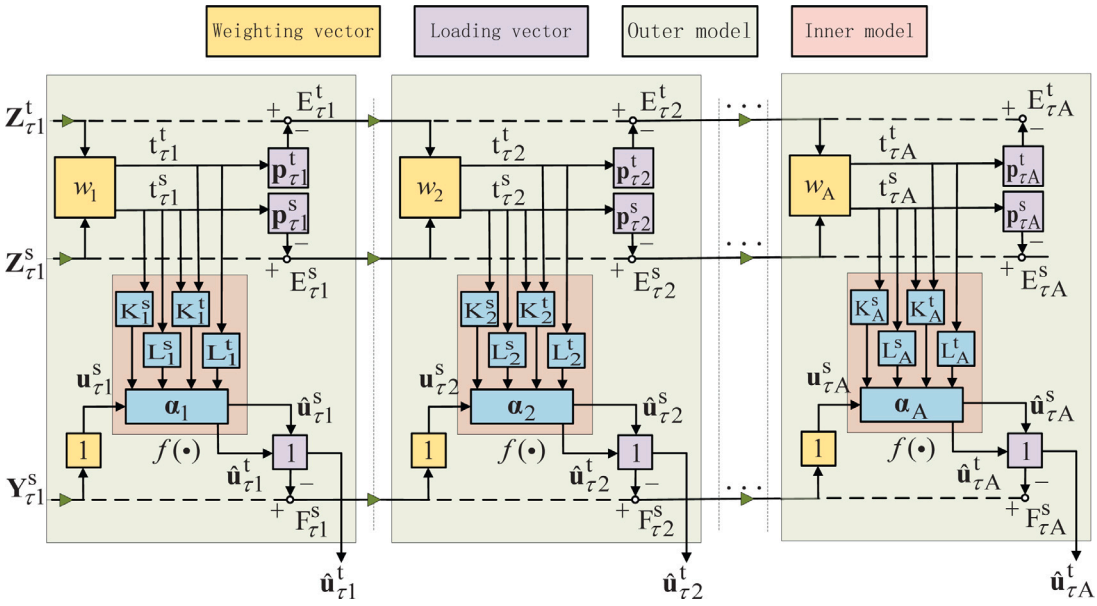

Fig. 2. The iterate flow chart of the proposed NDTPLS algorithm. The inner model is constructed by the kernel method with Laplacian regularization. The direction of the green arrow indicates the data deflation process. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2
The properties of the comparison methods.

Method	NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
Dynamic or Static	Static	Static	Dynamic	Dynamic	Static	Dynamic	Dynamic	Dynamic
Nonlinear or linear	Linear	Linear	Linear	Linear	Nonlinear	Nonlinear	Linear	Nonlinear
Transfer or not	No	Yes	No	No	No	No	Yes	Yes

information to deepen the understanding of known beneficial areas. By iterating through this process of data collection, model updating and optimization decisions, Bayesian optimization gradually approaches the global optimal solution while effectively managing the trade-off between exploration and exploitation.

In this paper, the data is divided into training set, validation set and test set. Bayesian optimization is selected as the hyperparameter tuning method, the optimal hyperparameter combination is searched on the verification set, and the obtained hyperparameter combination is applied to the test set to evaluate the effectiveness of the model. An extensive analysis of the hyperparameters will be carried out in Section 4.

4. Experiments

This paper selects the data generated by the WWTP Benchmark Simulation Model (BSM1) [40], Debutanize Column data set (DC) [41] and Sulfur Recovery Unit data set (SRU) [42] to verify the effectiveness of the proposed algorithm. Choose Nonlinear Iterative Partial Least Squares (NIPALS) [32], Dynamic Partial Least Squares (DPLS) [17], Dynamic inner Partial Least Squares (DiPLS) [18], Kernel Partial Least Squares (KPLS) [21], Domain-Invariant Iterative Partial Least Squares (DIPALS) [28], Dynamic LSSVM Partial Least Squares (D-LSSVM-PLS) [24], Dynamic Transfer Partial Least Squares (DTPLS) [29] as the comparison methods. Among the above methods, NIPALS is a static linear modeling method; DIPALS aligns the variance of hidden variables on the basis of the static model, and first introduces the transfer learning into the PLS modeling process; DPLS is a dynamic extension of the traditional PLS method; DiPLS considers the sequence structure of data but is still a linear method; The KPLS method is a kernel extension of the PLS method and is a nonlinear modeling method; D-LSSVM-PLS uses LSSVM to build the inner model, which is a dynamic nonlinear method; DTPLS not only considers the sequence structure of the inner model, but also aligns the variance of hidden variables, but it is still a linear method. The properties of the different comparison methods are summarized in Table 2. All comparison methods use Bayesian optimization to obtain hyperparameters for fair comparison.

The experiments are conducted on a high-performance hardware server with two Intel Xeon Gold 6226R Processors, 8X32G memory, and two Geforce RTX 3090 GPUs. The software is implemented using Python 3.10 in the Anaconda environment.

Three criteria are used to evaluate the performance of different models, which are mean square error (MSE), mean absolute error (MAE), and coefficient of determination (R^2),

$$\begin{aligned}
\mathrm{MSE}\left(\hat{y}_i, y_i\right) &= \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \\
\mathrm{MAE}\left(\hat{y}_i, y_i\right) &= \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \\
R^2\left(\hat{y}_i, y_i\right) &= 1 - \frac{\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2}{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}_i\right)^2}
\end{aligned} \tag{34}$$

where n is the number of test samples, $\hat{y}_i$, $\bar{y}_i$, and y_i are the predicted, mean, and measured values, respectively.

4.1. BSM1 data set

BSM1 is an activated sludge wastewater treatment model proposed by the European Organization for Scientific and Technological Cooperation in its project "COST 682". Based on the Anaerobic-Anoxic-Oxic (AAO) process flow, BSM1 is used to describe the precipitation reaction and biochemical reaction and is dedicated to the elimination of organic carbon and nitrogen. The BSM1 wastewater treatment process is shown in Fig. 3 [40], and the entire reaction process consists of an activated sludge reactor and a secondary settling tank. The activated sludge reactor includes two anaerobic sections and three aerobic sections, and

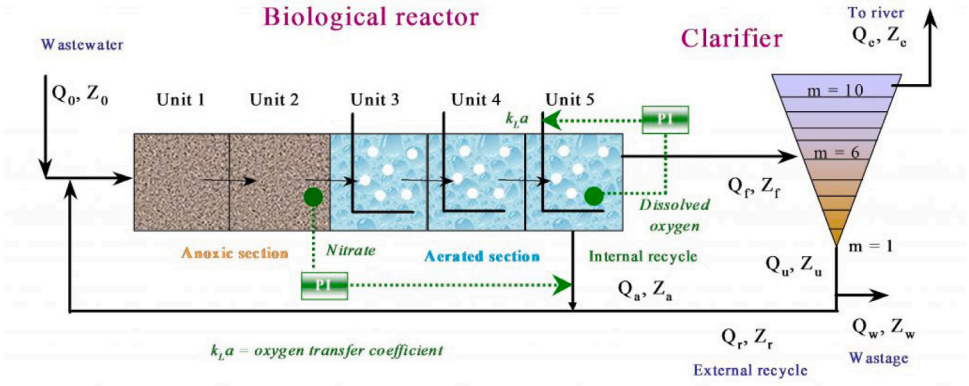

Fig. 3. BSM1:Benchmark Simulation Model 1 [40].

Table 3
The selection of process variables.

No.	Variable description	Symbol	Unit
1	Flow rate of influent	Q_in	m^3/d
2	Ammonia concentration of influent	S_{NH}_in	g N/m^3
3	Nitrate concentration of the second reactor	$S_{NO}\_reactor2$	g N/m^3
4	Dissolved oxygen concentration of the third reactor	$S_O\_reactor3$	g COD/m^3
5	Dissolved oxygen concentration of the fourth reactor	$S_O\_reactor4$	g COD/m^3
6	Total suspended solid concentration of the fourth reactor	$TSS\_reactor4$	g SS/m^3
7	Oxygen transfer coefficient of the fifth reactor	$K_La\_reactor5$	/d
8	Internal recycle rate	Q_intra	m^3/d
9	Nitrate concentration of effluent	S_{NO}_eff	g N/m^3

Table 4
Experimental results on BSM1.

Method		NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
Training set	MSE	1.217	1.215	0.386	0.677	1.017	0.180	0.286	**0.056**
	MAE	0.877	0.879	0.518	0.738	0.785	0.331	0.421	**0.186**
	R^2	0.668	0.669	0.890	0.806	0.723	0.949	0.919	**0.984**
Validation set	MSE	1.849	1.906	0.815	**0.674**	1.513	0.717	0.749	0.822
	MAE	1.044	1.052	0.677	0.678	0.948	0.599	**0.594**	0.599
	R^2	0.774	0.767	0.905	**0.921**	0.815	0.915	0.911	0.902
Test set	MSE	1.855	1.928	0.897	0.657	1.471	0.588	0.870	**0.485**
	MAE	1.055	1.073	0.654	0.673	0.916	0.537	0.590	**0.475**
	R^2	0.695	0.683	0.856	0.895	0.758	0.905	0.859	**0.921**

the sewage treated by the activated sludge reactor flows into a 4-meter-high 10-layer clarifier. The BSM1 simulation model can input data from three different weather types, to be specific, dry weather, rainy weather, and stormy weather. The data of each weather type corresponds to 14 days of data input, and the data is sampled every 15 min on average, accumulating 1345 samples.

The wastewater treatment process is a typical nonlinear dynamic system. The various biochemical reaction processes involved in it have different reaction times, and the composition and flow rate of the incoming wastewater are variable. Therefore, the collected data have a time delay and non-stationary, nonlinear, and dynamic characteristics. In this paper, the data under dry weather conditions are chosen as the training set, the data under rainy weather conditions are chosen as the test set, and the data under stormy weather are chosen as the validation set, so a multi-condition soft-sensing model for the wastewater treatment process is established to predict nitrate in the effluent concentration. The selection of input process variables refers to the selection method in [25], detailed in Table 3. Finally, the comparison results of different methods in the training, verification, and test sets are shown in Table 4. Table 5 shows the hyperparameters of different models obtained by the Bayesian optimization method. The final prediction results on the test set are visualized in Fig. 4.

From the experimental results in Fig. 4 and Table 4, it is apparent that on the BSM1 dataset, the dynamic modeling methods have better prediction accuracy than the static modeling methods. The MSE of the DiPLS method is 44.4%, 63.5%, and 64.6% lower than that of NIPALS on the training, validation, and test set respectively. The MSE of the DPLS method based on direct matrix augmentation is reduced by 68.3%, 55.9%, and 51.6% respectively compared with NIPALS. This fully illustrates the importance of obtaining data dynamics. Nonlinear modeling methods have higher prediction accuracy than linear modeling methods. The MSE of KPLS, a nonlinear modeling method based on kernel expansion, is 16.4%, 18.2%, and 20.7% lower than that of linear NIPALS respectively. The nonlinear nature of the data determines that nonlinear modeling methods are more accurate than linear ones. Simultaneously considering dynamic and nonlinear modeling methods, such as D-LSSVM-PLS and the method NDTPLS proposed in this paper, are better than pure dynamic modeling methods and pure nonlinear modeling methods. The MSE of the D-LSSVM-PLS method is 85.2%, 61.2%, and 68.3% lower than that of NIPALS respectively, and better than those of DiPLS, DPLS, and KPLS. The proposed method in this paper takes account of the non-stationary and drift characteristics of the data on the basis of the dynamics and nonlinearity, so it obtains the best prediction results. The MSE is reduced by 73.9%, 26.2%, and 17.5% compared to NIPALS, DiPLS, and D-LSSVM-PLS on the test set, respectively.

At the same time, comparing the data in Table 4, it can be seen that, traditional dynamic modeling methods such as DPLS and DiPLS

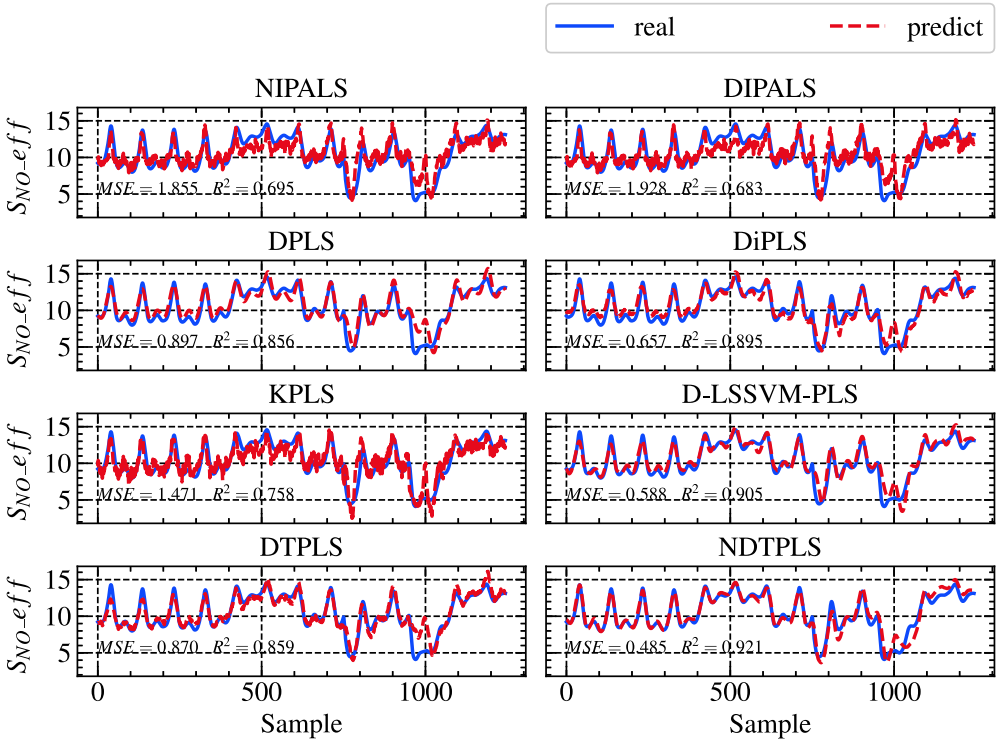

Fig. 4. Visualization results of the comparison methods on the BSM1 test set.

Table 5
List of hyperparameters obtained by Bayesian optimization on the BSM1 validation set.

Hyperparameter	NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
Latent variable dimension A	7	8	2	1	6	4	2	7
Time-lagged coefficient τ	–	–	89	87	–	60	68	65
Gaussian kernel width γ	–	–	–	–	0.013	5.754e−05	–	4.806e−05
Distribution alignment regularization λ	–	9.177e+07	–	–	–	–	4.555e−03	1.597+04
First order difference regularization ρ	–	–	–	–	–	–	5.317e−04	3.829e+04
L2 regularization ζ	–	–	–	–	–	0.010	–	1.657-05
Source lapace regularization η	–	–	–	–	–	–	–	5.598e−03
Target lapace regularization ξ	–	–	–	–	–	–	–	8.001e−04
Adjacency parameter κ	–	–	–	–	–	–	–	1332

methods are prone to overfitting in the process of parameter optimization on the verification set, making the model fit the data well on the verification set. But when the data distribution of the test set changes, the optimal model obtained on the validation set degenerates. This is because the traditional dynamic modeling methods DPLS and DiPLS have no mechanism to prevent the model from overfitting, so the generalization ability is poor. The D-LSSVM-PLS method, in the nonlinear inner model, is designed with L2 regularization, so the generalization ability is improved. In the method of this paper, both inner and outer models are designed with regularization parameters. While ensuring the accuracy of prediction, it effectively improves the generalization ability of the model, so that the prediction ability of the final model on the training set, verification set, and test set has been consistently improved.

From the parameters obtained from the Bayesian optimization in Table 5, it can be seen that the optimal hidden variable dimension of the DPLS and DiPLS methods is small (2 and 1, respectively), but requires a larger delay coefficient (89 and 87). The hidden variable dimension of the D-LSSVM-PLS method is in the middle (4), and the lag coefficient is smaller (60) than the DPLS and DiPLS methods. The hidden variable dimension (2) and lag coefficient (68) of DTPLS are both small. The method NDTPLS proposed in this paper has a relatively small delay coefficient (68) but requires a large hidden variable dimension (7). The larger the hidden variable dimension, the more components are extracted from the data, and the more data information is used. The larger the lag coefficient, the greater the expansion of the data, which means an increase in the amount of calculation.

4.2. DC data set

The Debutanizer Column is a fractionation column used to recover light gases (C1–C4) and liquefied petroleum gas (LPG) from the overhead distillate prior to the production of light naphtha in the refining process. The DC data set is a commonly used data set in the field of industrial soft sensors, which describes a real industrial distillation process. Since the DC data set and the SRU data set to be introduced later are often found in the literature, due to space limitations, the detailed description of these data sets will not be repeated in this article. The DC data set contains seven input process variables and one output variable (butane concentration), with a total of 2394 samples. The first 1100 data are chosen as the training set, the 1100th–1600th data as the verification set, and the rest as the test set. The final prediction results of different methods on the training set, verification set, and test set are shown in the Table 6. Table 7 shows the hyperparameters of the different models obtained by Bayesian optimization on the validation set. The prediction results of the final test set are visualized in Fig. 5.

It can be seen from the experimental results of Fig. 5 and Table 6 that for the DC data set, the dynamic modeling method DPLS and

Table 6
Experimental results on DC.

Method		NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
Training set	MSE	0.015	0.021	0.005	0.005	0.019	0.004	0.004	**0.002**
	MAE	0.073	0.091	0.053	0.052	0.080	0.049	0.049	**0.037**
	R^2	0.200	−0.168	0.749	0.742	−0.029	0.790	0.771	**0.880**
Validation set	MSE	0.025	0.023	0.004	**0.003**	0.021	0.004	0.004	0.004
	MAE	0.125	0.118	0.046	**0.040**	0.113	0.045	0.052	0.049
	R^2	−0.239	−0.138	0.713	**0.792**	−0.018	0.738	0.691	0.729
Test set	MSE	0.039	0.043	0.013	0.012	0.039	0.011	0.010	**0.010**
	MAE	0.154	0.158	0.089	0.085	0.149	0.084	0.081	**0.078**
	R^2	−0.030	−0.135	0.589	0.624	−0.045	0.650	0.667	**0.692**

Table 7
List of hyperparameters obtained by Bayesian optimization on the DC validation set.

Hyperparameter	NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
latent variable dimension A	4	1	3	4	1	5	3	5
time-lagged coefficient τ	–	–	29	26	–	24	27	20
Gaussian kernel width γ	–	–	–	–	0.003	2.784e−04	–	2.391e−05
distribution alignment regularization λ	–	9.99e+08	–	–	–	–	1.478e+09	6.765e+03
first order difference regularization ρ	–	–	–	–	–	–	6.631e+06	5.901e+04
L2 regularization ζ	–	–	–	–	–	7.660	–	2.546e−05
source lapace regularization η	–	–	–	–	–	–	–	6.215e−05
target lapace regularization ξ	–	–	–	–	–	–	–	2.776e−04
adjacency parameter κ	–	–	–	–	–	–	–	702

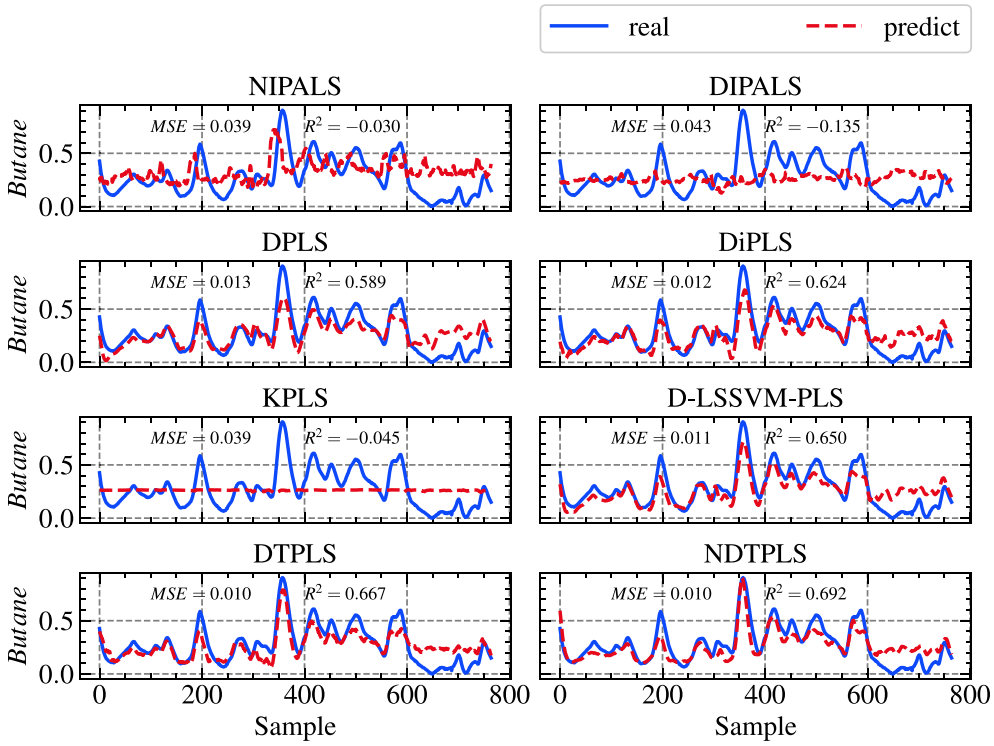

Fig. 5. Visualization results of the comparison methods on the DC test set.

DiPLS have achieved better prediction results, with MSE of 0.013 and 0.012 on the final test set, which are 66.7% and 69.2% lower than the baseline method NIPALS, respectively. However, DPLS and DiPLS lack an effective mechanism to prevent overfitting, which makes them achieve good results on the validation set, but have poor results on the final test set. The MSE of DiPLS on the validation set is 0.003, which is the lowest among all methods, but its results on the test set are not as good as D-LSSVM-PLS, DTPLS and NDTPLS, which indicates that the overfitting mechanism in the model is very important. The kernel function-based nonlinear modeling method KPLS shows a 16% decrease in the MSE metric on the validation set compared to the linear NIPALS method, but does not work as well on the test set, due to the fact that the model is trapped in a local optimum and thus does not generalize enough. The D-LSSVM-PLS method do not show sufficient advantage over DTPLS due to the insignificant non-linearity of the data. The DIPALS method cannot improve prediction performance, although the method attempts to reduce the distribution difference between the training set and the test set. Because the dynamic characteristics of data in this DC data set are more significant than the differences in data distribution. Only by prioritizing the acquisition of dynamic characteristics and then considering the distribution differences of the data can the prediction performance of the model be further improved. The NDTPLS method proposed in this paper takes into account both the dynamic properties and nonlinearity of the process, while introducing

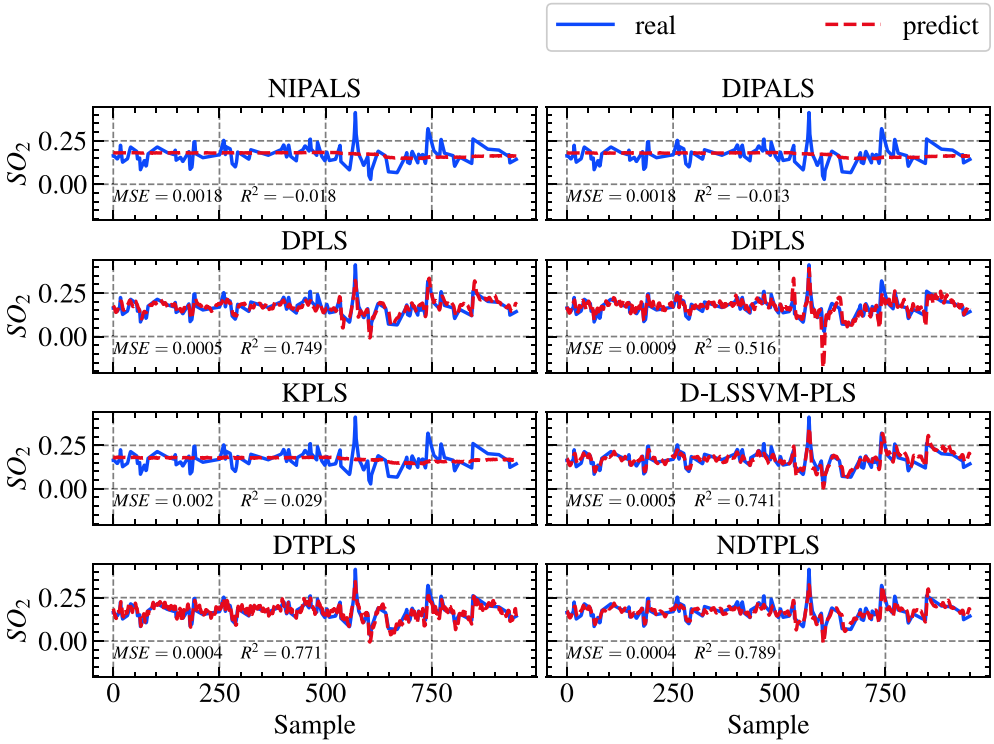

Fig. 6. Visualization results of the comparison methods on the SRU test set.

a mechanism to avoid overfitting the model, resulting in good accuracy and generalization.

4.3. SRU data set

The SRU data set contains five observed variables and two response variables. In this paper, SO2 concentration is selected as the response variable, the first 2000 data are used as the training set, the 2000th–3000th data are used as the verification set, and the 3000th–4000th data are used as the test set. The list of hyperparameters obtained by performing Bayesian optimization on the validation set is shown in the Table 9. The prediction results are shown in Fig. 6 and Table 8.

From the experimental results, it appears that the situation for the SRU dataset is similar to that of the DC. The nonlinear modeling method KPLS has improved over the linear modeling method NIPALS, but not much, R^2 increased from −0.018 to 0.029. All dynamic modeling methods achieve significant improvements in consistency. The DPLS method is superior to DiPLS in terms of accuracy. DIPALS fail to improve the model's predictive performance. The modeling methods D-LSSVM-PLS, DTPLS, and NDTPLS proposed in this paper, which consider the dynamic and nonlinear characteristics of the data at the same time, have achieved high prediction accuracy. In addition to the above dynamic and nonlinear characteristics, the method proposed in this paper also considers the model degradation caused by the difference in data distribution, so it has the best prediction results. In contrast, the DiPLS method only considers the dynamic characteristics of the inner model, and its final prediction effect is poor.

The experimental results on the above DC and SRU data set once again show that the dynamic and nonlinear modeling method can effectively improve the modeling accuracy of the soft sensor model. At the same time, the regularization technology introduced in this article ensures that the parameter optimization process does not fall into a local optimal solution.

4.4. Hyperparameter analysis

The method proposed in this article adds multiple regularization terms to the inner and outer models, which makes the model more complex. For the soft sensor modeling problem with complex multi-working conditions, the complexity of the data is high, and a corresponding high-complexity model is required to match it. However, high-complexity models also bring a series of problems, such as increased calculations and difficulties in hyperparameter optimization. In this paper, the Bayesian optimization framework [43] is used to optimize hyperparameters, the regression determination coefficient R^2 is used as the optimization goal, and TPE (Tree-structured Parzen Estimator) sampling is performed on the parameter search space. This Bayesian optimization method uses the optimization results of each step to optimize the search space, which effectively improves the search efficiency. Fig. 7 is the optimization history graph of Bayesian optimization on the verification set of the three data sets used in this paper. 100 trials are conducted on the WWTP dataset and 200 trials on the DC and SRU datasets respectively, and finally obtained the optimal combination of hyperparameters.

Through the above Bayesian optimization process, different hyperparameter combinations and their corresponding optimization target R^2 values can be obtained. By analyzing this data, the effect of various hyperparameters on different datasets can be clarified. The importance plot of the model hyperparameters was obtained by conducting a functional ANOVA [44] on the optimized data, as illustrated in Fig. 8. It can be seen from Fig. 8 that for the WWTP data set, the importance of the target domain Laplacian regularization parameter ξ(xi) and the source domain Laplacian regularization parameter η(eta) rank first and fifth respectively, indicating that the inner model regularization parameters are effective. The first-order difference regularity ρ(rho) and the distribution difference regularity λ(lambda) rank second and third in importance, which shows the effectiveness of the regularization parameters of the outer model. For the DC data set, the importance of the target domain Laplace regular parameter ξ(xi) and the neighbor parameter κ(kappa) rank first and second respectively, and the

Table 8
Experimental results on SRU.

Method		NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
Training set	MSE	0.0033	0.0033	0.0015	0.0016	0.0033	0.0013	**0.0011**	0.0014
	MAE	0.036	0.036	0.024	0.024	0.036	0.023	**0.021**	0.023
	R^2	0.034	0.034	0.565	0.545	0.038	0.622	**0.685**	0.576
Validation set	MSE	0.0030	0.0030	0.0007	0.0018	0.0030	**0.0006**	0.0007	**0.0006**
	MAE	0.034	0.033	0.020	0.030	0.034	0.020	0.020	**0.019**
	R^2	0.053	0.052	0.779	0.448	0.065	**0.804**	0.771	0.803
Test set	MSE	0.0018	0.0018	0.0005	0.0009	0.0017	0.0005	**0.0004**	**0.0004**
	MAE	0.031	0.031	0.016	0.020	0.031	0.017	0.016	**0.015**
	R^2	−0.018	−0.013	0.749	0.516	0.029	0.741	0.771	**0.789**

Table 9
List of hyperparameters obtained by Bayesian optimization on the SRU validation set.

Hyperparameter	NIPALS	DIPALS	DPLS	DiPLS	KPLS	D-LSSVM-PLS	DTPLS	NDTPLS
Latent variable dimension A	5	4	5	3	5	5	2	5
Time-lagged coefficient τ	–	–	13	49	–	12	12	12
Gaussian kernel width γ	–	–	–	–	0.003	2.992e−05	–	4.889e−05
Distribution alignment regularization λ	–	9.07e+08	–	–	–	–	6.677e+08	0.559
First order difference regularization ρ	–	–	–	–	–	–	2.341e+03	0.006
L2 regularization ζ	–	–	–	–	–	0.002	–	0.0798
Source lapace regularization η	–	–	–	–	–	–	–	1.794e−04
Target lapace regularization ξ	–	–	–	–	–	–	–	7.302e−04
Adjacency parameter κ	–	–	–	–	–	–	–	284

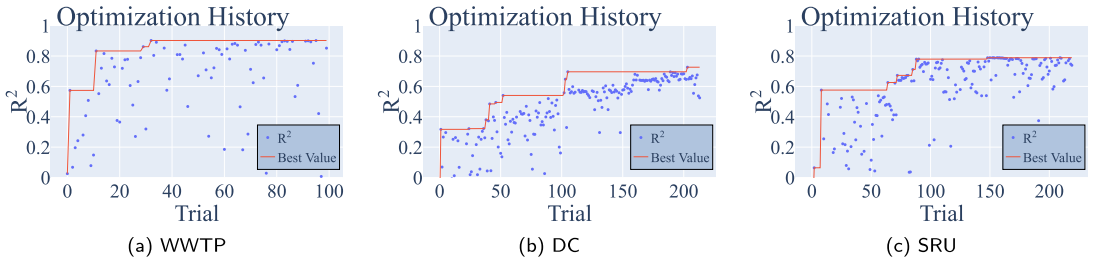

Fig. 7. Optimization history plot.

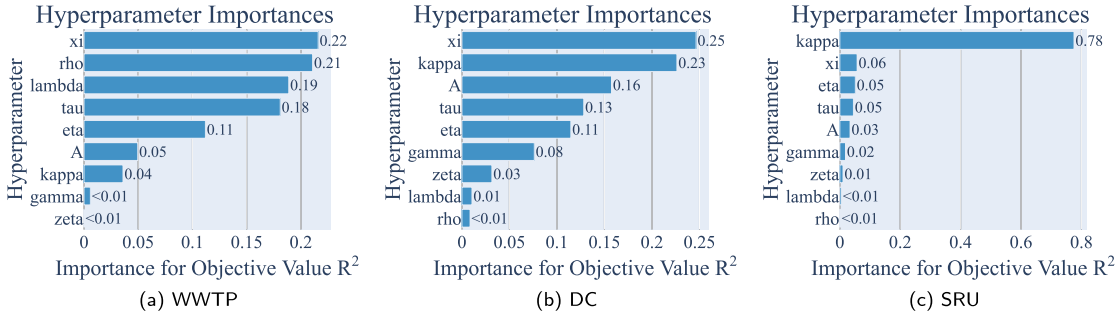

Fig. 8. Hyperparameter importances plot.

source domain Laplace regular parameter η(eta) ranks fifth, which also shows the effectiveness of the inner model parameters. Hidden variable dimension A and lag coefficient τ(tau) rank third and fourth in importance. For the SRU dataset, the importance of the neighbor parameter κ(kappa) ranks first, and other parameters are not significant.

Slice graphs of different hyperparameters under different data sets can be obtained according to the results of Bayesian optimization. Through the slice graphs, a visual display of the parameter distribution and optimization goals of different hyperparameters can be obtained. The slice diagram of the hyperparameters under the WWTP data set is shown in Figure A.1, It can be seen from the figure that a larger hidden variable dimension A and a lag parameter τ can obtain a higher optimization goal, but the lag parameter τ cannot be too large, otherwise it is easy to cause overfitting. The regular parameters λ and ρ of the outer model need to be selected within a suitable interval. The selection of parameters ζ and γ should not be too large. The selection of regular parameters ξ and κ of the inner model should not be too small.

The slice diagram of the hyperparameters under the DC dataset is shown in Figure A.2. It can be seen from the figure that the higher the hidden variable dimension A is, the easier it is to obtain a higher optimization target value. This is because there are more hidden variable dimensions and more useful information can be extracted. The lag parameter τ should not be too large or too small. If it is too large, it will easily cause overfitting, and if it is too small, the historical information retained is not enough. Smaller distribution difference regularization parameter λ and larger difference regularization parameter ρ can make the model obtain higher prediction performance. The selection of parameters ζ and γ should not be too large. The larger the selection of the inner model parameters η, ξ, and κ, the better the results are,

indicating that the maintenance of the inner model manifold structure has a positive impact on the prediction results.

The slice diagram of the hyperparameters under the SRU dataset is shown in Figure A.3. It can be seen from the figure that a larger hidden variable dimension A leads to better optimization results. But the selection of lag parameter τ should not be too large. At the same time, the regular parameters λ and ρ of the outer model should not be too large. The value of the parameter ζ needs to be selected within a suitable interval. The parameter γ and the inner model neighbor parameter κ tend to choose smaller values.

The experimental results of the above slice graphs show that the selection of regularization parameters of the inner and outer models of different data sets is different. This is due to the different dynamics of different data sets and distinct time constants. For data sets with large time constants, it is easy to choose large lag time and neighbor parameters. On the contrary, for data sets with small time constants, it is easy to choose the smaller lag time and neighbor parameters. Nevertheless, under the framework of Bayesian optimization, we can always obtain the optimal set of hyperparameters.

5. Conclusion

To handle the dynamic, nonlinear, and multi-condition characteristics of data in modern industrial processes, a new nonlinear dynamic transfer partial least squares algorithm has been proposed in this article. The traditional PLS framework of outer projection has been changed by minimizing the reconstruction error of the source domain. The paper proves that minimizing the reconstruction error is equivalent to maximizing the covariance of latent variables, but an empirical error upper bound can be found.

The algorithm adds distribution difference regularization and first-order difference regularization in the process of outer projection and adds Laplacian regularization in the process of inner nonlinear mapping. The hyperparameters are solved by a Bayesian optimization procedure. The experimental results show that the variance of the latent variables can be reduced, and the difference in distribution between the source and target domains can be decreased too. The first latent variable can capture the trend information of the data and thereafter remove the trend information, the remaining latent variables become more stable.

The above conclusion suggests that the proposed algorithm is well-suited for multiple working conditions and non-stationary processes. However, further improvement is still necessary. The inner model utilizes the kernel method to extract nonlinear features, but the computational efficiency of the algorithm is hindered by the need to calculate the inverse of the matrix. Moreover, the model was developed offline, and its applicability in an online environment requires additional research into the update mechanism. To overcome the current limitations, the study will prioritize the development of efficient nonlinear models and recursive model update mechanisms in the future.

CRediT authorship contribution statement

Zhijun Zhao: Writing – original draft, Software, Methodology, Conceptualization. **Gaowei Yan:** Writing – review & editing, Supervision, Methodology. **Mifeng Ren:** Resources, Investigation. **Lan Cheng:** Methodology, Investigation. **Rong Li:** Validation, Investigation. **Yusong Pang:** Validation, 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.

Acknowledgments

The authors are grateful for the financial support from the National Natural Science Foundation of China (61973226, 62073232, 62003233), Natural Science Foundation of Shanxi Province, China (20210302123189), Shanxi Province Major Special Program of Science and Technology, China (202201090301013, 202201090301001), and Gemeng Science and Technology Innovation Foundation Project (2022-05).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.isatra.2024.08.002.

References

[1] Yang T, Yi X, Lu S, Johansson KH, Chai T. Intelligent manufacturing for the process industry driven by industrial artificial intelligence. Engineering 2021;7:1224–30.
[2] Luo Y, Zhang X, Kano M, Deng L, Yang C, Song Z. Data-driven soft sensors in blast furnace ironmaking: a survey. Front Inf Technol Electron Eng 2023;24:327–54.
[3] Sun Q, Ge Z. A survey on deep learning for data-driven soft sensors. IEEE Trans Ind Inf 2021;17:5853–66.
[4] Zhao Y, Ding B, Zhang Y, Yang L, Hao X. Online cement clinker quality monitoring: A soft sensor model based on multivariate time series analysis and cnn. ISA Trans 2021;117:180–95.
[5] Zhang T, Yan G, Li R, Xiao S, Ren M, Cheng L. An online transfer kernel recursive algorithm for soft sensor modeling with variable working conditions. Control Eng Pract 2023;141:105726.
[6] Zhang J, Tang Z, Xie Y, Ai M, Gui W. Convolutional memory network-based flotation performance monitoring. Miner Eng 2020;151:106332.
[7] Xia H, Tang J, Aljerf L. Dioxin emission prediction based on improved deep forest regression for municipal solid waste incineration process. Chemosphere 2022;294:133716.
[8] Zhao Z, Yan G, Li R, Xiao S, Wang F, Ren M, Cheng L. Instance transfer partial least squares for semi-supervised adaptive soft sensor. Chemometr Intell Lab Syst 2024;245:105062.
[9] Geladi P, Kowalski BR. Partial least-squares regression: a tutorial. Anal Chim Acta 1986;185:1–17.
[10] Huang G-B, Zhu Q-Y, Siew C-K. Extreme learning machine: theory and applications. Neurocomputing 2006;70:489–501.
[11] Hallin M. Gauss–Markov theorem in statistics. Online: Wiley StatsRef Stat. Ref.; 2014.
[12] Yang C, Liu Q, Liu Y, Cheung Y-M. Transfer dynamic latent variable modeling for quality prediction of multimode processes. IEEE Trans Neural Netw Learn Syst 2023;1–14.
[13] Sun C, Zhang Y, Huang G, Liu L, Hao X. A soft sensor model based on long & short-term memory dual pathways convolutional gated recurrent unit network for predicting cement specific surface area. ISA Trans 2022;130:293–305.
[14] Zhao C. Perspectives on nonstationary process monitoring in the era of industrial artificial intelligence. J Process Control 2022;116:255–72.
[15] Zhao C, Yu W, Gao F. Data analytics and condition monitoring methods for nonstationary batch processes — Current status and future. Acta Automat Sinica 2020;46:2072–91.
[16] Zheng J, Zhao C, Gao F. Retrospective comparison of several typical linear dynamic latent variable models for industrial process monitoring. Comput Chem Eng 2022;157:107587.
[17] Ricker NL. The use of biased least-squares estimators for parameters in discrete-time pulse-response models. Ind Eng Chem Res 1988;27:343–50.
[18] Dong Y, Qin SJ. Regression on dynamic pls structures for supervised learning of dynamic data. J Process Control 2018;68:64–72.
[19] Yang L, Liu Y, Yang G, Peng S-T. Dynamic monitoring and anomaly tracing of the quality in tobacco strip processing based on improved canonical variable analysis and transfer entropy. Math Biosci Eng 2023;20:15309–25.
[20] Kong XY, Cao ZH, An QS, Xu ZY, Luo JY. Review of partial least squares linear models and their nonlinear dynamic expansion models. Control Decis 2018;33:1537–48.
[21] Rosipal R, Trejo LJ. Kernel partial least squares regression in reproducing kernel hilbert space. J Mach Learn Res 2001;2:97–123.
[22] Bennett KP, Embrechts MJ. An optimization perspective on kernel partial least squares regression. Nato Sci Ser Sub Ser III Comput Syst Sci 2003;190:227–50.
[23] Qin SJ, McAvoy TJ. Nonlinear pls modeling using neural networks. Comput Chem Eng 1992;16:379–91.
[24] Lv Y, Liu J, Yang T. Nonlinear pls integrated with error-based lssvm and its application to nox modeling. Ind Eng Chem Res 2012;51:16092–100.

[25] Yang C, Zhang Y, Huang M, Liu H. Adaptive dynamic prediction of effluent quality in wastewater treatment processes using partial least squares embedded with relevance vector machine. J Clean Prod 2021;314.
[26] Liu H, Yang C, Carlsson B, Qin SJ, Yoo C. Dynamic nonlinear partial least squares modeling using gaussian process regression. Ind Eng Chem Res 2019;58:16676–86.
[27] Wang K, Zhou W, Mo Y, Yuan X, Wang Y, Yang C. New mode cold start monitoring in industrial processes: A solution of spatial–temporal feature transfer. Knowl-Based Syst 2022;248:108851.
[28] Nikzad-Langerodi R, Zellinger W, Saminger-Platz S, Moser BA. Domain adaptation for regression under beer–lambert's law. Knowl-Based Syst 2020;210:106447.
[29] Zhao Z, Yan G, Ren M, Cheng L, Zhu Z, Pang Y. Dynamic transfer partial least squares for domain adaptive regression. J Process Control 2022;118:55–68.
[30] Gao X, Liu Y, Xie Y, Huang D. Novel multimodal data fusion soft sensor modeling framework based on meta-learning networks for complex chemical process. IFAC-PapersOnLine 2022;55:839–44.
[31] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2009;22:1345–59.
[32] Wold H. Soft modelling by latent variables: the non-linear iterative partial least squares (nipals) approach. J Appl Probab 1975;12:117–42.
[33] Wold S, Kettaneh-Wold N, Skagerberg B. Nonlinear pls modeling. Chemometr Intell Lab Syst 1989;7:53–65.
[34] Yang C, Yang C, Li J, Li Y, Yan F. Forecasting of iron ore sintering quality index: A latent variable method with deep inner structure. Comput Ind 2022;141:103713.
[35] Vapnik VN, Chervonenkis AY. On the uniform convergence of relative frequencies of events to their probabilities. In: Meas. complex. festschrift alexey chervonenkis. Springer; 2015, p. 11–30.
[36] Box GE, Jenkins GM, Reinsel GC, Ljung GM. Time series analysis: forecasting and control. John Wiley & Sons; 2015.
[37] Boyd SP, Vandenberghe L. Convex optimization. Cambridge University Press; 2004.
[38] Chiplunkar R, Huang B. Output relevant slow feature extraction using partial least squares. Chemometr Intell Lab Syst 2019;191:148–57.
[39] Belkin M, Niyogi P, Sindhwani V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. J Mach Learn Res 2006;7:2399–434.
[40] Alex J, Benedetti L, Copp J, Gernaey K, Jeppsson U, Nopens I, Pons M, Rieger L, Rosen C, Steyer J, et al. Benchmark simulation model (1) (bsm1). In: Rep. by IWA taskgr. benchmarking control strateg. wWTPs. vol. 1, 2008.
[41] Fortuna L, Graziani S, Xibilia MG. Soft sensors for product quality monitoring in debutanizer distillation columns. Control Eng Pract 2005;13:499–508.
[42] Fortuna L, Rizzo A, Sinatra M, Xibilia M. Soft analyzers for a sulfur recovery unit. Control Eng Pract 2003;11:1491–500.
[43] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: A next-generation hyperparameter optimization framework. In: Proc. 25th ACM SIGKDD int. conf. knowl. discov. & data min. 2019, p. 2623–31.
[44] Hutter F, Hoos H, Leyton-Brown K. An efficient approach for assessing hyperparameter importance. In: Int. conf. mach. learn. PMLR; 2014, p. 754–62.