

# Hybrid method integrating machine learning and particle swarm optimization for smart chemical process operations

Haoqin Fang<sup>1\*</sup>, Jianzhao Zhou<sup>1\*</sup>, Zhenyu Wang<sup>1</sup>, Ziqi Qiu<sup>1</sup>, Yihua Sun<sup>2</sup>, Yue Lin<sup>1</sup>, Ke Chen<sup>1</sup>, Xiantai Zhou (✉)<sup>1</sup>,  
Ming Pan (✉)<sup>1</sup>

<sup>1</sup> School of Chemical Engineering and Technology, Sun Yat-Sen University, Zhuhai 519082, China

<sup>2</sup> School of Mathematics, Sun Yat-Sen University, Zhuhai 519082, China

© Higher Education Press 2021

**Abstract** Modeling and optimization is crucial to smart chemical process operations. However, a large number of nonlinearities must be considered in a typical chemical process according to complex unit operations, chemical reactions and separations. This leads to a great challenge of implementing mechanistic models into industrial-scale problems due to the resulting computational complexity. Thus, this paper presents an efficient hybrid framework of integrating machine learning and particle swarm optimization to overcome the aforementioned difficulties. An industrial propane dehydrogenation process was carried out to demonstrate the validity and efficiency of our method. Firstly, a data set was generated based on process mechanistic simulation validated by industrial data, which provides sufficient and reasonable samples for model training and testing. Secondly, four well-known machine learning methods, namely, K-nearest neighbors, decision tree, support vector machine, and artificial neural network, were compared and used to obtain the prediction models of the processes operation. All of these methods achieved highly accurate model by adjusting model parameters on the basis of high-coverage data and properly features. Finally, optimal process operations were obtained by using the particle swarm optimization approach.

**Keywords** smart chemical process operations, data generation, hybrid method, machine learning, particle swarm optimization

## 1 Introduction

The chemical processing industry manufacturing products by mixing, separating, forming, and chemical reactions, is being promoted for its greatly important contribution to economic development, providing high quality products to many sectors, such as building, transportation, construction and health [1]. It is also the largest energy consumer among industrial sectors with 1078 million tons of oil equivalent of energy consumption in the year 2016 [2]. Unlike general artificial intelligence which builds computerized systems possessing intelligence like humans, industrial intelligence is more concerned with the application of technologies to build a smart system that can adapt to the changes in industrial environment. This requires process optimization by adjusting key operating parameters to maximize/minimize one or more of the process specifications (such as profit/cost and efficiency/consumption), while keeping all others within their constraints. It has been proved that optimal operations can influence the economic and energy efficiencies of the whole process by 24% [3].

Many optimization strategies have been reported for process operations, which can be mainly categorized as mechanism-based methods and data-driven methods [4]. The mechanism-based methods reflect the real material balance, equilibrium, summation, enthalpy balance, and hydraulic performance in each detailed unit and through the whole process. They are generally used for conceptual design, and have been implemented in many commercial simulators, such as Aspen Plus, Aspen Hysys, Pro-II, and gPROMS [5]. Recent studies have focused on linking external optimization algorithms to the above commercial simulators. Ibrahim et al. used a genetic algorithm to optimize a crude oil processing system, and defined the operational variables of the process in Aspen Hysys [6]. However, the mechanism-based models usually suffer

Received December 10, 2020; accepted January 19, 2021

E-mails: zhouxtai@mail.sysu.edu.cn (Zhou X),  
ming.pan@idtgz.com (Pan M)

\*These authors contributed equally to this work.

from the computational difficulty of solving complex nonlinear problems caused by the first-principle calculations [7]. In order to reduce the complexity of full-rigorous models, simplified models have been proposed. In some commercial software, e.g., PIMS (Aspen Tech) and RPMS (Honeywell), highly nonlinear models are simplified using linear or bi-linear constraints, and solved with the successive and sequential linear programming method [8]. Although the simplified models have been widely applied into industry, they may still cause a large deviation because of assuming approximate or near linearity [9].

Compared to the mechanism-based models, data-driven models presenting the relationship between input variables and output variables as “black boxes” can reduce the complexity. K-nearest neighbors (KNN), support vector machine (SVM), decision tree (DT), and artificial neural network (ANN) are the most commonly-used machine learning algorithms for data-driven models [10]. KNN is capable of addressing the issues arising from nonlinearity and insufficient training data, and has been applied in different areas such as pattern recognition [11], data mining [12], and outlier detection [13], to improve process operations [14]. SVM models combined with a genetic algorithm optimizer had been used for controlling production quality [15] and optimizing equipment structure [16], especially for minimizing energy consumption up to 43% in carbon fiber industry [17]. Both DT [18] and ANN [19,20] can predict the output measures very well in pyrolysis reaction networks, but DT is much faster to be trained than ANN, as it is less complicated and can be constructed by practitioners without deep understanding required for training ANN [21]. However, ANN shows prospective application in the correlations between molecular structures and properties [22–24]. Recently, Su et al. [25] developed innovative architecture of deep neural network combining tree structured long short-term memory network and back-propagation neural network to provide an intelligent tool to predict properties in the design or synthesis of chemical, pharmaceutical, biochemical products and processes. Further, for solving the optimization problems on chemical engineering effectively and efficiently, Schweidtmann et al. [26] maximized the net power generation in an organic Rankine cycles by deterministic global process optimization based on ANNs, and Chandrasekaran and Tamang [27] employed ANN and particle swarm optimization (PSO) technique to minimize the machining time in turning of Al-SiCp metal matrix composites. Besides, random forest and genetic algorithm were integrated to optimize the operating conditions of chocolate chip cookie production [28].

As discussed above, existing works mostly focus on developing one kind of machine learning method while rarely consider the four machine learning algorithms in the same chemical process to compare and analyze their performances. It is also difficult to obtain enough datasets for model training, as a practical chemical process is

usually operated continuously and steadily in a small region. Thus, this paper presents an efficient dataset generation strategy based on industrial data validation and commercial software simulation at first. A hybrid framework integrating machine learning algorithms and PSO is then proposed to achieve smart process operations. In this hybrid framework, the models of chemical processes are trained by using four well-known machine learning algorithms, namely KNN, DT, SVM, and ANN, and the most efficient model can be selected and combined with PSO for further process optimization.

---

## 2 Experimental

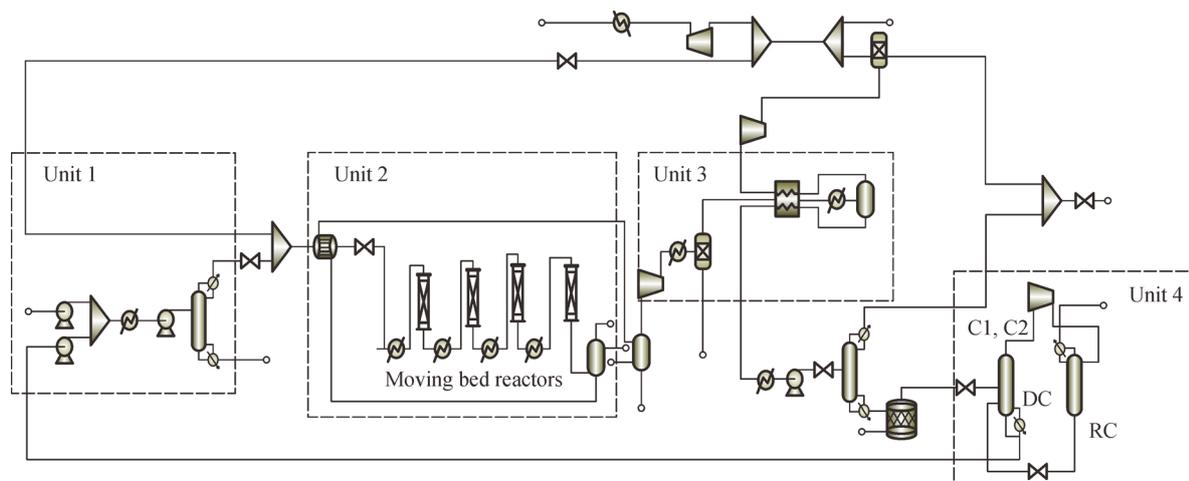
### 2.1 Data generation of chemical process operation

Machine learning comprises models that learn from existing data. These data require preprocessing to identify missing and spurious elements. Since a chemical process must run in a steady state, the data collected from industry are only in a limited operation region, which will affect the model prediction. Many specialized simulation software tools (such as Aspen Plus and Hysys) have been utilized for adjusting chemical process operations [29]. This can avoid the risk of operating the real processes in an unexpected region. In this paper, an industrial propane dehydrogenation (PDH) process is carried out as the case study, where Aspen Plus simulation process is built and validated by the industrial data, and the simulation model then can be used to generate the entire data set for model training and testing.

#### 2.1.1 Aspen Plus simulation of PDH process

A PDH process is simulated by using Aspen Plus according to the Oleflex process developed by UOP company [30]. Figure 1 describes the detailed flowsheet of the PDH process developed in Aspen Plus simulation environment. The process feed is pre-processed to make the propane volume fraction to meet the feed requirements for the reaction in Unit 1 at first and then reacts with H<sub>2</sub> in a moving bed reactor with four parallel stages and radial flow in Unit 2. The reaction product enters the rapid cooling, high pressure dehydration, and cryogenic dehydrogenation in Unit 3. In the refining section (Unit 4), light ends (C1 and C2) are removed from a deethanizer column, and the propylene and propane are then separated via a propylene rectification column (RC). Propylene can be sold as a final product, while propane enters the reactor as a cyclic feed. The Flowrate of propane and propylene in the four moving bed reactors are shown in Table 1.

Actually, it is difficult to compare simulation results with industrial data directly due to the commercial confidentiality. In this study, we compare our reaction process (Unit 2) with the model which performed identically to plant data



**Fig. 1** Flowsheet of the PDH process built in Aspen Plus.

**Table 1** The flowrate of propane and propylene in moving bed reactors

Feed stream	Reactor 1		Reactor 2		Reactor 3		Reactor 4	
	Inlet	Outlet	Inlet	Outlet	Inlet	Outlet	Inlet	Outlet
$C_3H_8/(kg \cdot h^{-1})$	84461.9	74492.6	74492.6	67199.7	67199.7	61336.7	61336.7	56342.1
$C_3H_6/(kg \cdot h^{-1})$	891.1	10166.3	10166.3	16746.8	16746.8	21905.0	21905.0	26296.5

in recent literature [31]. The details of our simulation process and comparison are shown in Table 2. Both of them are highly consistent with each other based on the same reaction kinetics [32,33], reactors (adiabatic moving bed reactors) and heating method (inter-stage reheating in fired furnaces), which means that our simulation can be used to reflex the real solutions. The propane conversion ( $C$ ), propylene selectivity ( $S$ ), propylene yield ( $Y$ ) are defined, respectively, as:

$$C = \frac{F_{C_3H_8,in} - F_{C_3H_8,out}}{F_{C_3H_8,in}}, \quad (1)$$

$$S = \frac{F_{C_3H_6,in} - F_{C_3H_6,out}}{F_{C_3H_8,in} - F_{C_3H_8,out}}, \quad (2)$$

$$Y = C \times S, \quad (3)$$

where  $F_{C_3H_8,in}$  and  $F_{C_3H_6,in}$  are the input molar flow rates of  $C_3H_8$  and  $C_3H_6$ , respectively, and  $F_{C_3H_8,out}$  and  $F_{C_3H_6,out}$  are the output molar flow rates of  $C_3H_8$  and  $C_3H_6$ , respectively.

### 2.1.2 Data generation for model training and testing

The PDH process is composed of pumps, heat exchangers, moving bed reactors, compressors and rectifying columns. For data-driven modeling, the total annual profit ( $y_1$ ) and the propylene yield ( $y_2$ ) are set as the outputs, and the

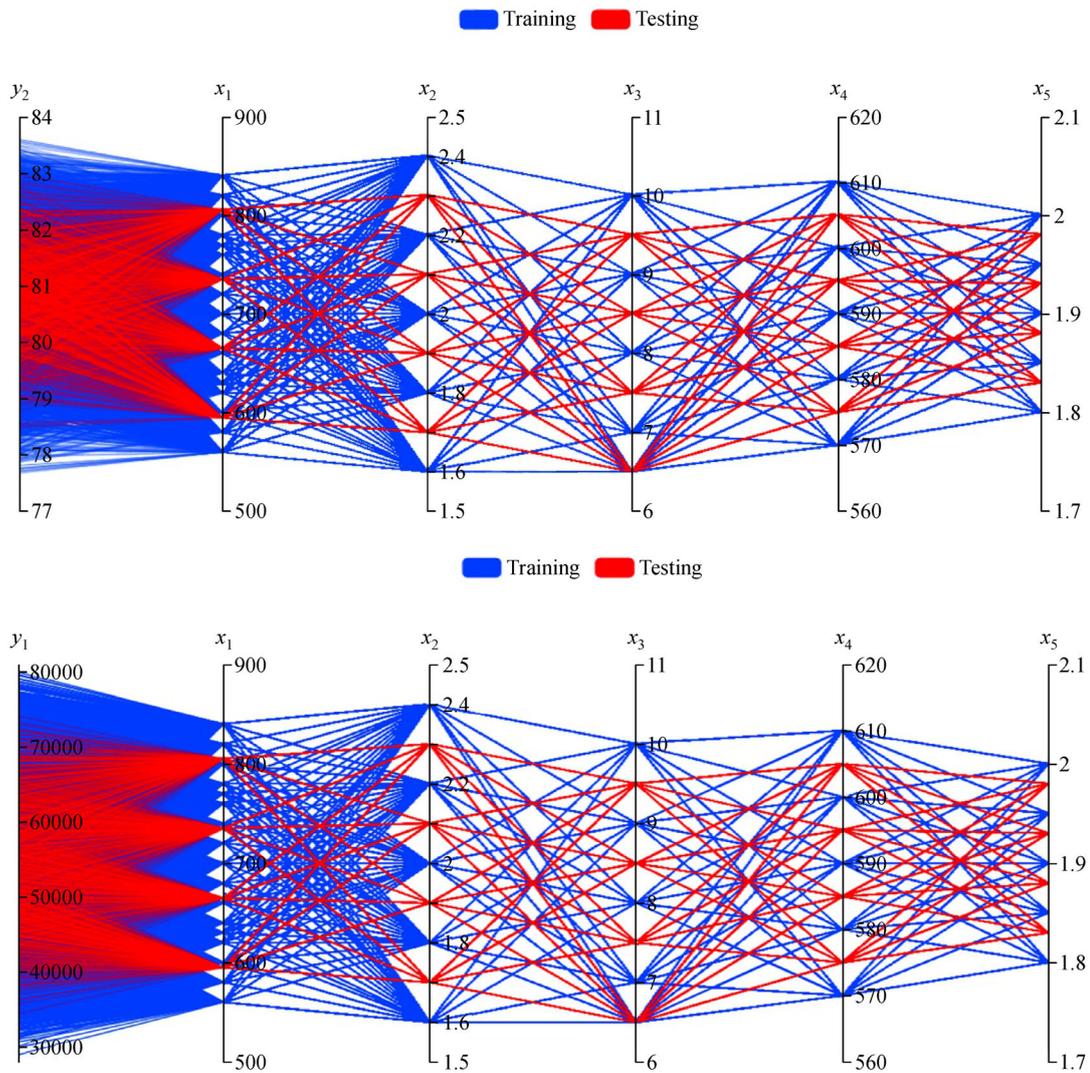
**Table 2** The results of previous model [31] and Aspen Plus simulation in reaction process

Index	Previous model	Aspen Plus simulation	Relative error/%
$C/\%$	33.2	33.29	0.271
$S/\%$	94.8	94.68	-0.127
$Y/\%$	31.5	31.52	0.0635

sensitivity analyses are conducted in Aspen Plus to find out the key parameters affecting the outputs. Firstly, ten candidate parameters are considered, such as the flowrate of propane feed, the flow ratio of hydrogen-to-propane, the pressure of propylene rectifier, the temperature and pressure of reactor, reflux rate, condenser temperature, condenser pressure, refining temperature, and refining pressure. By comparing the changing magnitude of outputs with the change of operating parameters, the flowrate of feed ( $x_1$ ), the flow ratio of hydrogen-to-propane ( $x_2$ ), the pressure of propylene rectifier ( $x_3$ ), and the temperature and pressure of reactor ( $x_4$  and  $x_5$ ) are identified as the key parameters. Data for model training and testing are further obtained by varying the five input parameters on Aspen Plus simulation. The operating ranges of the five parameters are determined according to the practical operating region, and thus 8750 groups of data are generated for model training with sampling average method as shown in Table 3. For the known operating region, average interpolation method is employed, i.e.,

**Table 3** Operating ranges and sampling points of inputs for data-driven modeling

Input	Operating range	Sampling points for training	Sampling points for testing
$x_1/(\text{kmol}\cdot\text{h}^{-1})$	560–840	560, 580, 600, 620, 630, 640, 660, 680, 700, 720, 740, 760, 770, 780, 800, 820, 840	595, 665, 735, 805
$x_2$	1.6–40	1.6, 1.8, 2, 2.2, 2.4	1.7, 1.9, 2.1, 2.3
$x_3/\text{bar}$	6.5–10	6.5, 7, 8, 9, 10	6.5, 7.5, 8.5, 9.5
$x_4/^\circ\text{C}$	570–610	570, 580, 590, 600, 610	575, 585, 595, 605
$x_5/\text{bar}$	1.8–2	1.8, 1.85, 1.9, 1.95, 2	1.83, 1.88, 1.93, 1.98

**Fig. 2** Sampling for model training and testing of the two outputs in the PDH process: total annual profit ( $y_1$ ) and propylene yield ( $y_2$ ).

1024 groups of data within the training set are selected for testing, where four points are sampled for each input parameter ( $4^5 = 1024$ ). Besides, random generation also can be used to obtain the training and testing data, if these data cover the most of the process operating region. Figure 2 presents the sampling features for both model training

and testing, which has covered most of the process operating region uniformly and densely to avoid over-fitting.

In order to consider more practical scenarios, some uncertain disturbances are also added in the Aspen Plus simulation, i.e., some parameters not selected as model

inputs are changed in a small range randomly. In the deethanizer column and the propylene rectification column, the temperature varying within  $\pm 2$  °C of error and the pressure varying within  $\pm 1\%$  of relative error randomly can represent the minor errors caused by the temperature and pressure sensors in practical process. Therefore, the process outputs might vary in an acceptable range even the same process inputs are set in the procedure of collecting training data.

## 2.2 Hybrid method for modeling and optimization of chemical processes

In our proposed hybrid method, four machine learning algorithms, KNN, support vector regression (SVR), DT, and ANN, are used for training the data-driven models based on the data provided in Section 2.1. Once the best model is identified, PSO can be applied to find the optimal operational solution of the process.

### 2.2.1 KNN method for modeling

KNN is a non-parametric method used for classification and regression [34]. KNN regression is presented to predict process performance in this study. It can estimate the property value of the object by using the average of the values of  $k$  nearest neighbors. The procedure of KNN can be described as the following five steps.

Step 1: initialize  $k$  to a positive integer value.

Step 2: select the type of distance, and calculate the distance based on the training data and testing data. There are several kinds of calculation, where the distances between testing data ( $x_i^{\text{test}}$ ) and training data ( $x_i^{\text{train}}$ ) are determined based on the dimensionality of their feature space ( $m$ ).

Euclidean distance:

$$d = \sqrt{\sum_{i=1}^m (x_i^{\text{test}} - x_i^{\text{train}})^2}. \quad (4)$$

Manhattan distance:

$$d = \sum_{i=1}^m |x_i^{\text{test}} - x_i^{\text{train}}|. \quad (5)$$

Chebyshev distance:

$$d = \max_i (|x_i^{\text{test}} - x_i^{\text{train}}|). \quad (6)$$

Step 3: predict the values of testing data. The  $k$  nearest neighbors can be used to predict the output values ( $y^{\text{pre}}$ ):

$$y^{\text{pre}} = \sum_{i=1}^k \lambda_i y_i^{\text{train}}. \quad (7)$$

There are two strategies of calculating. One is the average of the  $k$  nearest neighbors:

$$\lambda_i = 1/k. \quad (8)$$

The other is inversely proportional to the distance between testing data and train data ( $d_i$ ):

$$\lambda_i = \frac{e^{-d_i/\beta}}{\sum_{i=1}^k (e^{-d_i/\beta})}, \quad (9)$$

where  $\beta = 1$  or  $\beta = 2$  or  $\beta = 50$  or  $\beta = \sum_{i=1}^k \sqrt{(x_i^{\text{test}} - x_i^{\text{train}})^2} / (2k)$ .

Step 4: examine the accuracy of the model.  $R^2$  indicates the accuracy of the prediction:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}, \quad (10)$$

$$SS_{\text{res}} = \sum_i (y_i^{\text{pre}} - y^{\text{test}})^2, \quad (11)$$

$$SS_{\text{tot}} = \sum_i (y_i^{\text{pre}} - \bar{y}^{\text{pre}})^2. \quad (12)$$

Step 5: if the model accuracy is acceptable, stop the procedure; otherwise, update the value of  $k$ , and execute Steps 2 to 5 iteratively.

### 2.2.2 SVR method for modeling

SVR is developed from SVM [35]. SVM is a supervised learning theory originally used for solving classification problem, and later generalized to solve regression problems (called SVR). SVR is used to fit a hyperplane ( $y = f(x)$ ) through the input data ( $x$ ), where  $f(x, \omega) = \omega \cdot \Phi(x) + b$ ,  $\omega$  is the weight vector,  $b$  is a parameter, and  $\Phi(x)$  is the function of  $x$  for mapping the data into high-dimensional feature space. The goal of SVR is to minimize:

$$\frac{1}{2} \|\omega\|^2 + c \sum_{i=1}^m (\xi_i^{\vee} + \xi_i^{\wedge}), \quad (13)$$

subjected to:

$$-\varepsilon - \xi_i^{\vee} \leq y_i - \Phi(x_i) - b \leq \varepsilon + \xi_i^{\wedge}, \quad (14)$$

$$\xi_i^{\vee}, \xi_i^{\wedge} \geq 0, \quad (15)$$

where  $\varepsilon$  is a parameter representing the radius of insensitive zone around the hyperspace,  $\xi_i^{\vee}$  and  $\xi_i^{\wedge}$  are the slack variables which describe the difference between the outside point and insensitive zone, and  $c$  determines the trade-off between the training error and model flatness.

The above convex optimization problems (Eqs. (13–15)) can be transformed into a dual problem by introducing Lagrange multipliers, and shown as:

$$\begin{aligned} \min & \frac{1}{2} \sum_{i=1, j=1}^m (\alpha_i^{\wedge} - \alpha_i^{\vee}) (\alpha_j^{\wedge} - \alpha_j^{\vee}) K_{ij} \\ & + \sum_{i=1}^m [(\varepsilon - y_i) \alpha_i^{\wedge} + (\varepsilon + y_i) \alpha_i^{\vee}], \end{aligned} \quad (16)$$

subjected to:

$$\sum_{i=1}^m (\alpha_i^\wedge + \alpha_i^\vee) = 0, \quad (17)$$

$$0 \leq \alpha_i^\vee, \alpha_i^\wedge \leq Cm, \quad (18)$$

where  $K_{ij} = \Phi(x_i) \cdot \Phi(x_j)$  is kernel function and  $\alpha$  are the Lagrange multipliers.

There are three types of kernel function in this study.

Linear function:

$$K_{ij} = x_i \cdot x_j. \quad (19)$$

Polynomial function:

$$K_{ij} = (\gamma x_i \cdot x_j + r)^d, \quad (20)$$

where  $\gamma, r, d$  are kernel parameters.

Radial basis function:

$$K_{ij} = \exp\left(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\right) = \exp(-\gamma\|x_i - x_j\|^2), \quad (21)$$

where  $\sigma$  is the width of Gaussian kernel and  $\gamma$  is kernel function parameter (a positive value).

The dual problem can be solved by using sequential minimal optimization, and thus the values of  $\alpha_i^\wedge$  and  $\alpha_i^\vee$  are obtained. Then the weight vector  $\omega$  can be calculated by:

$$\omega = \sum_{i=1}^m (\alpha_i^\wedge - \alpha_i^\vee) \cdot \Phi(x_i), \quad (22)$$

while the support vectors  $(x_s, y_s)$  are calculated as:

$$\alpha_s^\wedge - \alpha_s^\vee \neq 0, \quad (23)$$

$$b_s = y_s - (\alpha_s^\wedge - \alpha_s^\vee) \cdot K(x, x_s) - \varepsilon (0 < \alpha_s^\wedge < c), \quad (24)$$

$$b_s = y_s - (\alpha_s^\wedge - \alpha_s^\vee) \cdot K(x, x_s) + \varepsilon (0 < \alpha_s^\vee < c). \quad (25)$$

Finally, the parameter  $b$  is computed as:

$$b = \frac{\sum_s b_s}{s}. \quad (26)$$

### 2.2.3 DT method for modeling

DT has been widely used in fault detection and classification, but less in solving process operating problems. In this study, classification and regression trees (CRT) is applied to model training [36]. The core of CRT is the linear regression for the parent node and child nodes in the tree:

$$\hat{y} = \mathbf{X} \cdot \mathbf{K}, \quad (27)$$

$$\mathbf{K} = [\mathbf{X}^T \cdot \mathbf{X}]^{-1} (\mathbf{X}^T \cdot \mathbf{Y}), \quad (28)$$

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1i} \\ \vdots & \ddots & \vdots \\ x_{j1} & \cdots & x_{ji} \end{pmatrix}, \quad (29)$$

$$\mathbf{Y} = \begin{pmatrix} y_{11} & \cdots & y_{1i} \\ \vdots & \ddots & \vdots \\ y_{j1} & \cdots & y_{ji} \end{pmatrix}, \quad (30)$$

where  $\mathbf{X}$  is the matrix of model inputs,  $\mathbf{Y}$  is matrix of model outputs,  $\mathbf{K}$  is the regression matrix, and  $\hat{y}$  is the prediction matrix. Once the predicted  $\hat{y}$  is obtained, the total variance ( $V$ ) can be given:

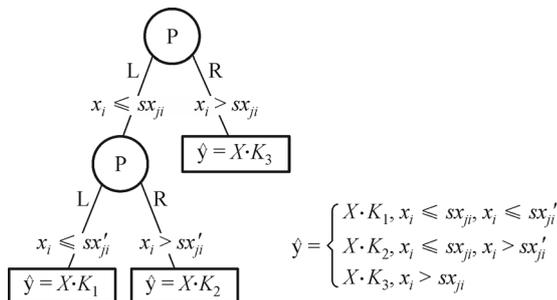
$$V = \sum_{j=1}^J \sum_{k=1}^K (\hat{y}_{jk} - y_{jk})^2. \quad (31)$$

The above regression will be used in the CRT procedure as follows. Step 1: set a value for parameter  $N$ . This  $N$  restricts the minimum number of training instances in a single node. Step 2: set the whole training data set as the parent node (P), and calculate its prediction ( $\hat{y}$ ) and total variance ( $V_p$ ). Step 3: split the data from the parent node (P) into two parts, namely left child node (L) and right child node (R). The left child node and right child node are separated based on a splitting value of  $x_i(sx_{ji})$ .  $x_i$  and its splitting value are selected arbitrarily at first. The data set with  $x_{ji} \leq sx_{ji}$  is assigned to the left child node (L), while the rest data set is assigned to the right child node (R). It must be restricted that the amounts of data in L and R must be equal to or larger than the minimum number  $N$  set in Step 1. Thus, the variances of L and R related to  $sx_{ji}$  splitting ( $V_{L,sx_{ji}}$  and  $V_{R,sx_{ji}}$ ) can be obtained according to Eqs. (24–28). The total variance related to  $sx_{ji}$  splitting ( $V_{sx_{ji}}$ ) is the sum of its left variance ( $V_{L,sx_{ji}}$ ) and right variance ( $V_{R,sx_{ji}}$ ). Step 4: find all possible  $x_i$  and its value ( $sx_{ji}$ ) for splitting, and calculate their relevant variances ( $V_{sx_{ji}}$ ). The  $dx_i$  and its splitting value ( $dx_{ji}$ ) with the minimum variance ( $V_{dx_{ji}}$ ) compared with all possible variances is selected as the best division of the parent node. Step 5: if the minimum variance ( $V_{dx_{ji}}$ ) found in Step 4 is less than the variance of parent node ( $V_p$ ), L and R will be set as two new parent nodes, and the splitting strategies proposed in Step 3 to Step 4 are executed to split the new parent nodes iteratively. The whole procedure will be terminated, if no better variance can be found in any child nodes.

Figure 3 illustrates a CRT structure with two parent nodes and three child nodes, where  $\hat{y} = f(\mathbf{X})$  can be formulated as a piecewise linear function.

### 2.2.4 ANN method for modeling

Typical ANN consists of three layers, the input layer ( $l_1$ ), hidden layer ( $l_2$ ), and output layer ( $l_3$ ) [37]. Each layer is a vector. The dimension of the vector is also called the number of nodes in the layer. The input layer receives the input data from training dataset. The adjacent layers are connected via weight matrices ( $W_i$ ), bias vectors ( $b_i$ ), and an activation function  $g$ :



**Fig. 3** Illustration of a CRT structure with two parent nodes and three child nodes.

$$l_{i+1} = g(W_i \cdot l_i + b_i), \quad i = 1, 2. \quad (32)$$

Four types of activation function (ReLU, Softplus, Sigmoid, and Tanh) are addressed.

$$\text{ReLU}(x) = \max(x, 0). \quad (33)$$

ReLU is a piecewise linear function, and works only if  $x$  is greater than or equal to 0. When  $x$  is less than 0, the node will not be updated during the whole training process.

$$\text{Softplus}(x) = \ln(1 + e^x). \quad (34)$$

Softplus can be regarded as a smoothing ReLU. It overcomes the disadvantage of ReLU but takes more calculations.

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (35)$$

Sigmoid is a non-linear function that converts the range of  $x$  from  $(-\infty, +\infty)$  to  $(0, 1)$ . However, when  $|x|$  is too large (such as  $|x| \rightarrow +\infty$ ),  $\text{Sigmoid}'(x) \rightarrow 0$ , which may slow down the training process of the model.

$$\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (36)$$

Tanh is a non-linear function that maps the range of  $x$  from  $(-\infty, +\infty)$  to  $(-1, 1)$ . It may also slow down the training process of the model when  $|x|$  is too large like sigmoid.

Then, a loss function is set to measure the loss between the output layer and the validation data. In this work, the loss function is expressed with the mean square error (*MSE* in 37), or the sum of *MSE* (37) and  $L_2$ -regulation function (38), where  $k$  is the regularization rate, and  $\omega_i$  is the weights of the neural network. The aim of  $L_2$ -regulation function is to prevent overfitting while minimizing the loss of prediction.

$$\text{MSE} = \sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}. \quad (37)$$

$$L_2 = \frac{1}{2} \cdot k \sum_i \omega_i^2. \quad (38)$$

$$f = \text{MAE} + L_2. \quad (39)$$

Five optimization algorithms, namely, gradient descent, momentum, adaptive gradient (AdaGrad), root mean square prop (RMSProp), and adaptive moment estimation (Adam), can be used to minimize the loss. Gradient descent works by taking the gradient of the weight space to find the path of steepest descent. Momentum improves gradient descent. It can accumulate velocity in the direction where the gradient is pointing towards the same direction across iterations, and thus avoid getting stuck in a local minimum. AdaGrad is a modified stochastic gradient descent with per-parameter learning rate. It increases the learning rate for sparser parameters and decreases the learning rate for fewer sparse ones. RMSProp is also a method of addressing learning rate. It describes the learning rate for a weight by a running average of the magnitudes of recent gradients. Adam is the combination of momentum and RMSProp.

The aforementioned activation functions and optimization algorithms for minimizing the loss are utilized in the ANN training procedure as follows. Step 1: set the nodes in the hidden layer, and generate the initial weights ( $\omega_i$ ) and bias ( $b_i$ ) of the neural network randomly. Step 2: calculate output value based on Eqs. (32) to (35), and then evaluate the loss between the output value and the real value through the loss function Eqs. (37–39). Step 3: update the weights ( $\omega_i$ ) and bias ( $b_i$ ) of the neural network by using one of the optimization algorithms to minimize the loss. Step 4: execute Step 2 and Step 3 iteratively, until the maximum number of iteration is reached or the loss is acceptable.

### 2.2.5 PSO for process optimization

Once the most efficient model of the chemical process is selected by comparing the four machine learning algorithms, the process operation can be optimized with PSO algorithm, a global random search algorithm with simple structure and easy programming [38] based on swarm intelligence [39].

In PSO algorithm, each particle  $i$  can be described with an  $N$ -dimensional position vector ( $x_i = (x_i^1, x_i^2, \dots, x_i^N)$ ) and velocity vector ( $v_i = (v_i^1, v_i^2, \dots, v_i^N)$ ,  $v_i \in (-v_{\max}, v_{\max})$ ). The particles will search the  $x_i$  for optimal objective value in each individual particle, and then determine the values of  $x_i$  for the global optimal objective value among all particles.

In the entire algorithm procedure, PSO requires a series of searching iterations. The position and velocity vector of each particle is randomly selected in the first iteration. In the rest iterations, particle swarm will update the position vector ( $x_i$ ) and velocity vector ( $v_i$ ) according to the inertia, individual optimal value ( $p_i$ ) and global optimal value ( $p_g$ ) as follow:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 (p_i - x_i^k) + c_2 r_2 (p_g - x_i^k), \quad (40)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \quad (41)$$

where  $x_i^k$  and  $v_i^k$  denote the position and the instantaneous velocity of particle  $i$  in iteration  $k$ ,  $\omega$  is the inertia coefficient,  $c_1$  and  $c_2$  are the acceleration factor, and  $r_1$  and  $r_2$  are the random number ranging from 0 to 1.

Thus, the model objective value found by particle  $r$  in iteration  $k$  ( $obj_i^k$ ) is calculated based on its updated position vector ( $x_i^k$ ). If  $obj_i^k$  is better than the previous optimal value ( $obj_i^{k-1}$ ),  $x_i^k$  is set to  $p_i$ ; otherwise,  $p_i$  is retained. Consequently, we can compare the  $obj_i^k$  of all particles, and select the best  $obj_i^k$  as the global optimal value. If this global optimal value is better than the  $p_g$  found in the previous iteration,  $p_g$  can be replaced with  $x_i^k$ ; otherwise,  $p_g$  is retained. The procedure will stop when the maximum iteration number is reached.

### 3 Results and discussion

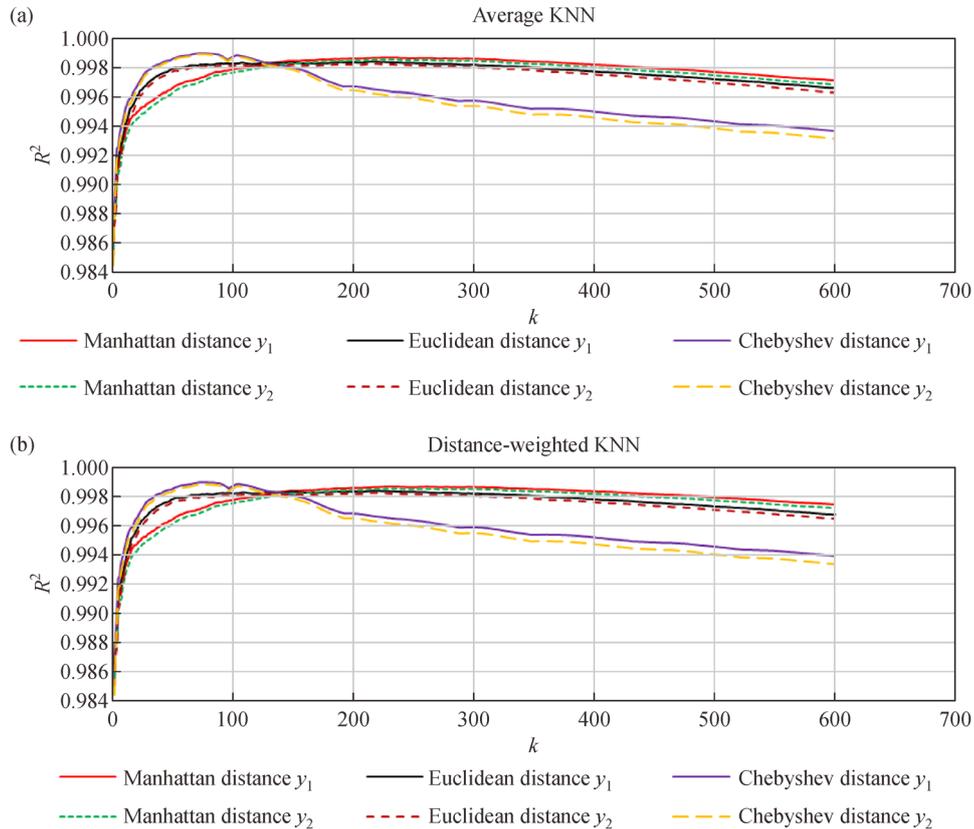
The datasets generated in Section 2.1 are used to train the data-driven models of process operations. Firstly, the training models are presented individually given by KNN, SVR, DT, and ANN. And then the most efficient model is selected to be combined with PSO for searching the optimal operations of PDH process.

#### 3.1 Data-driven models of the PDH process

As mentioned in Section 2.1, 8750 groups and 1024 groups of datasets for model training and testing are generated by changing the values of the five key inputs (Table 3), which covers all possible operating region of the PDH process related to total profit ( $y_1$ ) and propylene yield ( $y_2$ ) (Fig. 2). The following results present the performances of the four machine learning algorithms for the process modeling.

##### 3.1.1 KNN for process modeling

It is noted that three options can be adjusted to improve the KNN model, i.e., the value of  $k$  nearest neighbors, the type of distance calculation (Euclidean distance, Manhattan distance, and Chebyshev distance), and the calculations of  $\lambda_i$  for prediction (8) and (9), as shown in Section 2.2.1. Figure 4 presents the  $R^2$  of test data with different combinations of  $k$ , distance calculations, and predictive equations, where  $R^2$  increases significantly with  $k$  and reaches the maximum in a certain  $k$  value; but after this  $k$  value,  $R^2$  is slightly inverse to  $k$  increment. When  $k$  is very small, the model is very sensitive to the nearest neighbors. But when  $k$  becomes too large, the distinct data with great differences are classified into the same category, which leads to the error in model prediction.



**Fig. 4**  $R^2$  of test data with different combinations of  $k$ , distance calculations, and predictive equations based on (a) average KNN and (b) distanced-weighted KNN.

Table 4 lists the best KNN models under different combinations of  $k$ , distance calculations, and predictive equations, where the combination of Chebyshev distance and distance-weighted KNN can get the best results of  $y_1$  with ( $k = 76$ ) and  $y_2$  with ( $k = 77$ ).

### 3.1.2 SVR for process modeling

As stated in Section 2.2.2, parameter  $c$  determines the trade-off between the training error and flatness of the SVR model, and the kernel functions are the core of pattern analysis for finding and studying general types of relations. Thus, the three kernel functions (linear function, polynomial function, and radial basis function) and various values of  $c$  are taken into account for SVR model training. Figure 5 describes the  $R^2$  of test data under different combinations of kernel functions and parameter  $c$ . It can be found in Fig. 5 that the radial basis function performs the best when  $c$  increases to a certain value; the linear function is less affected by  $c$  but gives low  $R^2$  while the polynomial function performs similarly to the radial basis function for predicting  $y_2$ , but will suffer from overfitting problem for predicting  $y_1$  when  $c$  increases to 100.

Table 5 presents the best  $R^2$  of SVR models under the specific  $c$  and kernel functions, and indicates that the SVR model with radial basis function can predict the most accurate  $y_1$  and  $y_2$  in the conditions of  $c = 5$  and  $c = 90$ , respectively.

### 3.1.3 DT for process modeling

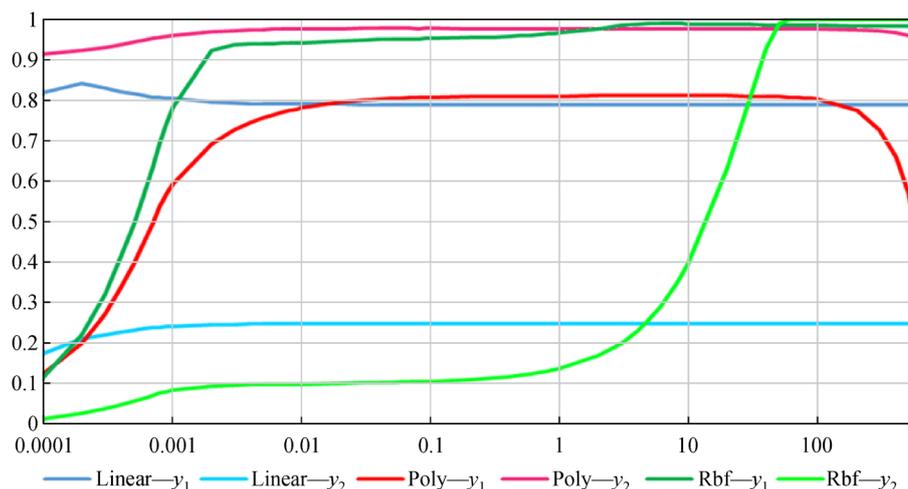
In the DT algorithm (Section 2.2.3), the minimum number of training instances in a single node ( $N$ ) is the key control parameter to determine the DT model. If  $N$  is too small, a singular matrix will occur, which may cause infeasible calculation to obtain the regression matrix ( $K$ ). However, a higher number of  $N$  will result in a smaller number of child nodes, and leads to inaccurate prediction. As shown in Fig. 6, when  $N$  ranges from 2501 to 3875, the DT models perform very well for predicting  $y_1$  ( $R^2 = 0.99482$ ) and  $y_2$  ( $R^2 = 0.99239$ ) while the predictions of  $y_1$  and  $y_2$  have a sharp decline ( $R^2 = 0.08526$  and  $0.04145$ ) when  $N$  increases over 3876 and no DT model is found when  $N$  is lower than 2500 due to the singular matrix. Thus, the DT models generated in the condition of  $2500 < N < 3876$  were used for the process prediction in this study.

### 3.1.4 ANN for process modeling

Four activation functions and five optimization algorithms have been introduced for ANN training in Section 2.2.4. Thus, the combination of different activation functions and optimization algorithms are compared. Figure 7 shows the above combinations with the best operating parameters under the ANN structure of 10 nodes in the hidden layer ( $l_2$ ). It is observed that all combinations can achieve high accuracy when the training sets are sufficient. The

**Table 4** The best KNN models under different combinations of  $k$ , distance calculations, and predictive equations

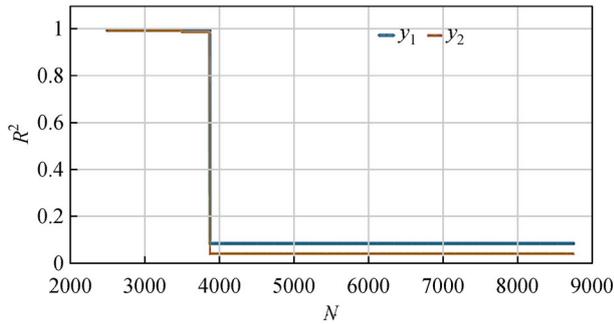
Output	Average KNN						Distance-weighted KNN					
	Chebyshev distance		Euclidean distance		Manhattan distance		Chebyshev distance		Euclidean distance		Manhattan distance	
	$k$	$R^2$	$k$	$R^2$	$k$	$R^2$	$k$	$R^2$	$k$	$R^2$	$k$	$R^2$
$y_1$ /(MS·year <sup>-1</sup> )	76	0.99899	217	0.99841	232	0.99870	76	0.99900	217	0.99844	234	0.99872
$y_2$ /%	77	0.99890	217	0.99827	232	0.99858	77	0.99891	217	0.99830	234	0.99860



**Fig. 5**  $R^2$  of test data under different combinations of kernel functions and parameters  $c$  in SVR models.

**Table 5** The best SVR models under different combinations of Kernel functions and parameter  $c$

Output	Linear function (linear)		Polynomial function (poly)		Radial basis function (rbf)	
	$c$	$R^2$	$c$	$R^2$	$c$	$R^2$
$y_1/(\text{MS}\cdot\text{year}^{-1})$	0.0002	0.84363	3	0.81170	5	0.99050
$y_2/\%$	0.07	0.24939	0.04	0.97768	90	0.99910

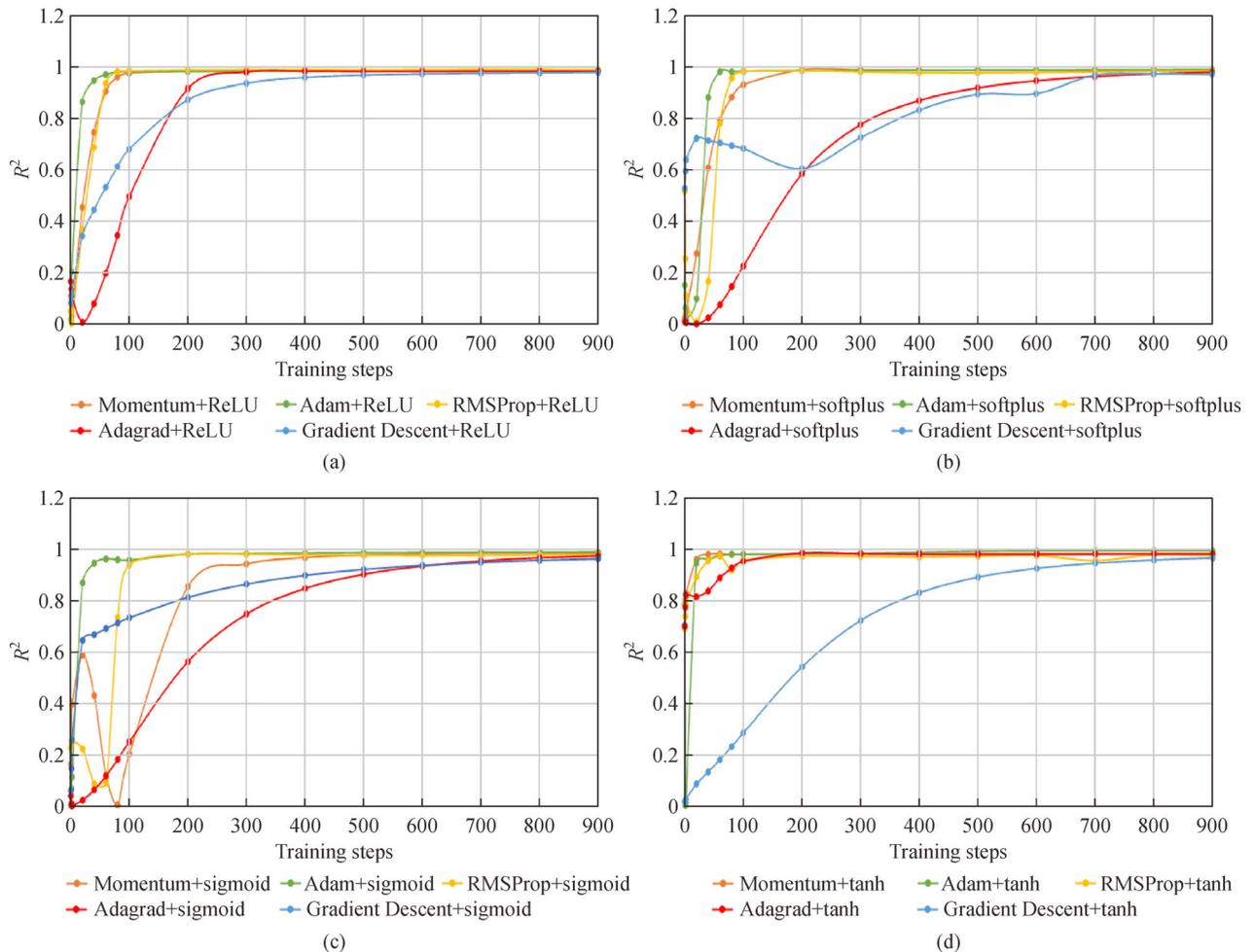


**Fig. 6**  $R^2$  of test data under different parameter  $N$  in DT models.

activation functions ReLU and Tanh, and the optimization algorithms Adam and RMSProp perform much better than the others. Some methods have fluctuation in the beginning steps showing their procedure of jumping out of a local optimum to find the global optimization. Table 6 lists the best ANN models under the different combinations of nodes in  $l_2$ , activation functions, and optimization algorithms, where the combination of Softplus and RMSProp can get the best results of prediction for  $y_1$  and  $y_2$ .

3.2 Optimization of the PDH process operations

For better optimization ability, the impact of key operating



**Fig. 7**  $R^2$  of test data with different optimization algorithms based on different activation functions: (a) ReLU, (b) softplus, (c) sigmoid, and (d) Tanh.

**Table 6** The best ANN models under different combinations of nodes in  $L_2$  ( $N$ ), activation functions, and optimization algorithms

Output	ReLU		Tanh		Softplus		Sigmoid	
	$N$	$R^2$	$N$	$R^2$	$N$	$R^2$	$N$	$R^2$
Root mean square prop (RMSProp)								
$y_1/(\text{MS}\cdot\text{year}^{-1})$	12	0.9934	8	0.9938	10	0.9940	8	0.9841
$y_2/\%$	12	0.9894	12	0.9897	12	0.9902	8	0.9901
Adaptive moment estimation (Adam)								
$y_1/(\text{MS}\cdot\text{year}^{-1})$	12	0.9934	6	0.9936	6	0.9937	10	0.9847
$y_2/\%$	12	0.9891	12	0.9901	12	0.9901	10	0.9897
Adaptive gradient (AdaGrad)								
$y_1/(\text{MS}\cdot\text{year}^{-1})$	10	0.9850	12	0.9817	8	0.9855	6	0.9805
$y_2/\%$	12	0.9897	6	0.9910	10	0.9723	8	0.9701
Momentum								
$y_1/(\text{MS}\cdot\text{year}^{-1})$	6	0.9922	8	0.9833	12	0.9850	12	0.9817
$y_2/\%$	12	0.9903	10	0.9757	8	0.9702	10	0.9696
Gradient descent								
$y_1/(\text{MS}\cdot\text{year}^{-1})$	8	0.9900	12	0.9827	12	0.9848	12	0.9820
$y_2/\%$	6	0.9899	12	0.9806	8	0.9713	12	0.9697

parameters (stated in (40)) on controlling PSO procedure is also considered, including the acceleration factors  $c_1$  and  $c_2$ , and the inertia coefficient  $\omega$  (Fig. 8). It can be found that,  $c_1$  does not influence the global optimization. A larger value of  $c_1$  will enhance the individual local optimization ability of the particles, which is not observed in Fig. 8, while  $c_2$  improves the global optimization ability of the particle swarm, thereby increasing the probability of finding the global optimal value as described in Fig. 8. A medium range (such as 0.45–3.0) of inertia coefficient  $\omega$  is required. In this range of  $\omega$ , most values of acceleration factors  $c_1$  and  $c_1$  can find the global optimization efficiently. To balance computational cost and optimization ability, proper parameters were selected for further improvements.

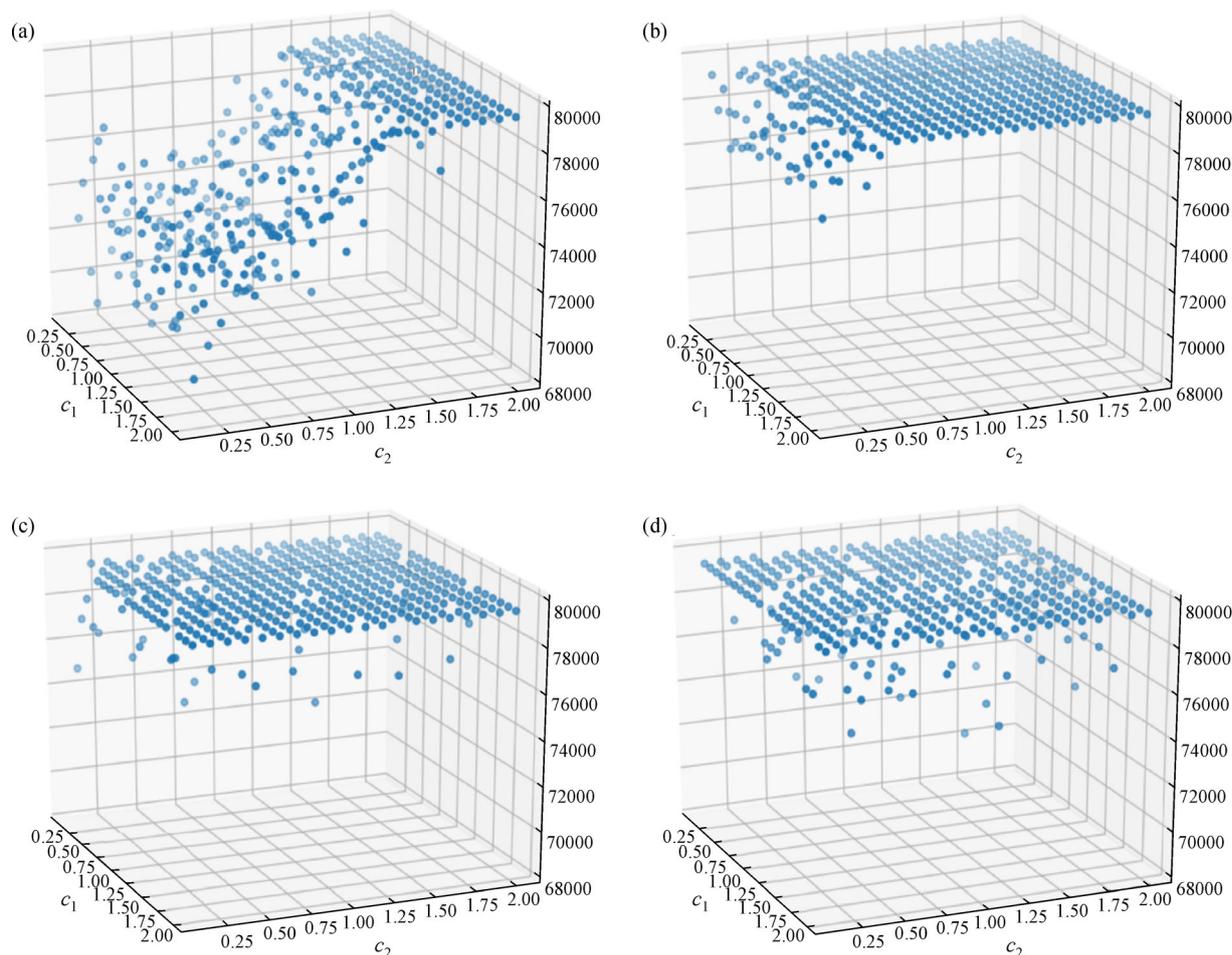
Since the training data generated from the PDH Process have covered most of the process operating region (Fig. 2), all the four machine learning methods can achieve very high accuracy in the prediction with suitable algorithm parameters. However, regarding process optimization, PSO algorithm (described in Section 2.2.5) need to evaluate the process objective values ( $y_1$  or  $y_2$ ) based on the data-driven model by updating the position vector ( $x_i^k$ ) in each iteration  $k$  and particle  $i$ . It is noted that KNN must compute the distance and sort all the training data at each prediction, which is very slow as there are a large number of training data (8750) in our case. Thus, due to the huge computational resources required by KNN, it is not recommended to use KNN for evaluating the process objective values in the PSO procedure. As demonstrated in the former section, SVR, DT, and ANN can learn knowledge from the training data, and give simple and

accurate models for process operations. After obtaining data-driven models, the models with the biggest  $R^2$  are selected for optimizing. Therefore, DT model ( $R^2 = 0.99482$ ) is used to predict  $y_1$ , and SVM model ( $R^2 = 0.99910$ ) is used to predict  $y_2$ .

In the PSO procedure, the particle group can reach a steady state around 5–10 iterations, showing a fast convergence. Table 7 presents the solutions of maximization of total profit ( $y_1$ ) and propylene yield ( $y_2$ ) obtained by PSO. These optimal solutions are identical to the Aspen Plus results under the same input parameters. Thus, the efficiency of our hybrid method is validated. It seems that enumeration is more straightforward than the data-driven models for optimization as there are four variables ( $x_1$ ,  $x_2$ ,  $x_4$  and  $x_5$ ) which reach the boundary of training set in this study. However, the proposed optimization method is capable of solving general nonlinear problems. Specially, if the product yield ( $y_2$ ) is restricted in a certain range, using the data-driven models is more effective than enumeration for maximizing total annual profit ( $y_1$ ), as there is a trade-off between  $y_1$  and  $y_2$ . Improving operating conditions for high yield may increase annual profit but it may increase the cost of operating, resulting in low profit. For example, increasing the temperature within a certain range can promote the endothermic reaction but requires a lot of heat, so there is an optimal temperature that brings about maximum profit but not the maximum yield.

## 4 Conclusions

Machine learning algorithms may not perform well if the



**Fig. 8** The impact of key operating parameters ( $c_1$ ,  $c_2$ ) on controlling PSO procedure: (a)  $\omega = 0.05$ , (b)  $\omega = 0.45$ , (c)  $\omega = 3.0$ , and (d)  $\omega = 10$ .

**Table 7** The optimal solutions obtained by PSO <sup>a)</sup>

Max	$x_1/(\text{kmol}\cdot\text{h}^{-1})$	$x_2$	$x_3/\text{bar}$	$x_4/^\circ\text{C}$	$x_5/\text{bar}$	Obj	Aspen Plus validation	Error/%
$y_1/(\text{M}\$ \cdot \text{year}^{-1})$	840	1.6	9.646	610	1.8	79438	79957	0.65
$y_2(\%)$	840	1.6	6.500	610	1.8	83.53	83.54	0.01

a)  $x_1$ : feed flowrate;  $x_2$ : hydrogen to propane ratio;  $x_3$ : propylene rectifier pressure;  $x_4$ : reactor temperature;  $x_5$ : reactor pressure,  $y_1$ : total annual profit;  $y_2$ : propylene yield.

training data are insufficient. There is a growing concern about the lack of variety from industrial data, as the process must run in a relatively steady state. This study presents an industry-based and mechanistic simulation strategy to generate sufficient data for model training and testing. The generated data sets can cover all possible process operating region. Based on the proposed data generation strategy, four machine learning algorithms, KNN, SVM, DT, and ANN all obtained highly accurate models for the process operations. Moreover, the most efficient models are selected to be combined with PSO to find the optimal process operations under maximum profit or product yield. The proposed data collection strategy and the hybrid

framework integrating machine learning with PSO are generalized to a broader class of chemical processes.

**Acknowledgements** This work was supported by the “Zhujiang Talent Program” High Talent Project of Guangdong Province (Grant No. 2017GC010614); and the National Natural Science Foundation of China (Grant No. 22078372).

## References

1. Jenck J F, Agterberg F, Droescher M J. Products and processes for a sustainable chemical industry: a review of achievements and

- prospects. *Green Chemistry*, 2004, 6(11): 544
- Vooradi R, Anne S B, Tula A K, Eden M R, Gani R. Energy and CO<sub>2</sub> management for chemical and related industries: issues, opportunities and challenges. *BMC Chemical Engineering*, 2019, 1(1): 7
  - Worrell E, Cuelenaere R F A, Blok K, Turkenburg W C. Energy consumption by industrial processes in the European Union. *Energy*, 1994, 19(11): 1113–1129
  - Ding J, Modares H, Chai T, Lewis F L. Data-based multiobjective plant-wide performance optimization of industrial processes under dynamic environments. *IEEE Transactions on Industrial Informatics*, 2016, 12(2): 454–465
  - Hammer M. *Management Approach for Resource-Productive Operations*. Wiesbaden: Springer Gabler, 2018, 11–26
  - Ibrahim D, Jobson M, Guillén-Gosálbez G. Optimization-based design of crude oil distillation units using rigorous simulation models. *Industrial & Engineering Chemistry Research*, 2017, 56(23): 6728–6740
  - Pattison R C, Gupta A M, Baldea M. Equation-oriented optimization of process flowsheets with dividing-wall columns. *AIChE Journal*. American Institute of Chemical Engineers, 2016, 62(3): 704–716
  - Menezes B C, Kelly J D, Grossmann I E. Improved swing-cut modeling for planning and scheduling of oil-refinery distillation units. *Industrial & Engineering Chemistry Research*, 2013, 52(51): 18324–18333
  - Bo D, Yang K, Xie Q, He C, Zhang B, Chen Q, Qi Z, Ren J, Pan M. A novel approach for detailed modeling and optimization to improve energy saving in multiple effect evaporator systems. *Industrial & Engineering Chemistry Research*, 2019, 58(16): 6613–6625
  - Butler K T, Davies D W, Cartwright H, Isayev O, Walsh A. Machine learning for molecular and materials science. *Nature*, 2018, 559(7715): 547–555
  - Hotta S, Kiyasu S, Miyahara S. Pattern recognition using average patterns of categorical k-nearest neighbors. In: *Proceedings of the 17th International Conference on Pattern Recognition*. Washington, DC: IEEE, 2004
  - Adeniyi D A, Wei Z, Yongquan Y. Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics*, 2016, 12(1): 90–108
  - Dang T T, Ngan H Y T, Liu W. Distance-based k-nearest neighbors outlier detection method in large-scale traffic data. In: *IEEE International Conference on Digital Signal Processing (DSP)*. Washington, DC: IEEE, 2015
  - Zhu W, Sun W, Romagnoli J. Adaptive k-nearest-neighbor method for process monitoring. *Industrial & Engineering Chemistry Research*, 2018, 57(7): 2574–2586
  - Al-Jamimi H A, Bagudu A, Saleh T A. An intelligent approach for the modeling and experimental optimization of molecular hydrodesulfurization over AlMoCoBi catalyst. *Journal of Molecular Liquids*, 2019, 278: 376–384
  - Yang D, Zhong W, Chen X, Zhan J, Wang G. Structure optimization of vessel seawater desulphurization scrubber based on CFD and SVM-GA methods. *Canadian Journal of Chemical Engineering*, 2019, 97(11): 2899–2909
  - Golkamarenji G, Naebe M, Badii K, Milani A S, Jazar R N, Khayyam H. Support vector regression modeling and optimization of energy consumption in carbon fiber production line. *Computers & Chemical Engineering*, 2018, 109: 276–288
  - Yu Z, Yousaf K, Ahmad M, Yousaf M, Gao Q, Chen K. Efficient pyrolysis of ginkgo biloba leaf residue and pharmaceutical sludge (mixture) with high production of clean energy: process optimization by particle swarm optimization and gradient boosting decision tree algorithm. *Bioresource Technology*, 2020, 304: 123020
  - Hough B R. *Computational approaches and tools for modeling biomass pyrolysis*. Dissertation for the Doctoral Degree. Washington: University of Washington, 2016, 78–94
  - Saleem M, Ali I. Machine learning based prediction of pyrolytic conversion for red sea seaweed. In: *7th International Conference on Biological, Chemical & Environmental Sciences*. Budapest (Hungary), 2017, 27–31
  - Hough B R, Beck D A, Schwartz D T, Pfaendtner J. Application of machine learning to pyrolysis reaction networks: reducing model solution time to enable process optimization. *Computers & Chemical Engineering*, 2017, 104: 56–63
  - Mirshahvalad H, Ghasemiasl R, Raoufi N, Malekzadeh dirin M. A neural network QSPR model for accurate prediction of flash point of pure hydrocarbons. *Molecular Informatics*, 2019, 38(4): 1800094
  - Wang Z, Su Y, Jin S, Shen W, Ren J, Zhang X, Clark J H. A novel unambiguous strategy of molecular feature extraction in machine learning assisted predictive models for environmental properties. *Green Chemistry*, 2020, 22(12): 3867–3876
  - Sosa A, Ortega J, Fernández L, Palomar J. Development of a method to model the mixing energy of solutions using COSMO molecular descriptors linked with a semi-empirical model using a combined ANN-QSPR methodology. *Chemical Engineering Science*, 2020, 224: 115764
  - Su Y, Wang Z, Jin S, Shen W, Ren J, Eden M R. An architecture of deep learning in QSPR modeling for the prediction of critical properties using molecular signatures. *AIChE Journal*. American Institute of Chemical Engineers, 2019, 65(9): e16678
  - Schweidtmann A M, Huster W R, Lüthje J T, Mitsos A. Deterministic global process optimization: accurate (single-species) properties via artificial neural networks. *Computers & Chemical Engineering*, 2019, 121: 67–74
  - Chandrasekaran M, Tamang S. ANN-PSO integrated optimization methodology for intelligent control of MMC machining. *Journal of the Institution of Engineers (India): Series C*, 2017, 98(4): 395–401
  - Zhang X, Zhou T, Zhang L, Fung K Y, Ng K M. Food product design: a hybrid machine learning and mechanistic modeling approach. *Industrial & Engineering Chemistry Research*, 2019, 58(36): 16743–16752
  - Zhu Y, Hou Z, Qian F, Du W. Dual RBFNNs-based model-free adaptive control with aspen HYSYS simulation. *IEEE Transactions on Industrial Informatics*, 2016, 28(3): 759–765
  - Myers D N, Myers D N, Zimmermann J E. US Patent, 20100916969, 2010-11-1
  - Chin S, Radzi S, Maharon I, Shafawi M. Kinetic model and simulation analysis for propane dehydrogenation in an industrial

- moving bed reactor. *World Academy of Science, Engineering and Technology*, 2011, 52: 183–189
32. Loc L C, Gaidai N, Kiperman S, Thoang H S. Kinetics of propane and n-butane dehydrogenation over platinum-alumina catalysts in the presence of hydrogen and water vapor. *Kinetics and Catalysis*, 1996, 37(6): 790–796
  33. Røsjorde A, Kjelstrup S, Johannessen E, Hansen R. Minimizing the entropy production in a chemical process for dehydrogenation of propane. *Energy*, 2007, 32(4): 335–343
  34. García-Pedrajas N, del Castillo J A R. A proposal for local k values for k-nearest neighbor rule. *IEEE Transactions on Industrial Informatics*, 2017, 28(2): 470–475
  35. Zhu F, Gao J, Xu C, Yang J, Tao D. On selecting effective patterns for fast support vector regression training. *IEEE Transactions on Industrial Informatics*, 2018, 29(8): 3610–3622
  36. Loh W Y. Classification and regression trees. *Wiley Interdisciplinary Reviews. Data Mining and Knowledge Discovery*, 2011, 1(1): 14–23
  37. Hsu K, Gupta H V, Sorooshian S. Artificial neural network modeling of the rainfall-runoff process. *Water Resources Research*, 1995, 31(10): 2517–2530
  38. Yang Q, Yang Z, Zhang T, Hu G. A random chemical reaction optimization algorithm based on dual containers strategy for multi-rotor UAV path planning in transmission line inspection. *Concurrency and Computation*, 2019, 31(12): e4658
  39. Kennedy J, Eberhart R. Particle swarm optimization. In: *Proceedings of ICNN'95-International Conference on Neural Networks*. Washington, DC: IEEE, 1995