Design and Application of a Variable Selection Method for Multilayer Perceptron Neural Network With LASSO

Kai Sun, Shao-Hsuan Huang, David Shan-Hill Wong, and Shi-Shang Jang

***Abstract*— In this paper, a novel variable selection method for neural network that can be applied to describe nonlinear industrial processes is developed. The proposed method is an iterative two-step approach. First, a multilayer perceptron is constructed. Second, the least absolute shrinkage and selection operator is introduced to select the input variables that are truly essential to the model with the shrinkage parameter is determined using a cross-validation method. Then, variables whose input weights are zero are eliminated from the data set. The algorithm is repeated until there is no improvement in the model accuracy. Simulation examples as well as an industrial application in a crude distillation unit are used to validate the proposed algorithm. The results show that the proposed approach can be used to construct a more compressed model, which incorporates a higher level of prediction accuracy than other existing methods.**

***Index Terms*— Crude distillation unit (CDU), inferential modeling, least absolute shrinkage and selection operator (LASSO), neural network, variable selection.**

I. Introduction

In THE past few decades, artificial neural networks (ANNs) have been widely employed in a variety of fields, such as pattern recognition, machine learning, combinatorial optimization, and nonlinear regression [1]. Many process variables are expensive, unreliable, or difficult to measure in real industrial processes. Inferential modeling provides a valuable alternative, which involves the inference of these variables using other more easily measurable variables [2]. The majority of industrial processes are of a highly nonlinear nature, which makes them difficult to model with the use of linear methods. ANNs are advanced algorithms that can describe complex, nonlinear processes, and have been widely applied to inferential modeling in recent years [3]–[9].

Manuscript received June 26, 2014; accepted March 9, 2016. The work of K. Sun was supported in part by the Shandong Provincial Natural Science Foundation of China under Grant ZR2010FQ009 and in part by the Construction Project of Shandong Provincial Characteristic Specialty. The work of S.-H. Huang, D. S.-H. Wong, and S.-S. Jang was supported in part by the Ministry of Economic Affairs under Grant 102-EC-17-A-09-S1-198, in part by the National Science Council under Grant 100-2221-E-007-058-MY2, and in part by the Advanced Manufacturing and Service Management Research Center of National Tsing-Hua University under Grant 101N2072E1.

K. Sun is with the Department of Automation, Qilu University of Technology, Jinan 250353, China (e-mail: sunkai79@qlu.edu.cn).

S.-H. Huang, D. S.-H. Wong, and S.-S. Jang are with the Department of Chemical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: aj114kimo@hotmail.com; dshwong@gmail.com; ssjang@mx.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2542866

An additional feature of tangible industrial processes is that there are usually a high number of candidate explanatory variables with a high cross correlation. Variable selection techniques can be applied in order to improve the prediction accuracy, reduce the complexity of the model, better capture the nature of the industrial process, and reduce the cost of measurements [10]. There are many researchers focusing on the development of effective variable selection methods for neural network-based inferential models [11]–[13]. A variable selection method termed sequential backward multiplayer perceptron (SBS-MLP) was proposed in [14]. It has significant performance in variable selection and model accuracy. Recently, some other proposed algorithms have considerably less computational cost, but the improvement of models and variable selection accuracy are insignificant compared with SBS-MLP [15]. Battiti [16] developed a mutual information criterion to evaluate a set of candidate input variables and then select an informative subset to be used as input data for a neural ANN classifier. Estévez *et al.* [17] proposed an advanced variable selection method over Battiti's approach by introducing the average normalized mutual information as a measure of redundancy. The approach is called normalized mutual information feature selection (NMIFS). The simulation results on several artificial data sets and benchmark problems show that the NMIFS algorithm has better performance than Battiti's algorithm.

A new linear regression method for shrinkage and variable selection, named the least absolute shrinkage and selection operator (LASSO) [18], has emerged in recent years. This method minimizes the usual sum of squared errors by placing a bound on the sum of the absolute values of the coefficients. Similar to the subset selection, this approach can produce interpretable models, while it also exhibits the stability of ridge regression. A least angle regression (LARS) algorithm for solving the LASSO has also been developed [19]. The LARS established a connection between LASSO and forward stage wise regression, and was found to be computationally efficient. In addition, an adaptive LASSO, where adaptive weights were used for penalizing different coefficients, was developed in [20]. Radchenko and James [21] proposed a forward-LASSO by combining LASSO with forward selection, which can be used in both the linear regression and the generalized linear model domains. However, most of these LASSO-related algorithms focus on the variable selection and shrinkage for linear problems, and there are few works studying the nonlinear variable selection with LASSO.

In addition, Fan and Li [22] proposed a model shrinkage method via penalized likelihood with a smoothly clipped absolute deviation (SCAD) penalty function, and demonstrated the efficiency and rationality of the algorithm with simulation and asymptotic theory. In recently years, the SCAD was widely used in linear variable selection and regression problems [23], [24].

The primary contribution of this paper is the development of an effective input variable selection method for neural network that can be applied to describe nonlinear industrial processes. The proposed approach implements LASSO to conduct the accurate shrinkage of input weights of the MLP.

The remainder of this paper is organized as follows. In Section II, the theory of the proposed methodology is discussed. Section III presents numerical examples that illustrate the performance of the proposed approach. Some state-of-the-art neural network-based variable selection algorithms, including SBS-MLP [14] and NMIFS [17], are used to make comparisons with our approach. To illustrate the shrinkage performance of LASSO, another model shrinkage method, SCAD [22], is applied to conduct the shrinkage of input weights of the MLP. In Section IV, a concrete application of the proposed method and the prediction of kerosene quality in a crude distillation unit (CDU) is demonstrated. Finally, the conclusion is drawn in Section V.

II. Proposed Methodology

A. LASSO Algorithm for Linear Problems

Consider the usual linear regression model

$$y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \in \tag{1}$$

where $x_1, x_2, \ldots, x_p$ are p input variables, y is a response variable, $\beta = \{\beta_1, \beta_2, \ldots, \beta_p\}$ is the coefficient, and $\in$ is the random error.

Suppose that $X \in \mathbb{R}^{n\times p}$ is the input data matrix, in which each column denotes a candidate input variable, and $Y \in \mathbb{R}^n$ is a vector representing the response variable. Then, for given p input variables $x_1, x_2, \ldots, x_p$, the response y is predicted by

$$\hat{y} = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p. \tag{2}$$

The ordinary least squares (OLSs) estimate, $\hat{\beta} = \{\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p\}$, is obtained by minimizing the residual sum of squares, which is formulated as

$$\hat{\beta} = \text{argmin} \sum_{t=1}^{n} \left(y_t - \sum_{i=1}^{p} \beta_i x_i \right)^2. \tag{3}$$

The LASSO algorithm is proposed by introducing an extra penalty into (3), which is shown as [18]

$$\hat{\beta} = \text{argmin} \sum_{t=1}^{n} \left(y_t - \sum_{i=1}^{p} \beta_i x_i \right)^2 + \lambda \sum_{i=1}^{p} |\beta_i| \tag{4}$$

where $\lambda \sum_{i=1}^{p} |\beta_i|$ is called the LASSO penalty, and λ is a non-negative tuning parameter. The algorithm causes the $\hat{\beta}$ value to continuous shrink toward zero as the parameter λ increases. The coefficient $\hat{\beta} = \{\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p\}$ shrinks exactly to zero if λ is sufficiently large, which implies that all variables are eliminated. These shrinkages can often improve prediction accuracy because of the bias-variance tradeoff [18].

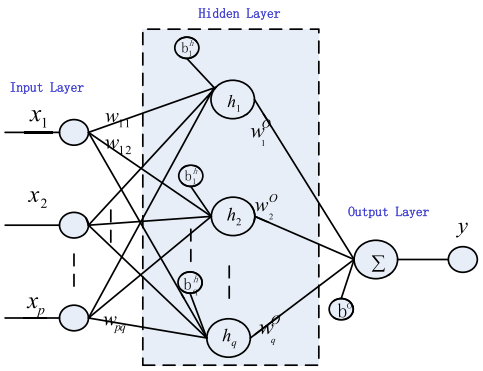

Fig. 1. Architecture of the MLP.

B. MLP Neural Network

MLP is a feedforward ANN that maps sets of input data onto a set of appropriate outputs. Among all the ANN structures, MLP is the most common in use and is very efficient in obtaining approximate models for extremely complex problems [25]. Besides, there are many other network structures, such as Elman network (ELM) [26], NARX [27], Cascade-correlation neural network [28], and self-organization map [29]. Every network structure has its advantage in certain applications.

This paper aims to developing an effective variable selection method for the MLP neural network of [25] with LASSO. Fig. 1 shows the structure of a three-layer MLP, which consists of an input layer, a hidden layer, and an output layer. Each layer consists of multiple neurons that are connected to neurons in adjacent layers.

Assume that the input variables of the network are given by the candidate variables $x = \{x_1, x_2, \ldots, x_p\}$ and the hidden layer has q nodes, denoted by $h = \{h_1, h_2, \ldots, h_q\}$. The weight $w_{ij} (i \in [1, p],\ j \in [1, q])$ represents the input weight between the input variable x_i and the jth hidden nodes h_j, while the bias of the jth neuron of the hidden layer is denoted by b_j^h. The output results of the jth neuron of the hidden layer, O_j^h, can be given by

$$O_j^h = f\left(\left(\sum_{i=1}^{p} w_{ij} x_j \right) + b_j^h \right) \tag{5}$$

where f denotes the activation function of the hidden layer.

Let weight $w_j^o (j \in [1, q])$ represent the jth output weight between the hidden layer and the output layer and b^o denote the bias of the output layer. g represents the activation function of the output layer. The input–output relationship of MLP is formulated as

$$y = g\left(\sum_{j=1}^{q} w_j^o f\left(\left(\sum_{i=1}^{p} w_{ij} x_j \right) + b_j^h \right) + b^o \right). \tag{6}$$

C. Integrate the LASSO Algorithm Into MLP

In this paper, we integrate the LASSO penalty into the neural network in order to achieve model shrinkage and variable selection. Equation (6) is reformulated by adding the parameter $\beta = \{\beta_1, \beta_2, \ldots, \beta_p\}$ in front of the input nodes of the MLP

$$y = g\left(\sum_{j=1}^{q} w_j^o f\left(\left(\sum_{i=1}^{p} \beta_i w_{ij} x_j\right) + b_j^h\right) + b^o\right). \tag{7}$$

In a similar way, (4) is also reformulated as

$$\hat{\beta} = \operatorname{argmin} \sum_{t=1}^{n} \left\{ y_t - g\left(\sum_{j=1}^{q} w_j^o f\left(\left(\sum_{i=1}^{p} \beta_i w_{ij} x_j\right) + b_j^h\right) + b^o\right)\right\}^2 + \lambda \sum_{i=1}^{p} |\beta_i|. \tag{8}$$

Apparently, (8) is a nonlinear quadratic minimization problem that can be solved using a trust region reflective optimization algorithm [30]. Following that, the new neural network can be obtained by replacing β with $\hat{\beta}$ in (8):

$$\hat{y} = g\left(\sum_{j=1}^{q} w_j^o f\left(\left(\sum_{i=1}^{p} \hat{\beta}_i w_{ij} x_j\right) + b_j^h\right) + b^o\right). \tag{9}$$

D. Determination of the Shrinkage Parameter

The selection of the shrinkage parameter λ is a key component of the proposed approach, because the choice of parameter significantly influences the performance of the algorithm. For a high value of λ, LASSO will always provide a null model, in which all input variables are deleted. On the other hand, in the case that $\lambda \to 0$, it becomes the standard OLS estimate. In order to select the best value for the parameter λ, the proposed approach performs an enumerative search within the domain $[\lambda_{\text{lb}}, \lambda_{\text{ub}}]$, where λ_{lb} is zero, and λ_{ub} is a sufficiently large value to ensure that all the values of $\hat{\beta}_i$ are equal to zero. Descriptions follow the model selection criterion and cross-validation (CV) strategy implemented in this approach.

1) Model Selection Criterion: Model selection techniques are helpful when it comes to selecting the best fitting model from a set of candidate models. An improved model selection criterion, named AICc [31], is employed for the task of model selection in this paper. The model selection criterion is formulated as

$$\text{AICc} = n \cdot lg\left(\frac{1}{n}\sum_{i=1}^{n_v} (\hat{y}_i - y_i)^2\right) + 2m + \frac{2m(m+1)}{n-m-1} \tag{10}$$

where n_v is the total number of validation data samples, m is the number of variables in the model, and y and $\hat{y}$ are the measured and predicted values of the output variable, respectively. In this paper, the predicted value $\hat{y}$ can be calculated by using (9).

Algorithm 1 K-fold CV for the determination of λ

Step 1. Initialize the bound domain of the parameter $\lambda \in [\lambda_{lb}, \lambda_{ub}]$, where $\lambda_{lb} = 0$ and λ_{ub} is a sufficiently large value that ensures all $\hat{\beta}_i$ are equal to zero. Set $\lambda = \lambda_{lb}$.

Step 2. Separate the entire dataset $S = \{X, Y\}$ into K disjoint sub-datasets $S_1, S_2, \ldots, S_K$.

Step 3. CV process

- **Step 3.1** For the sub-dataset S_k, employs the remaining $K-1$ sub-datasets to train a neural network.
- **Step 3.2** Introduce the LASSO penalty into the neural network, thus obtaining (7).
- **Step 3.3** Solve (8), and use (9) to obtain the new model.
- **Step 3.4** Calculate the AICc value for this model with dataset S_k, using (10).
- **Step 3.5** Set $k = k + 1$, and if $k < K$ goto **Step 3.1**
- **Step 3.6** Calculate the average AICc error for current λ.

Step 4. $\lambda \leftarrow \lambda + \Delta\lambda$, if $\lambda < \lambda_{ub}$, goto **Step 3**.

Step 5. Find the optimal λ with the minimum CV error.

2) Cross Validation: CV is considered to be the simplest and most widely used model validation method to minimize the prediction error [32]. In K-fold CV, the group of all samples is divided into K subgroups. In this method, a single subdata set is used as the validation data set, and the other $K-1$ subdata sets are used as the training data for the purpose of constructing the model. The procedure is repeated K times to validate each of the K subdata sets exactly once. Following the procedure, the K results can be averaged to achieve a single estimation. The process of the determination of λ is described in Algorithm 1.

E. Computational Flow of the Proposed Approach

In this paper, a new iterative backward deletion method for an MLP network is proposed, which introduces LASSO into an MLP, and is labeled as LASSO-MLP. At each iteration, the proposed LASSO-MLP trains a new network with the current data set, and shrinks $\hat{\beta}_i$ by invoking LASSO. Then, the input variables for which $\hat{\beta}_i = 0$ are deleted, and a new data set is constructed using the remaining variables. This process is repeated until the termination conditions are met, which comprise either a maximum number of iterations, or a state being reached where no further improvement is achieved in the model error. A flowchart of the proposed LASSO-MLP is shown in Fig. 2. It can also be described in Algorithm 2.

III. Simulation Results

A. Comparative Methodologies

In this section, the performance of the proposed nonlinear variable selection algorithm, abbreviated as LASSO-MLP, was investigated by artificial data set examples. Some state-of-the-art neural network-based variable selection algorithms, including SBS-MLP [14] and NMIFS [17], were used to make comparisons with our approach.

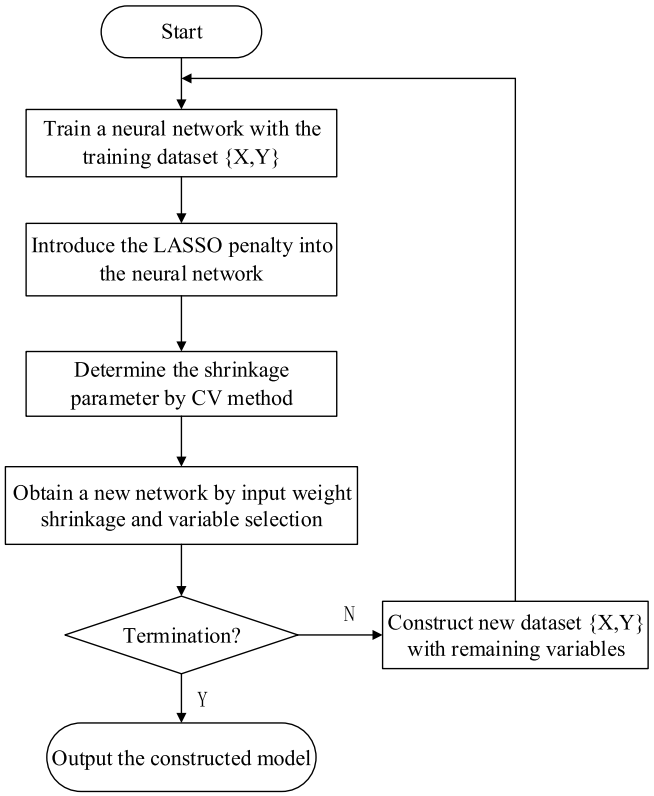

Fig. 2. Flowchart of the proposed LASSO-MLP.

Algorithm 2 LASSO-MLP

Step 1. Train or retrain a neural network with the training dataset $\{X, Y\}$

Step 2. Introduce the magnitude parameter β into the current neural network, and obtain (7), where p is the number of remaining variables.

Step 3. Determine the value of the parameter λ using K-fold CV, which is described in **Section II-D**.

Step 4. Solve (8) and obtain $\hat{\beta} = \{\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p\}$. Replace β with $\hat{\beta}$, and obtain the new neural network.

Step 5. Determine whether the termination condition is met. If not, delete the candidate variables from X which have $\hat{\beta}_i = 0$, and obtain a new training dataset $\{X', Y\}$. Let $\{X, Y\} = \{X', Y\}$, and then goto **Step 1.**

Step 6. Output the results.

In order to compare the variable selection performance of LASSO on different network architectures, we applied LASSO to shrink the input variables of ELM. ELM is a recurrent dynamic network with feedback in its architecture. The ELM structure employed in this paper has three layers: an input layer, a hidden layer, and an output layer, as shown in Fig. 3. Each layer consists of multiple nodes that are connected to nodes in the adjacent layers. Besides, there is a set of context units in ELM, which is different from MLP. The weights of the connections between the hidden layer and the context units are fixed as one, i.e., $c_i = 1, \forall i$. Therefore, these context units can maintain duplicates of the previous values of the hidden units.

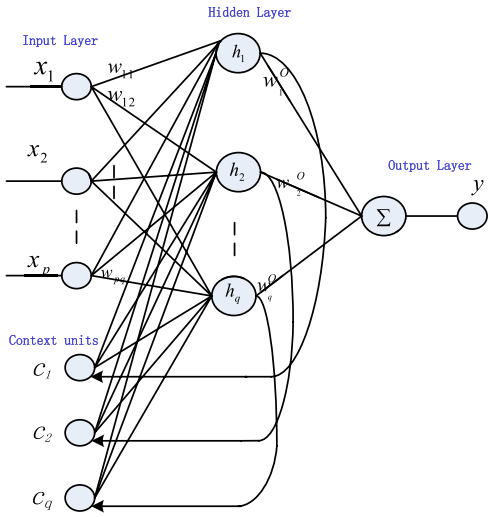

Fig. 3. Architecture of the ELM.

Let $x(t) = \{x_1(t), x_2(t), \ldots, x_p(t)\}$ denote the tth row of the input data matrix X. In the ELM, the output of the jth neuron of the hidden layer at time t, $O_j^h(t)$ can be given by

$$O_j^h(t) = f\left(\sum_{i=1}^{p} w_{ij} x_j(t) + \sum_{j=1}^{q} O_j^h(t-1) + b_j^h\right). \quad (11)$$

Then, the ELM can be formulated as

$$y(t) = g\left(\sum_{i=1}^{q} w_j^o f\left(\sum_{i=1}^{p} w_{ij} x_j(t) + \sum_{j=1}^{q} O_j^h(t-1) + b_j^h\right) + b^o\right). \quad (12)$$

Similar to LASSO-MLP, the shrinkage on ELM is performed by LASSO by

$$\hat{\beta} = \text{argmin} \sum_{t=1}^{n} \left\{ y(t) - g\left(\sum_{i=1}^{q} w_j^o f\left(\sum_{i=1}^{p} w_{ij} x_j(t) + \sum_{j=1}^{q} O_j^h(t-1) + b_j^h\right) + b^o\right)\right\}^2 + \lambda \sum_{i=1}^{p} |\beta_i|. \quad (13)$$

Thus, a new nonlinear variable selection method on ELM, labeled LASSO-ELM, is designed to make comparisons with LASSO-MLP.

Furthermore, we illustrated the shrinkage performance of LASSO with the application of another model shrinkage method. For a linear model selection problem of (1), $x = \{x_1, x_2, \ldots, x_p\}$ is a set of p input variables, y is a response variable, and $\beta = \{\beta_1, \beta_2, \ldots, \beta_p\}$ is the coefficient. The SCAD algorithm is formulated as

$$\hat{\beta} = \text{argmin} \sum_{t=1}^{n} \left(y_t - \sum_{i=1}^{p} \beta_i x_i\right)^2 + n \sum_{i=1}^{p} \rho_\lambda(|\beta_i|) \quad (14)$$

in which $\rho_\lambda|\cdot|$ is the penalty function and defined as

$$\rho_\lambda(\theta) = \begin{cases} \lambda\theta, & \theta \le \lambda \\ -(\theta^2 - 2a\lambda\theta + \lambda^2)/2(a-1), & \lambda \le \theta \le a\lambda \\ (a+1)\lambda^2/2, & \theta > a\lambda \end{cases} \tag{15}$$

where a and λ are shrink parameters that can determine the shrinkage degree of the SCAD algorithm. Therefore, the performance of SCAD highly depends on appropriately choosing the tuning parameter. The best pair (λ, a) could be obtained using 2-D grids search with a K-fold CV method [32] mentioned above. Similar to LASSO, the SCAD algorithm can be extended to input variable selection of MLP. The new nonlinear variable selection algorithm, abbreviated as SCAD-MLP, is formulated by modifying the penalty function of (8)

$$\hat{\beta} = \text{argmin} \sum_{t=1}^{n} \left\{ y_t - g\left(\sum_{j=1}^{q} w_j^o f\left(\left(\sum_{i=1}^{p} \beta_i w_{ij} x_j \right) + b_j^h \right) + b^o \right) \right\}^2 + n \sum_{i=1}^{p} \rho_\lambda(|\beta_i|) \tag{16}$$

with the function $\rho_\lambda|\cdot|$ of (15).

Equation (16) is a nonlinear quadratic minimization problem that can be solved using the trust region reflective optimization algorithm [30]. The rest of SCAD-MLP, including model selection criterion, K-fold CV, and computation flow, are the same as those of LASSO-MLP.

B. Experiment Setting

Each of these MLP-based variable selection methods was carried out in the same experimental setting. All methodologies mentioned in the experiments have the same input variables. They had the same MLP network structure with three layers. The activation function of the hidden layer was a hyperbolic tangent function, and that of the output layer was a linear function. The training function was a standard Back propagation algorithm.

The performance of these algorithms was evaluated by using five statistics as follows.

1) Root mean square error (RMSE) of prediction, which is defined as

$$\text{RMSE} = \sqrt{\frac{1}{n_t} \sum_{k=1}^{n_t} (\hat{y}(k) - y(k))^2} \tag{17}$$

where $\hat{y}$ and y are the measured and predicted values, respectively, and n_t is the total number of test data samples.

2) *Coefficient of Determination (R^2):* The square of the sample correlation coefficient between the outcomes and their predicted values.

3) *Model Size (MS):* The number of nonzero elements in the final $\beta^*(s) = \{\beta_1^*, \beta_2^*, \ldots, \beta_p^*\}$, that is, the remaining variables in the candidate variables pool following the model construction.

TABLE I
STATISTICAL RESULTS FOR FRIEDMAN DATA SET WITH TEN INPUT VARIABLES

	LASSO - MLP	LASSO-ELM	SCAD-MLP	NMIFS	SBS-MLP	MLP
RMSE	0.143	0.146	0.150	0.149	0.152	0.173
R^2	0.976	0.972	0.965	0.970	0.962	0.941
M.S.	5.45	5.63	5.87	5.79	6.07	10
FS+	0.90	1.16	1.43	1.25	1.55	5
FS-	0.45	0.53	0.47	0.47	0.48	0

TABLE II
STATISTICAL RESULTS FOR FRIEDMAN DATA SET WITH 50 INPUT VARIABLES

	LASSO-MLP	LASSO-ELM	SCAD-MLP	NMIFS	SBS-MLP	MLP
RMSE	0.195	0.216	0.223	0.219	0.233	0.983
R^2	0.935	0.928	0.925	0.926	0.922	0.262
M.S.	6.51	6.93	7.11	7.08	7.62	50
FS+	2.12	2.56	2.78	2.72	3.28	45
FS-	0.61	0.63	0.67	0.64	0.66	0

4) *False Positive Selection (FS+):* The number of irrelative variables that are selected into the model

$$\text{FS+} = \underset{i \in \{6,7,\ldots,p\}}{\text{Count}} \hat{\beta}_i \neq 0. \tag{18}$$

5) *False Negative Selection (FS−):* The number of relative variables that are not selected into the model

$$\text{FS-} = \underset{i \in \{1,2,\ldots,5\}}{\text{Count}} \hat{\beta}_i = 0. \tag{19}$$

C. Friedman Data Set

We used an artificial data set designed in [33] to analyze the performance of the proposed approach. The data set consisted of ten input variables, with each variable generated with uniform distribution over the range [0, 1]. The response variable was obtained by

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \tag{20}$$

where ϵ is a Gaussian noise with zero mean and unit variance.

Table I shows the statistical performance of five algorithms over 100 runs. It can be seen from Table I that the LASSO-MLP produced the best performance, and RMSE, R^2, and MS demonstrated an obvious improvement compared with the other approaches. The FS− results of these variable selection methods were considerably close, whereas the FS+ results of the other approaches were apparently worse than those of LASSO-MLP. The results indicated that the LASSO-MLP had a lower probability of selecting irrelative variables, which resulted in a more accurate model.

Furthermore, a larger Friedman data set was applied in order to test the performance of our approach. This data set had 50 input variables and 1 output variable, which was calculated by (20), meaning that, in this model, there were 45 irrelative input variables requiring deletion. Table II shows the statistical results obtained by each method

TABLE III
STATISTICAL RESULTS FOR FRIEDMAN DATA SET WITH HIGH CROSS CORRELATION

	LASSO - MLP	LASSO-ELM	SCAD-MLP	NMIFS	SBS-MLP	MLP
RMSE	0.185	0.198	0.203	0.209	0.216	0.593
R^2	0.986	0.970	0.963	0.956	0.953	0.670
M.S.	7.06	7.95	8.50	8.85	9.11	50
FS+	3.39	4.53	5.12	5.51	5.78	45
FS-	1.33	1.58	1.62	1.66	1.67	0

over 100 runs. The performance of MLP was considerably poor in comparison with the other variable selection methods. It appears that the presence of many irrelative input variables in the model resulted in the occurrence of overfitting. Effective variable selection methods could eliminate those irrelative variables, reduce the complexity of the process, and avoid the occurrence of the overfitting. Even in the large-scale problems, LASSO-MLP performed consistently better than other algorithms.

The original Friedman data sets have few cross correlations between the relative and irrelative input variables. In order to further investigate the performance of our approach, we revised the Friedman data set with 50 input variables by introducing the cross correlation into it. At first, the first five columns of the data set, i.e., the relative input variables remained the same as before. Then, for each relative input variable $x_i (i = 1, 2, \ldots, 5)$, nine high correlative columns were generated by the formula [34]

$$x_{ij} = \begin{cases} 0.9x_i + 0.1U(0,1), & j = 1 \\ 0.9x_{i \cdot j-1} + 0.1U(0,1), & j = 2, 3, \ldots, 9 \end{cases} \quad (21)$$

where $U(0, 1)$ is a vector of random numbers between 0 and 1. Following that, all the values of x_i and x_{ij} were combined together to create a new data set that had high cross correlation. This cross correlation between the input variables will increase the difficulty of selecting correct variables from the candidate variables pool.

The statistical performance is shown in Table III. Similar to the previous cases, our approach performed consistently better than other approaches. Compared with Table II, the FS+ result of LASSO-MLP increased from 2.12 to 3.39, that of LASSO-ELM increased from 2.56 to 4.53, that of SCAD-MLP increased from 2.78 to 5.12, that of NMIFS increased from 2.72 to 5.51, and that of SBS-MLP increased from 3.28 to 5.78. The increase rates of FS+ with other approaches were considerably higher than that with LASSO-MLP, which means that these methodologies are more likely to make wrong selection than LASSO-MLP when the cross correlation of input variables increases.

D. Pymdyn32nh Data Set

Moreover, we used one of Pumadyn family of data sets that came from the Delve library [35] to illustrate the performance of our approach. The data sets are the realistic simulations of the dynamics of a Puma 560 robotic arm. The task of the problem is to predict the angular acceleration of one of the robotic links. The input variables include angular positions, velocities, and torques of the arm. Each data set in the family has a unique combination of three attributes: dimensionality (8 or 32 input variables), nonlinearity (linear or nonlinear), and output noise (moderate or high). A data set with 32 input variables, nonlinear, and high output noise, marked as Pymdyn32nh, was chosen to make comparisons among all these approaches.

TABLE IV
STATISTICAL RESULTS FOR Pymdyn32nh DATA SET

	LASSO-MLP	LASSO-ELM	SCAD-MLP	NMIFS	SBS-MLP	MLP
RMSE	0.0061	0. 0072	0. 0076	0.0083	0.0093	0.288
R^2	0.982	0.975	0.971	0.968	0.963	0.760
M.S.	5.60	6.12	6.38	6.73	7.75	32

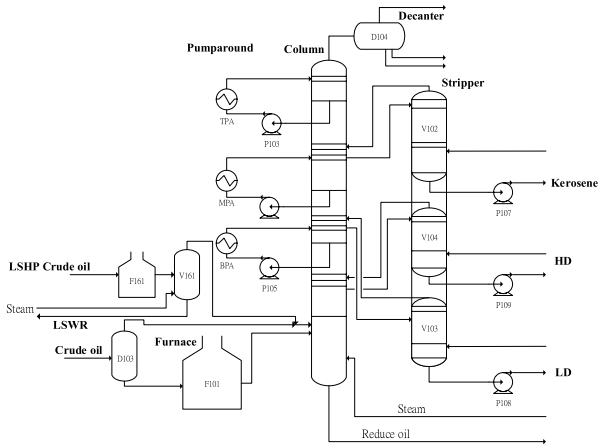

Fig. 4. Schematic flow diagram of the CDU system.

The experimental data set had 1024 instances, and was divided into two subdata sets. The first 512 instances were training data, and other instances were testing data. Table IV shows the statistical results for Pymdyn32nh data set. The MLP with no variable selection constructed obvious poor model compared with other algorithms, because too many irrelative variables included in the model would lead to the phenomenon of overfitting. In addition, the LASSO-MLP had best results compared with other approaches in RMSE, R^2, and MS. These results indicate that the performance of our proposed algorithm was consistently better than others' when high noise was imposed on the output variable.

IV. APPLICATION TO KEROSENE QUALITY ESTIMATION IN CRUDE DISTILLATION UNIT

CDU has been widely used by chemical and petroleum industries to separate incoming crude oil into its component fractions, by exploiting differences in their boiling points. Fig. 4 shows a schematic flow diagram representing a real crude oil distillation unit, which consists of a crude distillation column at atmospheric pressure as well as a furnace, stripper unit, pumparounds, and a decanter. The incoming crude oil can fall into one of two categories: crude oil with high viscosity

and common crude oil. The crude oil with high viscosity is heated and predistillated in the Low sulfur high paraffin section (F161 and V161), and the steam obtained in this process is imported into the main column (V101). Common crude oil is preheated by D103, and the resulting steam is imported into V101 directly. In addition, the liquid obtained from D103 flows into furnace (F101), where it is heated to a temperature of ∼340 °C. Following that, the steam obtained from the liquid in F101 is imported into V101. There are three pumparounds at the top of the CDU: the top pumparound, middle pumparound (MPA), and bottom pumparound (BPA). These pumparounds control the temperature of the steam by connecting to either one, or several heat exchangers. They also keep the vapor loading of the column at a stable rate, in addition to regulating the amount of liquid traffic in the column to achieve effective fractionation.

As shown in the flow diagram, the top distillate fraction extracted in V101 is naphtha. The residual oil is discharged from the bottom of V101. The products of the side strippers are kerosene, light diesel, and heavy diesel, respectively. Kerosene is a considerably important product, which can be used for burning in lamps and domestic heaters, and also as a fuel for jets and turboprop aircraft engines.

The quality of the kerosene would be verified by laboratory assays once a day, with the American Society of Testing Materials (ASTM) method D86. The distillation endpoint of a product was defined as the maximum reading of the temperature sensor obtained during the test. However, a 95% distillation endpoint (D95) was commonly used, as the endpoint was difficult to measure with a good level of repeatability [36].

In [37], the nonlinearity of the CDU process was analyzed, and a neural network-based soft sensor for the prediction of the kerosene property was developed. A soft sensor for the estimating the kerosene property by integrating a neural network with Principal component analysis was developed in [38].

This paper focuses on the development of an inferential model for the estimation of the kerosene properties by employing LASSO-MLP. The data were collected on a daily basis in 2013. Altogether, there were 361 samples, from which the first 240 samples were training data, and the other 121 samples were testing data. The D95 for the kerosene was determined by using laboratory assays with an ASTM D86 standard test method. The input data set was composed of the average values of 60-min data, before the kerosene sample was taken every day.

Fig. 5 shows the data characteristic of the kerosene D95, using laboratory assays in the CDU process. The kerosene D95 shown frequent and significant variations, implying that the quality of the kerosene was very unstable during that process. In addition, the verification of kerosene quality using laboratory assays proved to be significantly time-consuming. Therefore, it was shown that the development of a reliable inferential model for the estimation of kerosene would be necessary for real-time monitoring and control in the CDU process.

The data set consisted of 25 candidate input variables and one target output variable, namely, kerosene D95. The input variables are shown in Table V.

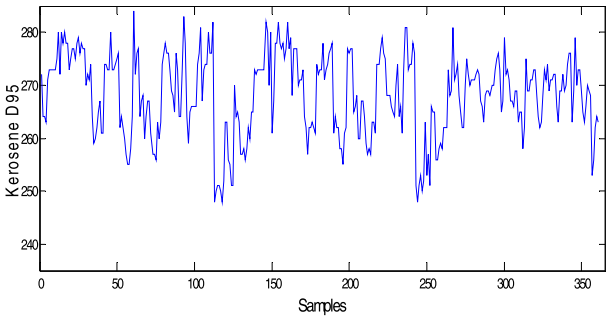

Fig. 5. Data characteristic of the kerosene D95 in CDU process.

TABLE V
CANDIDATE VARIABLES FOR THE KEROSENE D95 PREDICTION

Variable	Quantity	Unit
1	D103 Input Flow Rate	Nm3/h
2	D103 Output Flow Rate	Nm3/h
3	TPA Flow Rate	Nm3/h
4	TPA Temperature	Nm3/h
5	MPA Flow Rate	Nm3/h
6	MPA temperature	℃
7	BPA Flow Rate	Nm3/h
8	BPA Temperature	℃
9	V101 Column Top Pressure	mmAq
10	V101 Column Bottom Pressure	mmAq
11	V102 Bottom Flow Temperature	℃
12	V103 Bottom Flow Temperature	℃
13	V104 Bottom Flow Temperature	℃
14	V161 Top Output Pressure	mmAq
15	V161 Top Output Temperature	℃
16	V161 Bottom Output Flow rate	Nm3/h
17	F161 Output Temperature	℃
18	Furnace Output Temperature	℃
19	F161 Input Flow Rate	Nm3/h
20	V101 SHLP Steam Temperature	℃
21	Steam Flow rate into V101 Bottom	Nm3/h
22	V161 SHLP Steam Flow Rate	Nm3/h
23	Distillation Light Diesel Flow Rate	Nm3/h
24	Distillation Kerosene Flow Rate	Nm3/h
25	Distillation High Diesel Flow Rate	Nm3/h

Fig. 6 shows the correlation matrix of all the variables, in which variable 26 represents kerosene D95. It can be seen from Fig. 6 that there is a high degree of cross correlation between the input variables, making it very difficult to select the correct variables from the candidate variable pool.

All algorithms discussed in this paper used the same MLP network architecture with three layers. Initially, a tenfold CV method was employed to determine the number of hidden nodes. The network architecture with four hidden nodes had the best RMSE after CV. Consequently, the MLP network architecture of these approaches for the kerosene D95 prediction was 25-4-1.

Table VI shows the statistical results of these different approaches over 100 runs. It is clearly that the LASSO-MLP had a better prediction accuracy and smaller MS than the other algorithms.

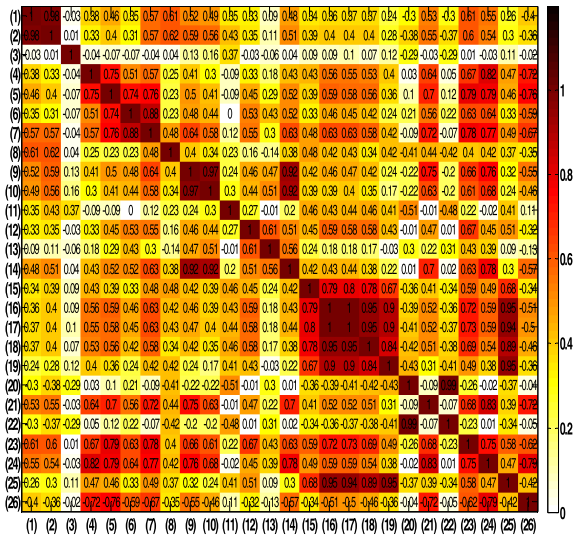

Fig. 6. Correlation matrix of all the variables.

TABLE VI
STATISTICAL RESULTS FOR KEROSENE D95

	LASSO-MLP	LASSO-ELM	SCAD-MLP	NMIFS	SBS-MLP	MLP
RMSE	3.68	3.83	3.92	4.03	4.27	4.95
R^2	0.86	0.84	0.83	0.82	0.79	0.73
M.S.	10.18	10.62	10.87	11.02	11.75	25

Fig. 7 shows a comparison between the kerosene D95 as verified by laboratory assays and the predictions given by different approaches. It appears that the model constructed by LASSO-MLP could successfully follow the dynamics of the kerosene D95. The detailed error distributions of the three approaches are shown in Fig. 8. According to the field requirement, the estimation whose error is within ±4 °C is a good estimation. Fig. 8 shows that most of estimation with LASSO-MLP can achieve this accuracy, and our approach had the best performance in estimation error among the six methods.

Figs. 9–13 show the chart of the probabilities of selection for each candidate input variables using different approaches, where the probability of selection was calculated by counting the selected numbers of input variables over 100 runs. Fig. 9 shows that the variables 24, 13, 5, and 7 are chosen on over 80% of occasions by LASSO-MLP, whereas the other variables are selected at a frequency of below 60%. In Fig. 10, there are six variables with a selection probability over 60% and eight variables with a selection probability over 50%. Meanwhile, the numbers of those variables in Fig. 11 are six and eight; the numbers of those variables in Fig. 12 are six and nine; the numbers of those variables in Fig. 13 are six and eleven. These numbers are obviously higher than those in Fig. 9, which demonstrates that the LASSO-MLP can build a more compact model than other algorithms.

In the information theory, the entropy is used as an assessment of the uncertainty in a random variable [39]. In this paper, entropy was applied in order to make a certainty comparison

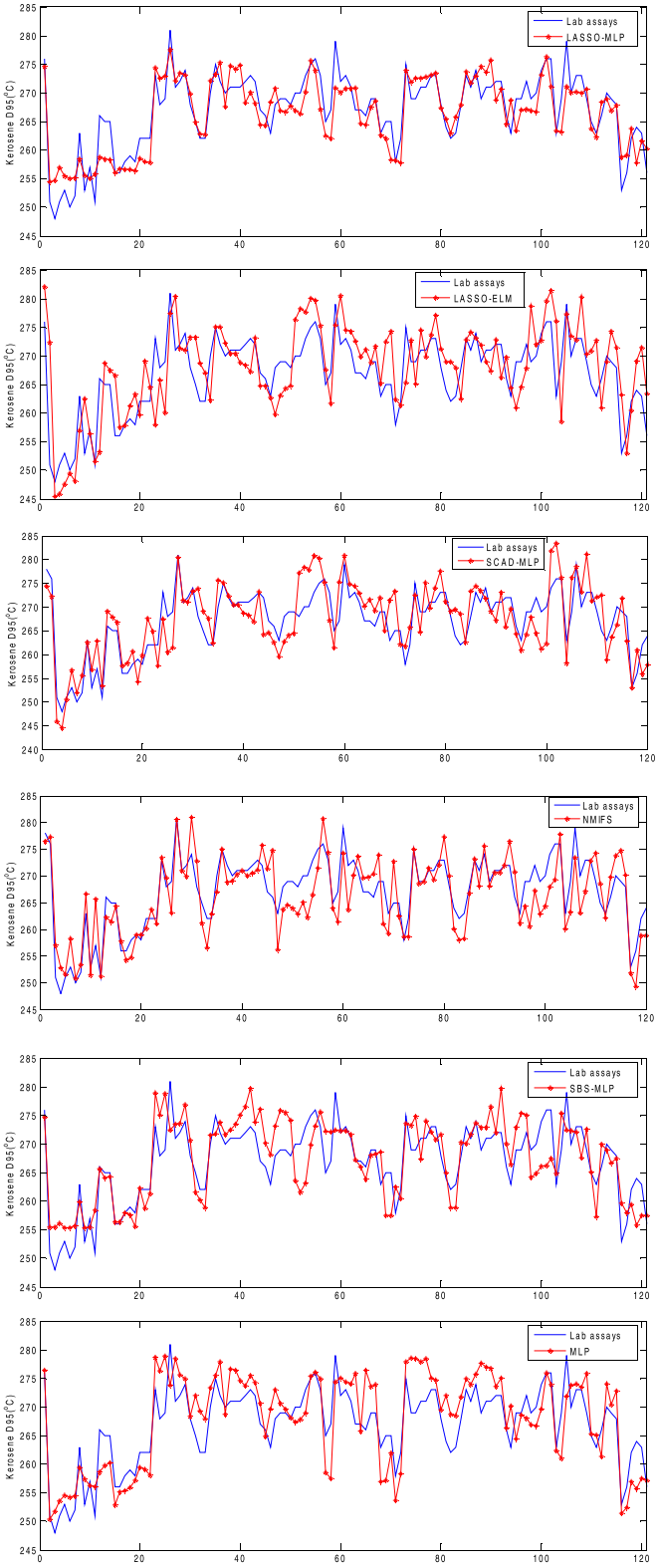

Fig. 7. D95 with lab assays and different algorithms.

between the proposed approach and the other approaches. The entropy formula is given as

$$H(X) = -\sum_{i=1}^{25}\{p(x_i)\log p(x_i) + ((1-p(x_i))\log(1-p(x_i))\} \tag{22}$$

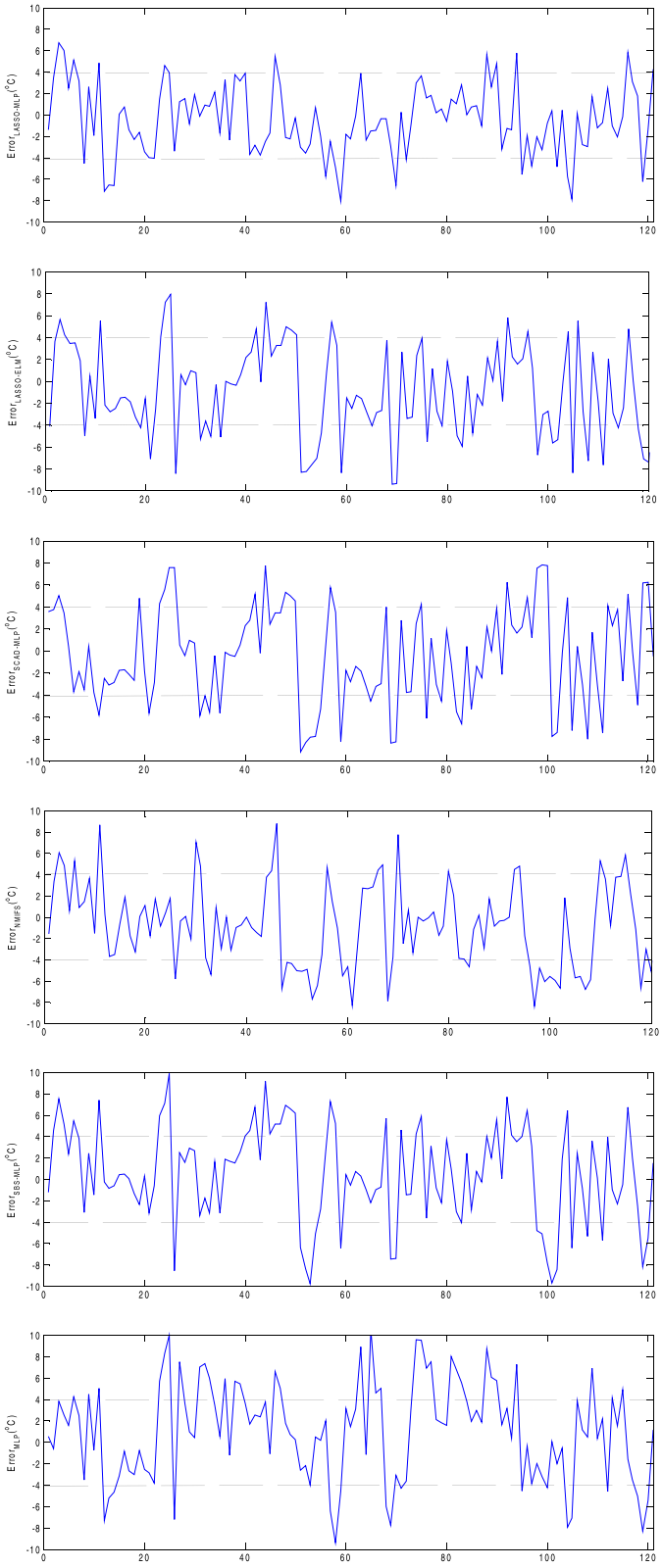

Fig. 8. Detailed error comparisons of different algorithms.

where $p(x_i)$ is the selection probability of the variable i. The entropies of LASSO-MLP, LASSO-ELM, SCAD-MLP, NMIFS, and SBS-MLP were 19.45, 20.03, 20.38, 20.51, and 21.32, respectively, which demonstrated the higher certainty and stability of LASSO-MLP in variable selection.

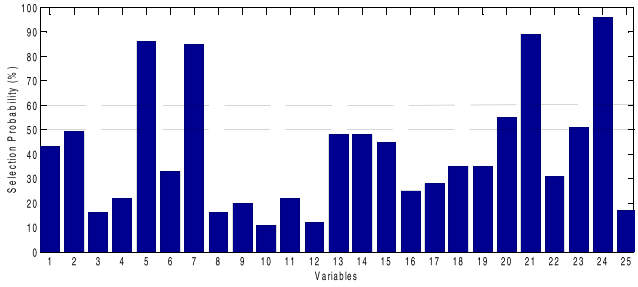

Fig. 9. Selection probability of CDU process variables with LASSO-MLP.

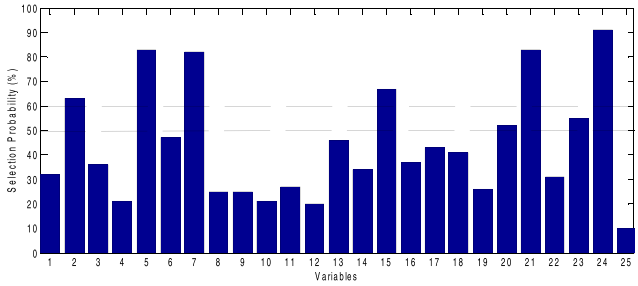

Fig. 10. Selection probability of CDU process variables with LASSO-ELM.

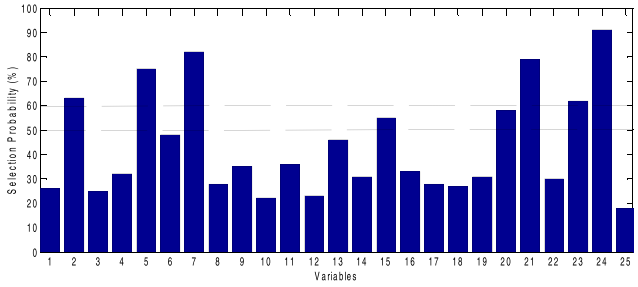

Fig. 11. Selection probability of CDU process variables with SCAD-MLP.

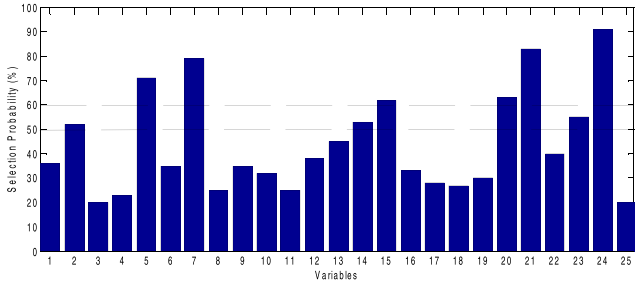

Fig. 12. Selection probability of CDU process variables with NMIFS.

Fig. 9 shows that variables 24, 13, 5, and 7 are very important for the CDU process. In the practical CDU process, the field operators preferred are those that regulate the flow rate, as controlling of the flow rate is considerably more convenient than controlling the temperature. Variable 24, the distillation kerosene flow rate, had the highest selection probability in the model, as shown in Fig. 9. That result is consistent with experiences from the field. The kerosene product has a lower boiling point and density if the kerosene flow rate is lower. On the contrary, it has a higher boiling point and greater

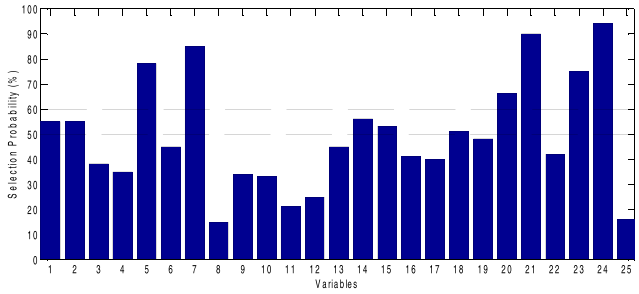

Fig. 13. Selection probability of CDU process variables with SBS-MLP.

density if the kerosene flow rate is increased. In addition, the steam entering the bottom of V101 introduces heat energy into the column V101, which could influence the distillation process of the CDU. For that reason, the flow rate of the bottom steam, variable 21, had the second highest selection probability in the model. Variables 5 and 7 were the MPA and BPA flow rates, which had the third and fourth highest selection probabilities, respectively. This is because these two variables regulated the loading of the oil vapor and the amount of liquid traffic in the column for effective distillation.

V. Conclusion

The development of an effective variable selection approach is challenging, because a large number of predictor variables displaying a high level of cross correlation are usually inherent in tangible industrial processes. A nonlinear variable selection method for an MLP neural network was proposed in this paper. The proposed approach introduces the LASSO penalty into the general MLP, and carries out shrinkage on the input weights of MLP in order to achieve accurate variable selections. The proposed LASSO-MLP combines the MLP's advantage of describing nonlinear process with the superior accuracy of variable selection that is provided by LASSO. The superiority of the proposed LASSO-MLP over other state-of-the-art variable selection methods was demonstrated with artificial examples as well as a concrete industrial application to predict the kerosene D95 in a CDU. In addition, the successful design and the application of LASSO on MLP are very inspiring for implementing appropriate model shrinkage to other network structures.

References

[1] G. P. Liu, *Nonlinear Identification and Control: A Neural Network Approach*. London, UK: Springer, 2001.

[2] S. Bhartiya and J. R. Whiteley, "Development of inferential measurements using neural networks," *ISA Trans.*, vol. 40, pp. 307–323, Sep. 2001.

[3] P. Dufour, S. Bhartiya, P. S. Dhurjati, and F. J. Doyle, III, "Neural network-based software sensor: Training set design and application to a continuous pulp digester," *Control Eng. Pract.*, vol. 13, no. 2, pp. 135–143, 2005.

[4] L. Fortuna, S. Graziani, and M. G. Xibilia, "Soft sensors for product quality monitoring in debutanizer distillation columns," *Control Eng. Pract.*, vol. 13, no. 4, pp. 499–508, 2005.

[5] J. C. B. Gonzaga, L. A. C. Meleiro, C. Kiang, and R. M. Filho, "ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process," *Comput. Chem. Eng.*, vol. 33, no. 1, pp. 43–49, 2009.

[6] A. Rani, V. Singh, and J. R. P. Gupta, "Development of soft sensor for neural network based control of distillation column," *ISA Trans.*, vol. 52, no. 3, pp. 438–449, 2013.

[7] J. A. Rodger, "A fuzzy nearest neighbor neural network statistical model for predicting demand for natural gas and energy cost savings in public buildings," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1813–1829, 2014.

[8] W. Yan, "Toward automatic time-series forecasting using neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1028–1039, Jul. 2012.

[9] J. Zhang, "Inferential estimation of polymer quality using bootstrap aggregated neural networks," *Neural Netw.*, vol. 12, no. 6, pp. 927–938, 1999.

[10] C. M. Andersen and R. Bro, "Variable selection in regression—A tutorial," *J. Chemometrics*, vol. 24, pp. 728–737, Nov./Dec. 2010.

[11] R. Chakraborty and N. R. Pal, "Feature selection using a neural framework with controlled redundancy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 35–50, Jan. 2014.

[12] E. Fock, "Global sensitivity analysis approach for input selection and system identification purposes—A new framework for feedforward neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1484–1495, Aug. 2014.

[13] J. A. Pérez-Benitez and L. R. Padovese, "Feature selection and neural network for analysis of microstructural changes in magnetic materials," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10547–10553, 2011.

[14] E. Romero and J. M. Sopena, "Performing feature selection with multilayer perceptrons," *IEEE Trans. Neural Netw.*, vol. 19, no. 3, pp. 431–441, Mar. 2008.

[15] F. A. A. Souza, R. Araújo, T. Matias, and J. Mendes, "A multilayer-perceptron based method for variable selection in soft sensor design," *J. Process Control*, vol. 23, pp. 1371–1378, Nov. 2013.

[16] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[17] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, "Normalized mutual information feature selection," *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[18] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.

[19] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.

[20] H. Zou, "The adaptive lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.

[21] P. Radchenko and G. M. James, "Improved variable selection with forward-lasso adaptive shrinkage," *Ann. Appl. Statist.*, vol. 5, no. 1, pp. 427–448, 2011.

[22] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 96, no. 456, pp. 1348–1360, 2001.

[23] J. Fan and J. Lv, "A selective overview of variable selection in high dimensional feature space," *Statist. Sinica*, vol. 20, no. 1, pp. 101–148, 2010.

[24] Y. Kim, H. Choi, and H.-S. Oh, "Smoothly clipped absolute deviation on high dimensions," *J. Amer. Statist. Assoc.*, vol. 103, no. 484, pp. 1665–1673, 2008.

[25] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.

[26] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

[27] J. M. P. Menezes, Jr., and G. A. Barreto, "Long-term time series prediction with the NARX network: An empirical evaluation," *Neurocomputing*, vol. 71, pp. 3335–3343, Oct. 2008.

[28] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," Tech. Rep. CMU-CS-90-100, Dept. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, USA, 1990.

[29] P. Demartines and J. Herault, "Curvilinear component analysis: A self-organizing neural network for nonlinear mapping of data sets," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 148–154, Jan. 1997.

[30] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM J. Optim.*, vol. 6, no. 2, pp. 418–445, 1996.

[31] K. P. Burnham and D. R. Anderson, *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. New York, NY, USA: Springer, 2002.

[32] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. 14th IJCAI*, 1995, pp. 1137–1143.

[33] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Statist.*, vol. 19, no. 1, pp. 1–67, 1991.
[34] H. Kaneko and K. Funatsu, "Nonlinear regression method with variable region selection and application to soft sensors," *Chemometrics Intell. Lab. Syst.*, vol. 121, pp. 26–32, Feb. 2013.
[35] C. E. Rasmussen, R. M. Neal, G. Hinton, D. V. Camp, M. Revow, and Z. Ghahramani. (1998). *Data for Evaluating Learning in Valid Experiments*. [Online]. Available: http://www.cs.utoronto.ca/~delve/data/datasets.html
[36] I. Mohler, Z. U. Andrijic, N. Bolf, and G. Galinec, "Distillation end point estimation in diesel fuel production," *Chem. Biochem. Eng. Quart.*, vol. 27, no. 2, pp. 125–132, 2013.
[37] N. Bolf, G. Galinec, and M. Ivandić, "Soft sensors for kerosene properties estimation and control in crude distillation unit," *Chem. Biochem. Eng. Quart.*, vol. 23, no. 3, pp. 277–286, 2009.
[38] R. Caponetto, G. Dongola, A. Gallo, and M. G. Xibilia, "FPGA based soft sensor for the estimation of the kerosene freezing point," in *Proc. IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Jul. 2009, pp. 228–236.
[39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY, USA: Wiley, 2012.

Kai Sun received the B.S. and M.S. degrees from Shandong University, Jinan, China, in 2000 and 2003, respectively, and the Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2009.

He is currently an Associate Professor with the Department of Automation, Qilu University of Technology, Jinan. His current research interests include scheduling and optimization, complex system modeling, and process control.

Shao-Hsuan Huang received the B.S. degree in chemical engineering from National Chung Cheng University, Chiayi, Taiwan, and the M.S. degree in chemical engineering from National Tsing Hua University, Hsinchu, Taiwan.

David Shan-Hill Wong received the B.S. degree in chemical engineering from the California Institute of Technology, Pasadena, CA, USA, in 1978, and the Ph.D. degree from the University of Delaware, Delaware, NY, USA, in 1982.

He has been a Professor with the Department of Chemical Engineering, National Tsing Hua University, Hsinchu, Taiwan, since 1983. His current research interests include process system engineering which includes design and control of energy-efficient separation processes and advanced process control in batch-based manufacturing.

Shi-Shang Jang received the M.S. degree in chemical engineering from National Taiwan University, Taipei, Taiwan, and the Ph.D. degree in chemical engineering from Washington University in St. Louis, St. Louis, MO, USA.

He is an expert in the area of process control, data-driven methods, and optimizing control. He has been a Professor with the Department of Chemical Engineering, National Tsing Hua University, since 1992, and was the chairman of the department from 2000 to 2004.