sensors

MDPI

Article

An Improved Normalized Mutual Information Variable Selection Algorithm for Neural Network-Based Soft Sensors

Kai Sun [1,*], Pengxin Tian [1], Huanning Qi [1], Fengying Ma [1] and Genke Yang [2,3,*]

[1] School of Electrical Engineering and Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China; pt_shandong@126.com (P.T.); qhning2006@163.com (H.Q.); fengyma@163.com (F.M.)

[2] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

[3] Ningbo Artificial Intelligence Institute, Shanghai Jiao Tong University, Ningbo 315000, China

[*] Correspondence: sunkai79@qlu.edu.cn (K.S.); gkyang@sjtu.edu.cn (G.Y.);
Tel.: +86-15269190537 (K.S.); +86-13651869523 (G.Y.)

Received: 1 October 2019; Accepted: 2 December 2019; Published: 5 December 2019

Abstract: In this paper, normalized mutual information feature selection (NMIFS) and tabu search (TS) are integrated to develop a new variable selection algorithm for soft sensors. NMIFS is applied to select influential variables contributing to the output variable and avoids selecting redundant variables by calculating mutual information (MI). A TS based strategy is designed to prevent NMIFS from falling into a local optimal solution. The proposed algorithm performs the variable selection by combining the entropy information and MI and validating error information of artificial neural networks (ANNs); therefore, it has advantages over previous MI-based variable selection algorithms. Several simulation datasets with different scales, correlations and noise parameters are implemented to demonstrate the performance of the proposed algorithm. A set of actual production data from a power plant is also used to check the performance of these algorithms. The experiments showed that the developed variable selection algorithm presents better model accuracy with fewer selected variables, compared with other state-of-the-art methods. The application of this algorithm to soft sensors can achieve reliable results.

Keywords: soft sensor; variable selection; neural network; mutual information; tabu search

1. Introduction

In real industrial processes, many variables cannot be measured at all or frequently enough with traditional instrumentation, owing to the technological or economical constraints of insufficient space, poor environmental conditions, extreme operating conditions, etc. Soft sensors provide valuable solutions for estimating these process variables through inference modeling with other easily measured variables. Paulsson, et al. [1] applied a soft sensor to a microbial recombinant protein production process in a bioreactor by exploiting bio-calorimetric methodology and obtained better results compared to other researchers. Chen, et al. [2] proposed a local soft-sensing method based on real-time correlation to predict silicon content, and better prediction performance of the method was validated on the online silicon content prediction. Kaneko, et al. [3] applied a soft sensor to the fault detection of chemical plants in order to accurately detect faults in the production process. Ahmed, et al. [4] successfully predicted the change of melt index during the production of high-density polyethylene using soft sensor technology. Wang, et al. [5] developed and applied a soft sensor to a refining process for quality prediction and achieved a high prediction accuracy. Xing, et al. [6] proposed a soft sensor for silicon content in industrial blast furnaces based on bagging local semi-supervised models (BLSM), the application results showed that BLSM had better prediction performance compared to other supervised soft sensors. Yuan, et al. [7] developed a novel variable-wise weighted stacked auto encoder (VW-SAE) for a soft sensor, and industrial data-based experiments showed that the proposed VW-SAE could present better prediction performance than traditional algorithms. Mambou, et al. [8,9] have also done some valuable research in this field.

In fact, most industrial processes are highly non-linear. Artificial neural networks (ANNs) are widely used in soft sensor technology because of their strong nonlinear fitting ability. Feil, et al. [10] presented a soft sensor for melt index by using a semi-mechanistic modeling approach; the ANN-based modeling approach can be efficiently used for on-line state estimation. Ko, et al. [11] used a soft sensor based on ANNs to predict the grain size of ore, and the resulting model can accurately measure the grain size distribution in industrial production in real time. Shakil, et al. [12] developed a soft sensor for measuring gas emissions during industrial boiler combustion using a dynamic ANN, and measurement results with a high accuracy were obtained. Bo, et al. [13] constructed an adaptive soft sensor instrument based on ANN technology and applied it to an advanced control system, the control system realized the quality closed-loop control very well. Jiesheng, et al. [14] used an ANN soft sensor model to predict the concentrate grade. Simulations show that the model has better generalization results and prediction accuracy.

The rapid development of distributed control systems (DCS) provides us with a huge amount of industrial information [15,16]. The big data also presents us with another problem in non-linear soft sensing: there are many redundant input variables in the process. If too many input variables are introduced in the training process of the ANN, the data dimensions increase and more computational power is required. At the same time, introducing variables that are unrelated to the prediction results will increase the noise in the dataset and reduce the prediction accuracy of the ANN. Therefore, the study of effective variable selection algorithms for soft sensors has attracted interest from many researchers. Variable selection algorithms can be broadly divided into two categories: wrapper and filter algorithms. The wrapper algorithms essentially treat the selection of inputs as an optimization of the model structure. They compare and evaluate either all or a subset of the possible input variable sets and select the set that yields optimal performance [17,18]. Wrapper algorithms utilize the performance of machine learning to evaluate the goodness of feature subset and, therefore, have great chance of leading to good prediction performances. Due to this advantage, the wrapper algorithms are widely studied and applied [19–24]. In spite of the performance, wrappers can be very computationally expensive since many predictive models with different feature subsets have to be built. Moreover, the results of the wrapper strategy lack generality as their use is limited to a specific regression model.

The filter algorithms utilize a statistical measure of the degree of dependence between the input and output variables as the criterion for variable selection. Compared to the model-based wrapper algorithm, the filter algorithms are model free and have less computational complexity. Among them, mutual information (MI) is a very suitable measure of dependence for variable selection during ANN development, since it is an arbitrary measure and makes no assumption regarding the structure of the dependence between variables. For a modeling problem, the ultimate objective is to reduce as much as possible an error criterion; the most frequently used is the mean squared error (MSE). Frénay et al. [25] demonstrated that that MI was an adequate criterion for feature selection with respect to the MSE, when the estimation error was identically distributed for any input variables with a uniform, Laplacian or Gaussian distribution. The MI also has been found to be robust due to its insensitivity to noise and data transformations [26,27]. Therefore, many MI-based feature/variable selection algorithms have been proposed. Hanchuan, et al. [28] developed a minimal redundancy maximal relevance (mRMR) criterion to minimize redundancy, and made use of a series of intuitive measures of relevance and redundancy to select meaningful features. Estévez, et al. [29] proposed an enhanced version of mutual information feature selection (MIFS) and mRMR that introduced the normalized mutual information (NMI) as a measure of redundancy. The proposed normalized mutual information feature selection (NMIFS) algorithm showed better performance than MIFS and mRMR, by compensating for the MI bias toward multivalued features and restricting its values to the range [0, 1]. Bennasar, et al. [30] gave a normalized joint mutual information maximisation (NJMIM) for feature selection, which is based on the "maximum of the minimum" criterion. The results showed that the NJMIM method outperforms the other methods on several tested public datasets. Novovicová, et al. [31] designed a sequential forward feature selection method based on conditional mutual information called CMIFS to find a subset of features that are most relevant to the classification task. The feature selection algorithm has better classification accuracy on high dimensional Reuters textual data.

However, the NMIFS also has some disadvantages. One of them is that the execution of the variable selection only depends on the data characteristics and has nothing to do with the structure and model accuracy of the ANN. Another disadvantage is that the algorithm is essentially a greedy search; it only deletes one variable at every iteration. This mechanism will bring about an inherent drawback, in that the solution tends to fall into a local optimum.

The tabu search (TS) algorithm can avoid local optima by extending local neighborhood searches. The research of Zhang, et al. [32] showed that TS is a promising tool for variable selection due to the quality of its obtained variable subset and its computational efficiency. Pachecoaab, J. [33] proposed a TS method that could find a smaller subset from a set of variables, to effectively classify cases. Osman, I. H. [34] used TS for the vehicle routing problem; the new method provided a set of lowest-cost delivery routes and achieved satisfactory results. Lin, et al. [35] applied TS to an economic dispatch problem, and results showed that the proposed algorithm can provide accurate solutions with reasonable performance.

In this paper, an improved NMIFS for soft sensors is proposed. To the best of our knowledge, this is the first implementation developing a neural network-based soft sensor by combining NMIFS with TS. The contributions of this paper are as follows:

(1) The proposed algorithm performs variable selection by combining the NMI and training accuracy of neural networks.
(2) TS is used as an auxiliary optimization method, and it effectively prevents the proposed algorithm from falling into local optimums.
(3) The developed soft sensor is applied to predict the sulfur dioxide (SO_2) emissions of flue gas in a practical power plant, and it has better performance than other algorithms.

The paper is organized as follows. Section 2 reviews the theories of NMIFS and ANN. Section 3 presents the design and analysis of the proposed algorithm. Section 4 gives the numerical simulation results of two artificial dataset examples. Section 5 shows that the soft sensor based on NMI and TS (NMI-TS) achieved an accurate measurement of flue gas SO_2 concentration. Finally, some concluding remarks are given in Section 6.

2. Theoretical Overview

2.1. Input Variable Selection Techniques

In the training processes of ANNs, redundant candidate variables tend to complicate the model, reduce the modeling accuracy, and can even lead to the over-fitting phenomenon. The purpose of the input variable selection (IVS) technique is to correctly select the relative variables from the candidate variable set **C** that contains all the potential input variables of the process. In the variable selection process, variables that contribute less or even have negative effects to the model accuracy will be excluded.

In general, the IVS can be implemented in three ways: (1) sequential forward selection, starting from a null set and selecting a variable into target set at every iteration until there is no improvement in the prediction accuracy of the model; (2) sequential backward selection, starting from set **C** and deleting a variable at a time until the termination rule is met; (3) global optimization, finding the best variable set among all the possible schemes with optimization algorithms.

Mathematically, if there are **m** input variables in **C**, there will be totally ($2^m - 1$) subsets in the selection result set [36]. Therefore, it is very difficult to find the best solution if the number of input variables is large. Based on this consideration, it is necessary to find a statistical metric by which to scale the degree of dependency between the input and output variables. Then, we can preliminarily select the input variable before modeling with ANNs. This statistical pre-processing input variables simplifies the complexity of modeling and can provide resulting solutions that have wider adaptability to different ANN structures. However, the performance of the IVS algorithm is highly dependent on the statistical metrics used.

During the development of IVS algorithms for ANNs, MI has been found to be a valuable evaluation criterion, as it is an arbitrary measure and makes no assumption regarding the structure of the inter-variable dependencies.

2.2. Mutual Information (MI)

Suppose that Y is a random output variable there will be some uncertainty in the observation value $y \in Y$. The uncertainty is defined on the basis of Shannon entropy, H. Suppose X is a random input variable and Y is output variable dependent on X, the mutual observation of (X, Y) will lower uncertainty as the knowledge of X allows inference toward Y. For a continuous variable, the definition of MI can be expressed as follows [34]:

$$I(X;Y) = \iint p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy \qquad (1)$$

where "p(x, y)" is the joint probability density function (PDF), and "p(x)" and "p(y)" are the marginal PDFs of X and Y, respectively. The entropy of X and Y and its relationship to their MI is shown in Figure 1.

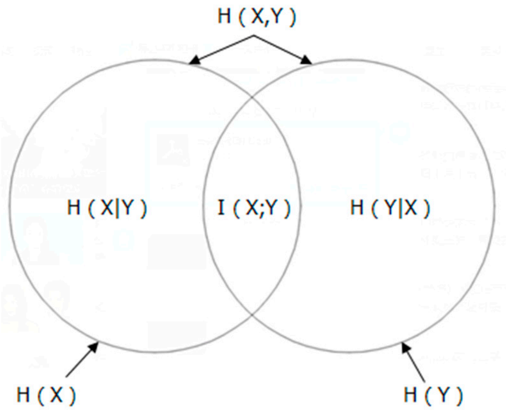

Figure 1. Relationship between entropy and mutual information.

MI has three primary properties:

1. Symmetry: I(X;Y) = I (Y;X).The information extracted from Y about X is of the same amount as the information extracted from X about Y. The only difference between them is the angle of the observer.
2. Positive: I(X;Y) ≥ 0. For all extracting information, the worst case scenario is that there is no information, that is I(X;Y) = 0
3. Extremum: I(X;Y) ≤ H(X), I(Y;X) ≤ H(Y). The amount of information extracted from one event about another is at most equal to the entropy of the other event, and does not exceed the amount of information contained in this other event itself.

The I(X;Y) measures the dependence of X and Y and provides useful information for variable selection. Therefore, how to compute I(X;Y) is very pivotal to MI-based variable selection algorithms. The difficulty of calculating I(X;Y) is that the real functional form of the PDF in (1) is unknown in real world. To solve the difficulty, many approximate estimation algorithms of MI have been widely studied to resolve PDFs. For instance, a kernel density estimation (KDE) exerts a basis function to every point of the feature data and has good computational quality [28]. Then, the PDF approximation can be achieved by adopting an envelope of all the basic functions exerted on each point. K-nearest neighbor (KNN) [37,38] is also an effective method to solve this problem. Despite superior approximation results, these kinds of algorithms have significantly high computational load, especially for large-scale problems. Histogram methods [39] present an alternative solution with acceptable accuracy and conspicuously higher efficiency than KDE methods.

The computational results fluctuate a lot when MI is used to actual production problems. For this reason, it can hardly be taken as an indicator to compare the similarity between different variables. A new method is introduced to normalize MI in which the entropy is applied as the denominator to adjust the value of MI within [0, 1]. The expression is shown as:

$$\mathrm{N}(\mathrm{X},\mathrm{Y}) = 2\frac{\mathrm{I}(\mathrm{X},\mathrm{Y})}{\mathrm{H}(\mathrm{X})+\mathrm{H}(\mathrm{Y})} \tag{2}$$

Following that, NMI can be effectively applied to assess the similarity between the input variables and the target variables.

Inspired by Battiti and Hanchuan [26,28], the evaluation criterion whether a variable is selected is designed as follows:

$$\mathrm{B}_{\mathrm{j}} = \mathrm{I}(\mathrm{Y};\mathrm{pj}) - \xi\frac{1}{\mathrm{k}}\sum_{\mathrm{i}=1}^{\mathrm{k}}\mathrm{I}\left(\mathrm{a}_{\mathrm{i}};\mathrm{p}_{\mathrm{j}}\right) \tag{3}$$

where Y is the output variable, p_{j} is the pending variable, a_{i} is the selected variable, and ξ is a constant that is used to control the ratio of redundant items in B_{j}. The parameter ξ is used as a factor for controlling the redundancy penalization among single features and has a great influence on FS. The values of ξ are different according to the sample size, complexity and number of variables, and are usually determined experimentally based on the specific problem.

2.3. Tabu Search

TS is an intelligent optimization algorithm that extends local neighborhood searches. The most important idea of TS is to mark the local optimal solution that has been reached and try to avoid these objects in the further iterative search, to ensure the exploration of different effective search approaches.

The solution quality of greedy search algorithm depends on the starting point and neighborhood structure, and it is very easy to fall into a local optimum. To obtain better solutions, researchers usually expand or change neighborhood structure, or start searching from multiple starting points. However, these strategies still cannot guarantee the global optimum of the solution due to the nature of their "greed".

Unlike a greedy search, TS has the advantage of being able to escape from a local optimum by accepting inferior solutions. The implementation of TS involves the design of neighborhood, tabu list, tabu length, amnesty rule, termination rule and constitutes a loop with them in the search process.

At the beginning of TS, an initial solution is generated randomly or by some heuristic algorithm. Then, the quality of all solutions in the neighborhood is evaluated according to the fitness function. For a minimization problem, the solution that has smallest objective value has largest fitness function and will be selected as the solution of next iteration. The tabu list is a short-term memory of the specific changes of recent moves within the neighborhood, and can be used to preventing future moves from undoing those changes. Tabu length is the length of the tabu list. The amnesty rule is applied to avoid lost superior solution. If a solution in the tabu list is obviously better than other solutions, it can be taken as the solution of the current iteration ignoring the tabu rules. The termination rule is to stop the search process when the number of iterations reaching the limit or the obtained solution satisfying the preset accuracy.

Define the current solution as x, neighborhood as N(x), fitness function as C(x), tabu list as T. The algorithm flow of TS is described as follows:

$$C(x*) = \max\{C(x), x \in N(x) \backslash T\} \tag{4}$$

where X* is the optimal solution that is not written into the tabu list in the neighborhood. The algorithm flow as follows:

Step 1 Generate an initial solution $x \in X$, set $x^* = x$, $T = \emptyset$, define amnesty rule as $A(x) = C(x^*)$, iterations $k = 0$.
Step 2 $k = k + 1$, if $k > MK$ (MK is the maximum number of iterations), Stop the search.
Step 3 $C(x_L)$ $\max\{C(x),x \in N(x)\}$, if $A(x) < C(x_L)$, set $x = x_L$, jump to Step 5.
Step 4 $C(x_K)$ $\max\{C(x),x \in N(x)\backslash T\}$, set $x = x_K$.
Step 5 If $C(x) < C(x^*)$, set $x = x^*$, $C(x^*) = C(x)$, $A(x) = C(x^*)$.
Step 6 Add x* to tabu list, release the solution that reaches the tabu length and return to Step 2

TS is widely used for combination optimization problems and has superiority in global search [40,41]. In this paper, we adopted the search mechanism of TS and incorporated it with NMIFS to develop a new variable selection method for soft sensors.

3. Design and Analysis of Algorithms

3.1. Proposed Methods for Feature Selection

Equation (4) not only contains MI between p_j and Y, but also takes MI between p_j and selected variables as punishment. This means that there should be a high degree of dependency between p_j and the output variables, and p_j should also have a lower dependency on the variables in **A**. In this way, variables that have less impact on the prediction accuracy, or redundant input variables, can be prevented from being added to **A**. It avoids increasing the size of the model and reduces the data processing time required in training the ANN. At the same time, the model input noise can be reduced to improve the accuracy of the model.

If the input variables are selected as described above, a termination criterion needs to be set for the algorithm. First, for each variable $p_j \in \mathbf{P}$, compute B_j, then find the p_j that maximizes B_j; set $\mathbf{P} \leftarrow \mathbf{P}\backslash\{p_j\}$; set $\mathbf{A} \leftarrow \{p_j\}$, and the algorithm stops when all B_j are less than a scalar quantity. The scalar is a parameter that needs to be input manually, and different values are required for different IVS problems. Because this algorithm is a greedy search algorithm, it will produce a local optimal solution. To ensure the exploration of different effective searches, we used a TS algorithm.

In the process of programming the algorithm, we can set the tabu list as a "tube"; at each iteration, the element that needs to be put into the tabu table is packed into a ""ball", and the number of balls that the tube can accommodate is called tabu length. Then we press the ball into the tube inlet; if a ball emerges from the outlet of the tube, it indicates that the elements have reached the tabu length. If there are no balls appearing from the outlet of the tube, it means that the elements in the tabu list have not reached the tabu length. The size of the neighborhood is set to N and the tabu list to S. The algorithm flow as follows:

Step 1 Evaluate $I(Y; p_j)$; for each variable in P, Select the variable corresponding to the biggest $I(Y; p_j)$ as the starting point.
Step 2 If $P \neq \varnothing$ or $S \neq \varnothing$, Stop the search.
Step 3 The variables satisfying the tabu length were taken from S and added to P.
Step 4 Evaluate B_j for each variable in P.

Step 5 Sort the variables in P by B_j, select the first N variables and put them in neighborhood.
Step 6 Evaluate λ_n (fitness value) for each variable in the neighborhood.
Step 7 Select the variable corresponding to the smallest λ_n as the optimal solution of this iteration
Step 8 Put the other variables in S and return to Step 2.

The pseudo-code of the algorithm is shown in Algorithm 1. The algorithm can be regarded as a combination of local search (LS) algorithm and TS. The LS algorithm is a sequential forward selection approach that consists of two loops. The outer loop performs variable selection by choosing the variables with the highest fitness at every step, where the fitness of a variable is defined by its NMI. Then the algorithm takes the variables selected by the outer loop and enters them into the inner one. The inner loop performs TS by selecting the variable with the lowest fitness at every step, then putting the other variables into the tabu list.

The algorithm proceeds until all variables are selected into **A**, then the algorithm can obtain a "path" according to the order in which variables are selected into **A**. Along this known path, I (I = 1, 2, 3, . . . , i, i is the length of **A**) variables are selected to train the ANN at each iteration, and the cross-validation (CV) MSE is used to evaluate the modelling accuracy and select the optimal solution.

Algorithm 1: Pseudo-code of NMI-TS algorithm

```
Input: dataset {P, Y}
Results: new dataset A after selection
While P ≠ ∅ or S ≠ ∅
    release the solution that reaches the tabu length;
    j = 1;
    While j <= the length of P
        $B_j = I(Y; p_j) - \xi \frac{1}{k} \sum_{i=1}^{k} I(a_i; p_j)$;
        J = j + 1;
    end
    Sort the variables in P by $B_j$ from largest to smallest and select the first N variables, put them in set M;
    N = 1;
    While n <= N
        Evaluate $\lambda_n$ for nth variable in M; (by using Algorithm 2)
    n = n + 1;
    end
    Select the variable corresponding to the smallest $\lambda_n$ and add it to A, Put the other variables in S;
End
```

3.2. Subroutine of Cross-Validation (CV)

CV is a common method used in machine learning to build models and validate model parameters. To achieve the *k*-fold CV, we need to reserve a single sub-dataset (validation set) at the start and use the remaining $k - 1$ sub-datasets (training set) to train a new ANN, then obtain MSE by validating the reserved sub-dataset with the new ANN. The *k*- fold CV process is repeated *k* times with each of the *k* sub-datasets used only once as the validation set, and then produce a single estimation λ_n with the averaged results.

Algorithm 2 presents a detailed subroutine of CV, in which the MSE is used as the evaluation index in selecting variables. The *k*-fold CV procedure is implemented by the loop: (1) reserves a single sub-dataset, (2) uses the remaining $k - 1$ sub-datasets to retrain a new ANN and (3) obtains the MSE by validating the reserved sub-dataset with the retrained ANN.

Algorithm 2: Subroutine of CV

```
Input: dataset {A, Y}
Results: fitness value $\lambda_n$
Divide{A,Y}into k disjoint sub-datasets{$A_1$,$Y_1$},{$A_2$,$Y_2$},{$A_3$,$Y_3$}, … … ,{$A_k$,$Y_k$};
For p = 1:k
    Set A as the input variable, Y as the output variable;
    Train a new ANN with dataset {A,Y}={A,Y}-{$A_k$,$Y_k$};
    Validate the dataset {$A_k$,$Y_k$} on the new ANN and the MSE denoted as $MSE_p$;
End
$\lambda_n = \frac{1}{k}\sum_{p=1}^{k} MSE_p$;
```

4. Numerical Examples

4.1. Experimental Setting

All the algorithms in the paper are tested in the same experiment setting. The programs used for conducting the numerical experiments was coded in MATLAB 2016 and run on a windows 8.1 operating system. They used the same ANN structure that has one hidden layer with the hyperbolic tangent activation function and an output layer with the linear activation function. The ANN are trained using the standard back-propagation algorithm [42]. The number of hidden nodes are determined by trial experiments.

In the experiments of Section 4, 80% of the dataset is used for modeling, and 20% is used to test the accuracy of the model. The simulation results are obtained with the following definitions:

(1) Model size (M.S): The number of input variables selected by the algorithm.

(2) MSE: During the whole modeling process, part of the data is used for modeling, and the other part is used to verify the accuracy of the model. We recorded MSE between the predicted and desired output to evaluate the prediction accuracy of the model.

(3) Coefficient of determination $\left(R^2\right)$: the square of the sample correlation coefficients between the actual and predicted values.

(4) Positive wrong selection (WS+): The ratio of unselected valid variables to all valid variables and the number of unselected valid variables.

(5) Negative wrong selection (WS−):The ratio and number of irrelevant variables in the results.

4.2. Case 1

In the example, a nonlinear dataset containing a pool of candidate variables was generated to validate the performance of the methods. The true non-linear model [24] was:

$$y = \begin{cases} \sqrt{X\cdot\beta - 0.5} + \varepsilon & X\cdot\beta \geq 0.5 \\ e^{X\cdot\beta + 0.5} + \varepsilon & X\cdot\beta \leq 0.5 \end{cases} \tag{5}$$

The input dataset X was generated from a multivariate normal distribution with covariance matrix $\sum$. The covariance between two different variables (columns) was:

$$\sum\nolimits_{i,j} = \rho^{|i-j|}, \forall i \neq j \tag{6}$$

and the coefficient ρ of the covariance matrix was set to 0.2, meaning that multi-collinearity was introduced, ε is white Gaussian noise, where $\beta = [3.0, 1.5, 2.0, 4.0, 0.5, 1.3, -2.6, -3.5, -5.1, 2.0]^T$ which was taken from [43]. Set the dataset $X_1 \in R^{2000\times10}$ was the set of relevant variables, $X_2 \in R^{2000\times10}$ was the set of irrelevant variables, then the input dataset $X = \{X_1,\ X_2\}$.

In order to compare the performance of the proposed algorithm, some state-of-art MI-based variable selection algorithms such as NMIFS, and CMIFS [31] are applied to the numerical dataset.

Moreover, an advanced wrapper algorithm that combines nonnegative garrote with ANN, called NNG-ANN [24], is also applied in the experiments. Table 1 presents the performance of the simulation results with different algorithms. It can be seen from the table that NMI-TS has the best MSE and R^2 with the fewest variable selected. The WS+ is zero, meaning that the algorithm selects all the relative input variables. The WS- is one, meaning that only one irrelative variable is selected in the final model. The results show that the NMI-TS is superior in variable selection accuracy to other algorithms.

Table 1. Simulation results of dataset.

	M.S	MSE	R^2	WS+	WS−
ANN	20	0.1895	0.8747	N/A	N/A
NNG-ANN	13	0.1616	0.8960	10.0%/1	30.7%/4
NMIFS	16	0.1842	0.8781	10.0%/1	43.7%/7
CMIFS	17	0.1617	0.8941	0	41.2%/7
NMI-TS	11	0.1534	0.9003	0	9.1%/1

The curves of MSE to the number of features of this dataset with MI-based algorithms are shown in Figure 2. Note that the characteristic of NNG-ANN is to give the best solution at a time instead of presenting a solution path. Therefore, the curves of NNG-ANN are not presented in this type of figures. It can be seen from the figure that the MSE reaches the best result at 11 features. At those points fewer than 11 features, the performance deteriorates drastically because of the lack of necessary input information. At the right of the best point, MSE slowly deteriorates due to the introduction of redundant information as the number of features increases. Moreover, the other algorithms converge more slowly to reach a better point than NMI-TS.

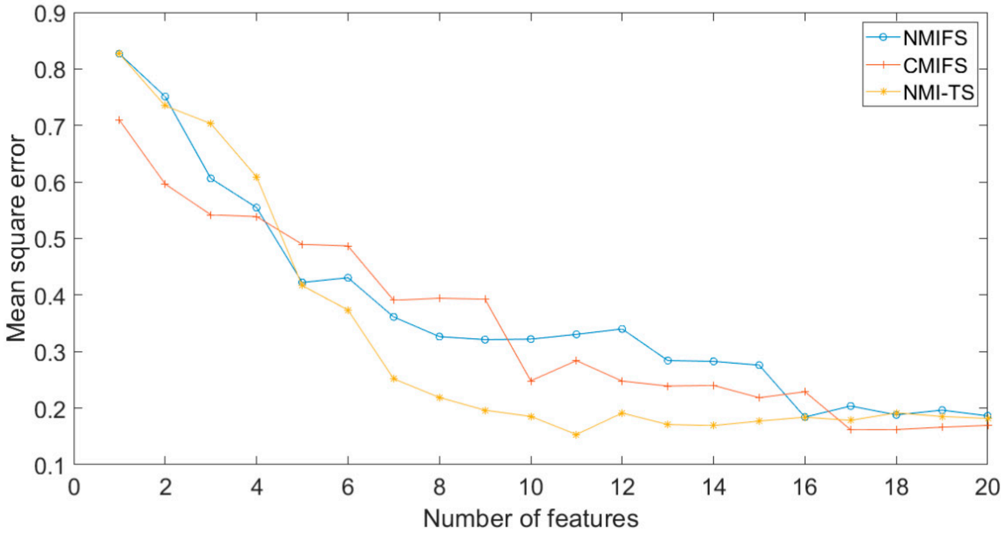

Figure 2. The curves of mean squared error (MSE) number of features of Case 1 dataset with mutual information (MI)-based algorithms.

To further verify the performance of the algorithm, we conducted the following experiments on the above dataset

(1) Sample size

We scale the sample size down to 200–800 and then conducted four experiments using MSE as the evaluation criterion. The results are shown in Figure 3.

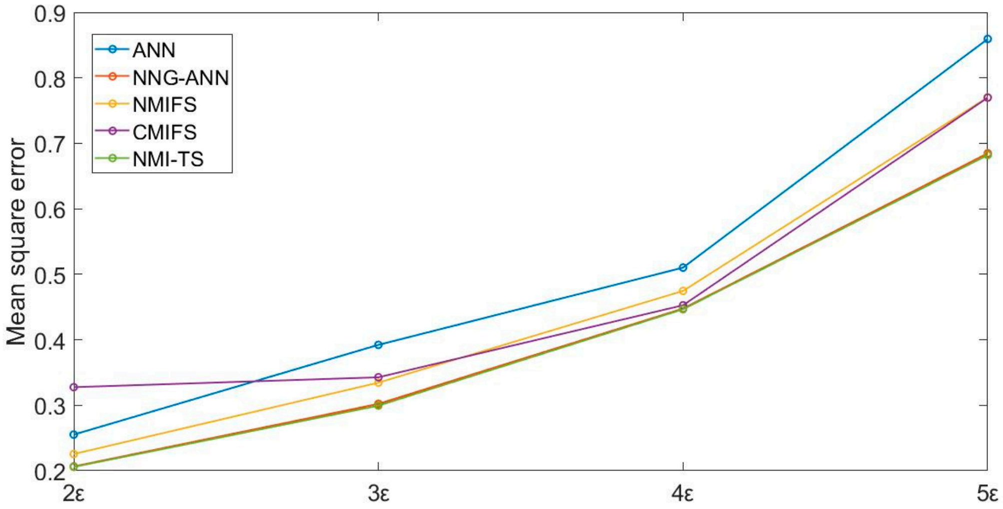

Figure 3. The simulation experiment with different sample size.

(2) Correlation

The correlation between the input variables ρ increased from 0.2 to 0.8, and we then conducted experiments with MSE as the evaluation criterion. Results are shown in Figure 4.

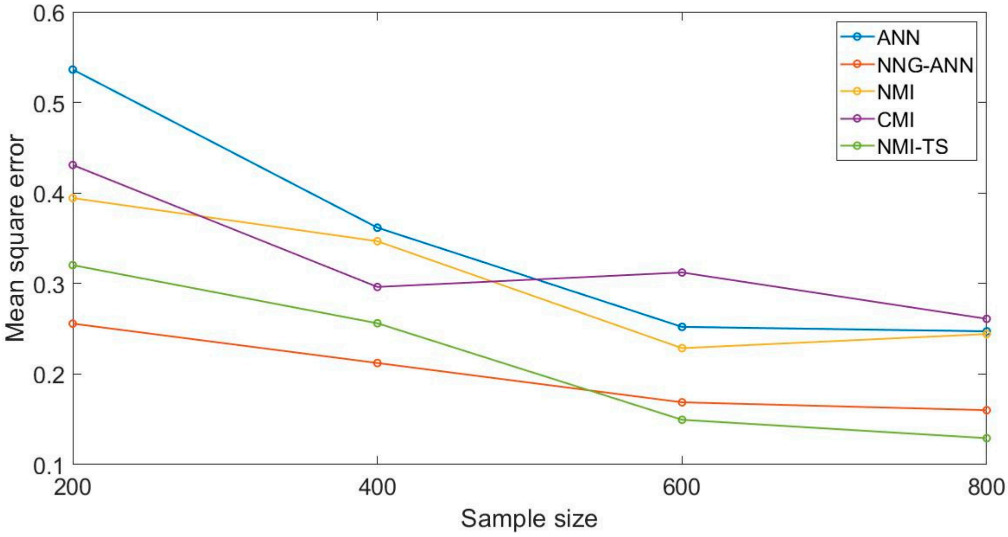

Figure 4. The simulation experiment with different correlation.

(3) Noise intensity

ε in Equation (5) was increased and we designed the experiments with MSE as the evaluation criterion. The results are shown in Figure 5.

Sample size, correlation, and noise intensity all affect the prediction accuracy of the algorithm, especially the modeling accuracy of the ANN. Because MI is used as the evaluation criterion, CMIFS NMIFS and NMI-TS are less affected. Although the MSE of CMIFS NMIFS and NMI-TS are essentially the same in some experiments, NMI-TS usually presents a better model accuracy with fewer variables selected. This algorithm can solve the problem of strong coupling between input variables. It is worth noting that the accuracy of the proposed algorithm has certain dependence the amount of training samples. If the training sample is too small, the sparse distribution of experimental data will reduce the reliability of histogram method. Then, the PDFs between variables cannot be effectively calculated, which will lead to the deterioration of the algorithm performance. Although the utilization of KDE or KNN can improve this situation, the cost of operation is also greatly increased.

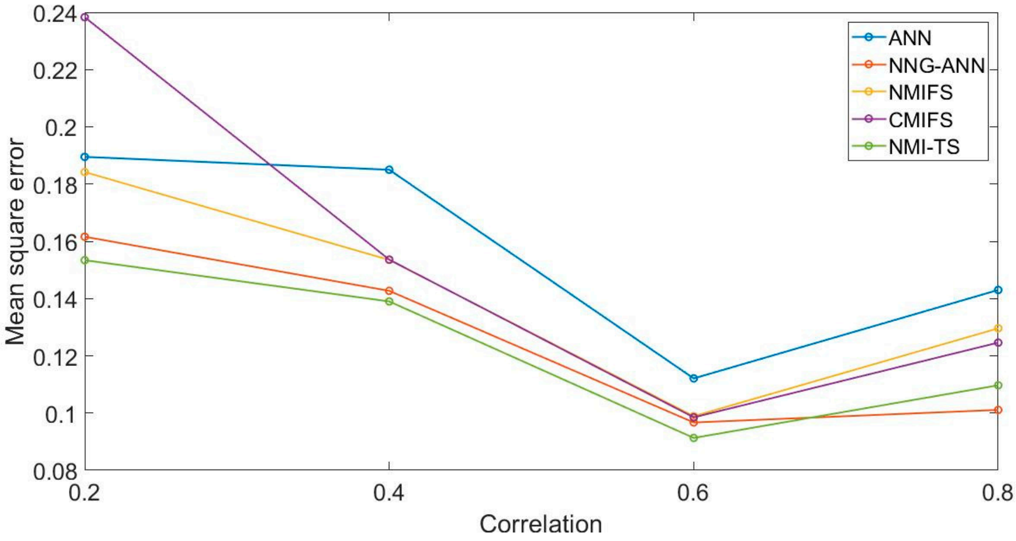

Figure 5. The simulation experiment with different noise.

4.3. Case 2

In order to further validate the proposed algorithm, a benchmark dataset that was first proposed by Friedman, J. H. [44] is applied in our paper. The dataset used in the paper has five input variables relevant to the output variable, which is calculated by

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon \tag{7}$$

where $\{x_1, x_2, x_3, x_4, x_5\}$ are relevant input variables and ε is white Gaussian noise. In addition, this dataset also contains 45 input variables that are irrelevant to the output variable. The role of the variable selection algorithms is to select $\{x_1, x_2, x_3, x_4, x_5\}$ from the total of 50 variables. However, the irrelevant variables have different degrees of correlation to $\{x_1, x_2, x_3, x_4, x_5\}$ ranging from 0.3 to 0.9, which increases the difficulty of variable selection. Table 2 presents the performance of the simulation results on the dataset with different algorithms. As we can see, the NMI-TS has smaller MS and better MSE than other approaches. The curves of MSE to the number of features of this dataset with MI-based algorithms are shown in Figure 6. The figure shows that the NMI-TS converges fast to the best solution, while the other methods require more variables to converge.

Table 2. Simulation results of Friedman dataset.

	M.S	**MSE**	**R^2**	**WS+**	**WS-**
ANN	50	3.8992	0.9216	N/A	N/A
NNG-ANN	20	2.0691	0.9598	20.0%/1	80.0%/16
NMIFS	6	1.5943	0.9687	100.0%/5	100.0%/5
CMIFS	7	1.4257	0.9723	60.0%/3	71.4%/5
NMI-TS	6	1.2076	0.9764	20.0%/1	33.3%/2

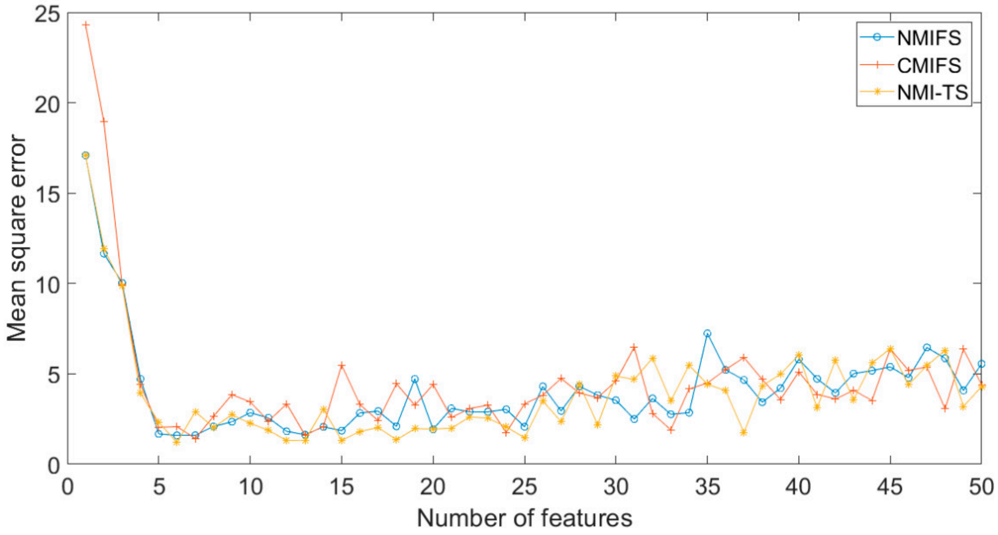

Figure 6. The curves of MSE number of features of Case 2 dataset with MI-based algorithms.

5. Application to Actual Desulfurization Process

5.1. Desulfurization Technology of the Power Plant

The power plant providing the data used in this article operates a limestone-gypsum wet flue-gas desulfurization technology, which uses lime or limestone to react with SO_2 and absorb it. The chemical reaction principles are:

$$SO_2{+}CaO{+}\frac{1}{2}H_2O \rightarrow CaSO_3{\cdot}\frac{1}{2}H_2O \quad (8)$$

$$SO_2{+}CaCO_3 + \frac{1}{2}H_2O \rightarrow CaSO_3{\cdot}\frac{1}{2}H_2O + CO_2 \quad (9)$$

The flue gas containing SO_2 moves up from the bottom of the absorption tower, it encounters the spray layer of alkaline spray and reacts with it to form liquid drops. Tiny liquid drops fall into the slurry pool, where the oxidation of $CaSO_3$ and the crystallization of $CaSO_3{\cdot}1/2H_2O$ are completed; finally, gypsum is formed at the bottom of the absorption tower.

The desulfurization process and related parameters are mainly taken from unit 9 of the power plant. The main feature of this unit is that it adopts the structure of a double absorption tower. Compared with the design of the single absorption tower, this structure can be used for secondary treatment of the discharged flue gas, and can remove SO_2 in the flue gas more effectively. Figure 7 shows the flow direction of flue gas in the flue-gas system.

In the entire desulfurization process, the untreated flue gas is first be introduced into the two-stage absorption tower, and once desulfurized it is introduced into the chimney through a high-pressure dust catcher.

A fan pressurizes the flue gas and sends it to the bottom of the No.9-1 absorption tower, but before this happens it needs to be cooled down. The reason for this is that, within a certain range, lower temperatures are more conducive to the absorption of SO_2 in the flue gas. The flue gas rises from the bottom of the first absorption tower and reaches the top after being treated, and then it is sent to the bottom of the secondary absorption tower for the second treatment. With wet flue-gas desulfurization technology it is easy to produce "fog" with particle sizes of 10–60 microns in the absorption tower. Because the "fog" produced by this process contains SO_2, the gas inside the tower is defogged through a mist eliminator before leaving the absorber. To prevent the liquid in the slurry tank of the absorber from pouring back into the flue-gas inlet, it is necessary to set an overflow port at the upper part of the slurry pool of the first stage absorber. The overflow port of the secondary absorber will overflow the slurry to the first stage absorber when a high liquid level is reached. Due to the large volume of the slurry pool, if the liquid within it remains stationary for a long time, solid precipitation will be generated. To prevent the depositing of solids in the slurry pool, pulsed suspension pumps are used to stir the slurry. Given that oxidation and crystallization processes are required in the slurry tank, an oxidation air blower is used to feed the air to the tower through an aeration device arranged in the slurry tank. Because there is less gypsum produced by the slurry pool in the secondary absorber, slurry is directly fed into the slurry pool of the first stage absorber by the gypsum swirl pump, then the oxidation and crystallization process is carried out. Apart from the different assembly of overflow ports and oxidation air blowers, the structure of the two absorption towers is largely the same, as shown in Figure 8.

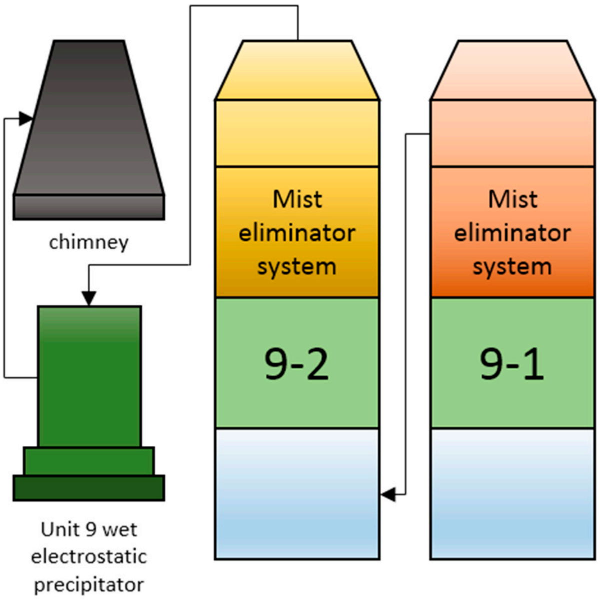

Figure 7. The flow direction of flue gas in the flue-gas system.

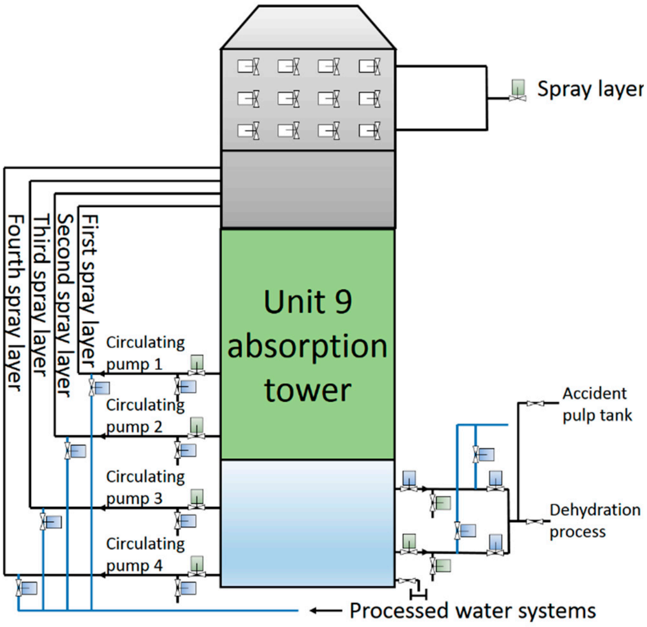

Figure 8. The structure of absorption tower.

5.2. Expriment's Design and Results

In the desulfurization process, the untreated flue gas is first introduced into the two-stage absorption tower, then, after desulfurization, it is introduced into the chimney through the high pressure dust catcher. The flue-gas desulfurization system consists of SO_2 absorption system, flue-gas system, absorbent preparation system, gypsum treatment system, processed water system and accident slurry system. The whole system is very complex and involves many candidate variables that we do not know whether they are related to the output variable or not. There all 52 input variables provided by factory's DCS. Table A1 presents the name, description and unit of these variables. In fact, there are too many variables in the candidate variables pool that may cause modeling inaccuracies. Meanwhile, it is very difficult to precisely select variables and model for such a complex process simply through domain knowledge. Therefore, the variable selection algorithm based on statistical theory is very meaningful to the accurate modeling of the process.

The dataset from the power plant is entered into the three algorithms used in Section 4, 50% of the dataset is used for modeling, and 50% of the dataset is used to test the accuracy of the model. Table 3 shows the modeling accuracy.

Table 3. Simulation results of power plant dataset.

	M.S	MSE	R^2
ANN	52	23.4420	0.5124
NNG-ANN	10	8.8661	0.8243
NMIFS	8	10.2379	0.7710
CMIFS	9	13.6934	0.7225
NMI-TS	4	5.1309	0.8272

The soft sensor presented in this paper achieved the highest accuracy in predicting the SO_2 concentration. Some data was input into the soft sensor based on NMI-TS, and the predicted value was compared with the real value, as shown in Figure 9. Figure 10 presents the curves of MSE to the number of features of desulfurization process with MI-based algorithms. The best solution appears at the point of four input variables. At the left of the point, the MSE deteriorates drastically as the number of variables decreases. Similar to Figures 2 and 6, the NMI-TS has the best convergence rate among these algorithms. But unlike Figures 2 and 6, the MSE tends to increase with fluctuating changes at the right of the point. The reason for this phenomenon is that the practical dataset has large cross-correlation between different input variables as well as uncertainty between redundant and useful information.

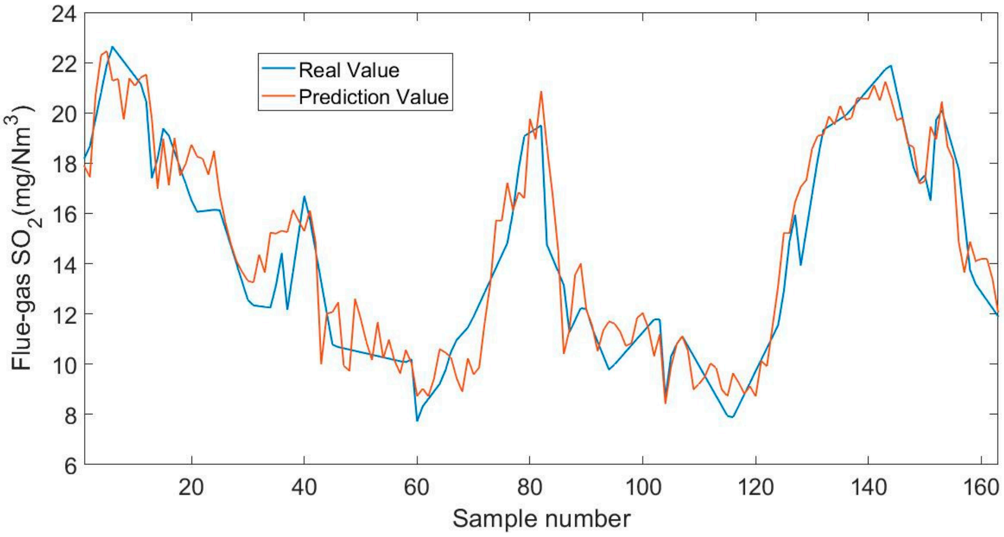

Figure 9. Measured and the predicted flue gas SO_2 concentration.

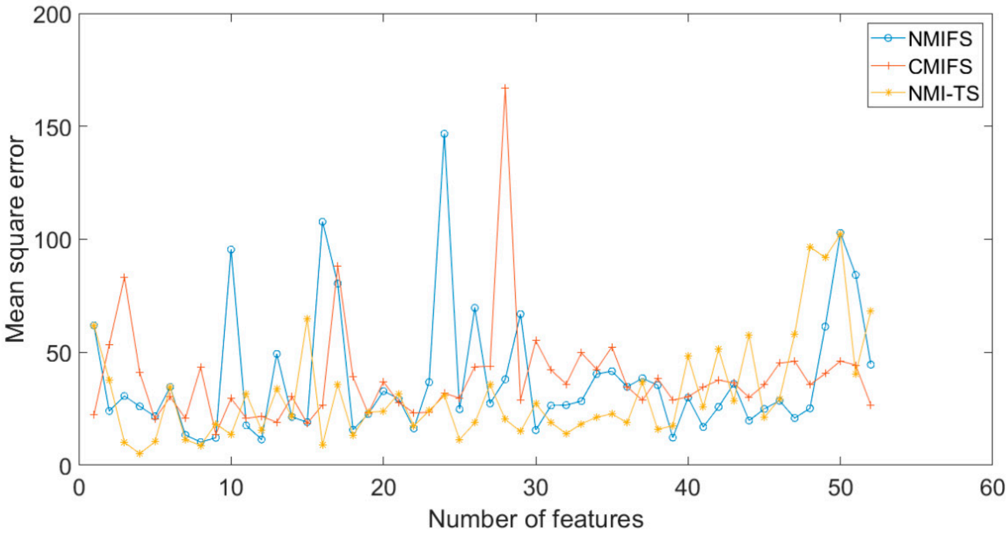

Figure 10. The curves of MSE number of features of the practical example with MI-based algorithms.

According to statistical results, there are four input variables selected by NMI-TS in the optimal solution, shown as Table 4.

Table 4. Variables selected by normalized mutual information-tabu search (NMI-TS) in the optimal solution.

Number	Input Variable	Units
3	#9-3 Circulating SP current	A
33	#9-1AT outlet's flue gas SO_2	mg/Nm^3
38	Limestone slurry to #9 AT flow	Nm^3/h
50	#9-2AT's pH value 1	

The first variable is variable 3 that is the current of the third circulating slurry pump. There are four pumps in the tower 9-2, and every pump has its own motor to adjust the spraying speed of the slurry to control the rate of absorption. Usually, the operator takes the default operation mode under which there are two pumps normally open in the tower 9-2, and manually adjusts the switch and frequency of the third motors according to the production situation. Therefore, variable 3 is highly related to the target variable and included in the optimal solution. Secondly, variable 33 is the outlet's flue gas SO_2 concentration of the first tower and the inlet's flue gas SO_2 concentration of the second tower, which is obviously correlated to the final SO_2 concentration. The third one is variable 38 that is the limestone slurry to the absorption towers. It can be seen from reaction Formulaes (8) and (9) that the degree of this reaction is closely related to the amount of CaO and $CaCO_3$, i.e., the limestone slurry. The fourth one is variable 50 that is the pH value of the slurry in the tower 9-2. This result is quite consistent with basic chemical reaction mechanism and operational experience in the field. If the SO_2 concentration of flue gas is relatively high, the SO_2 absorbed by the slurry will be relatively more and therefore lower the pH value of the slurry. Conversely, a relatively lower SO_2 concentration results in a higher pH value of the slurry.

6. Conclusions

The proposed algorithm for soft sensors is an innovative combination of NMI and TS. NMI-TS makes use of model accurateness to guide the process of variable selection by the NMI algorithm, and avoids the local optimal situation by using TS. We use simulation examples with different scales, correlations, and noise intensities to verify the superiority of our algorithm. Furthermore, the NMI-TS soft sensor was applied to a power plant desulfurization process. The simulation results show that the

NMI-TS soft sensor can achieve high prediction accuracy with a simpler model of SO_2 concentration. The proposed soft sensor is very easily implemented and adaptable since the program can be stored as a subroutine in the host computer of DCS. With the huge amount of production data newly provided by the DCS, the soft sensor model could be easily retrained and updated by calling the subroutine at regular intervals.

In future research, it would be of interest to develop distributed parallel framework [15,16] of the proposed algorithm to solve the problem of increased computational complexity created by big data. The idea of distributed parallel framework could be combined with the proposed algorithm to obtain better and faster feature selection methods that may be successfully applied to large databases.

Author Contributions: Conceptualization, K.S.; Funding acquisition, K.S. and G.Y.; Investigation F.M.; Methodology, P.T.; Supervision G.Y.; Validation, H.Q.; Writing—original draft, P.T.; Writing—review & editing, K.S.

Funding: The work was partially supported by the Shandong Provincial Natural Science Foundation of China (Grant No. ZR2017MF064),the Major Science and Technology Innovation Projects of Shandong Province (Grant No. 2019JZZY010731), the Key Research and Development Program of Shandong Province (Grant No. 2019GGX104037), the China national R&D Key Research Program (2017YFA60700602), and the Ministry of Industry and Information Technology of China under Project “Industrial Internet Platform Test Bed for Optimizing the Operation of Motor and Driving Equipment”.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. All input variables in distributed control systems (DCS).

Number	Input Variable	Units
1	#9-1 Circulating SP current	A
2	#9-2 Circulating SP current	A
3	#9-3 Circulating SP current	A
4	#9-4 Circulating SP current	A
5	#9Cycle SP E B phase current	A
6	#9Cycle SP G B phase current	A
7	AT gypsum DP 2’s current	A
8	AT gypsum DP 1’s current	A
9	Gypsum slurry DP A’s current	A
10	Water ring vacuum pump’s current	A
11	#9AT’s inlet flue-gas temperature	K
12	#9AT’s outlet flue-gas temp.1	K
13	#9AT’s outlet flue-gas temp. 2	K
14	#9AT’s outlet flue-gas temp.3	K
15	#9-2 AT outlet temperature 1	K
16	#9-2 AT outlet temperature 2	K
17	#9 AT’s gypsum slurry density	kg/m^3
18	Limestone slurry density	kg/m^3
19	#9 Furnace chimney inlet net flue-gas pressure	kPa
20	#9-2 AT outlet pressure	kPa
21	AT gypsum DP 1’s inlet pressure	kPa
22	AT gypsum DP 2’s inlet pressure	kPa
23	#9 Gypsum slurry cyclone’s inlet pressure	kPa
24	Gypsum SP to #9 AT tube’s pressure	kPa
25	Generator Power	kW
26	#9 Absorber liquid level’s value	m
27	#9-2 AT liquid level 1	m
28	#9-2 AT liquid level 2	m
29	#9-2 AT liquid level 3	m
30	Gypsum slurry tank’s level	m

Table A1. *Cont.*

Number	Input Variable	Units
31	Belt dewatering machine A's gypsum filter cake thickness	m
32	#9Furnace's original flue-gas SO_2	mg/Nm^3
33	#9-1AT outlet's flue-gas SO_2	mg/Nm^3
34	#9 Furnace original concentration of flue-gas and dust	mg/Nm^3
35	#9 Furnace original flue-gas NOx	mg/Nm^3
36	#9 Furnace original flue-gas O_2	mg/Nm^3
37	#9-1 AT outlet flue-gas O_2	mg/Nm^3
38	Limestone slurry to #9 AT flow	Nm^3/h
39	#9-2AT's feed flow	Nm^3/h
40	Total air volume	Nm^3/h
41	Total coal	Nm^3/h
42	#9Furnace's chimney inlet net flue-gas flow	Nm^3/h
43	#9 original smoke flow	Nm^3/h
44	#9 AT gypsum's slurry flow	Nm^3/h
45	AT gypsum DP 1's speed feedback	r/s
46	AT gypsum DP 2's speed feedback	r/s
47	Gypsum SP A's speed feedback	r/s
48	Vacuum belt dewatering machine's speed feedback	r/s
49	#9 AT gypsum's slurry pH value	
50	#9-2AT's pH value 1	
51	Unit 9 desulfurization efficiency	
52	Intermediate value of desulfurization efficiency of Unit 9	

AT = Absorption tower; DP = discharge pump; SP = slurry pump.

References

1. Paulsson, D.; Gustavsson, R.; Mandenius, C.F. A soft sensor for bioprocess control based on sequential filtering of metabolic heat signals. *Sensors* **2014**, *14*, 17864–17882. [CrossRef] [PubMed]
2. Chen, K.; Liang, Y.; Gao, Z.; Liu, Y. Just-in-time correntropy soft sensor with noisy data for industrial silicon content prediction. *Sensors* **2017**, *17*, 1830. [CrossRef] [PubMed]
3. Kaneko, H.; Arakawa, M.; Funatsu, K. Development of a new soft sensor method using independent component analysis and partial least squares. *Aiche J.* **2010**, *55*, 87–98. [CrossRef]
4. Ahmed, F.; Nazir, S.; Yeo, Y.K. A recursive PLS-based soft sensor for prediction of the melt index during grade change operations in HDPE plant. *Korean J. Chem. Eng.* **2009**, *26*, 14–20. [CrossRef]
5. Wang, D.; Liu, J.; Srinivasan, R. Data-driven soft sensor approach for quality prediction in a refining process. *IEEE Trans. Ind. Inform.* **2010**, *6*, 11–17. [CrossRef]
6. Xing, H.; Jun, J.; Kaixin, L.; Zengliang, G.; Yi, L. Soft sensing of silicon content via bagging local semi-supervised models. *Sensors* **2019**, *19*, 3814.
7. Yuan, X.; Huang, B.; Wang, Y.; Yang, C.; Gui, W. Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3235–3243. [CrossRef]
8. Mambou, S.; Krejcar, O.; Kuca, K.; Selamat, A. Novel cross-view human action model recognition based on the powerful view-invariant features technique. *Future Internet* **2018**, *10*, 89. [CrossRef]
9. Mambou, S.; Krejcar, O.; Maresova, P.; Selamat, A.; Kuca, K. Novel hand gesture alert system. *Appl. Sci.* **2019**, *9*, 3419. [CrossRef]
10. Feil, B.; Abonyi, J.; Pach, P.; Nemeth, S.; Arva, P.; Nemeth, M.; Nagy, G. Semi-mechanistic models for state-estimation–soft sensor for polymer melt index prediction. In *Artificial Intelligence and Soft Computing*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3070, pp. 1111–1117.
11. Ko, Y.D.; Shang, H. A neural network-based soft sensor for particle size distribution using image analysis. *Powder Technol.* **2011**, *212*, 359–366. [CrossRef]
12. Shakil, M.; Elshafei, M.; Habib, M.A.; Maleki, F.A. Soft sensor for NO_x and O_2 using dynamic neural networks. *Comput. Electr. Eng.* **2009**, *35*, 578–586. [CrossRef]

13. Bo, C.M.; Li, J.; Sun, C.Y.; Wang, Y.R. The application of neural network soft sensor technology to an advanced control system of distillation operation. In Proceedings of the International Joint Conference on Neural Networks, Portland, OR, USA, 20–24 July 2003; pp. 1054–1058.
14. Jiesheng, W.; Shuang, H.; Nana, S. Features extraction of flotation froth images and BP neural network soft-sensor model of concentrate grade optimized by shuffled cuckoo searching algorithm. *Sci. World J.* **2014**, *2014*, 17.
15. Le, Y.; Zhiqiang, G. Big data quality prediction in the process industry: A distributed parallel modeling framework. *J. Process. Control* **2018**, *68*, 1–13.
16. Xinyu, Z.; Zhiqiang, G. Local parameter optimization of LSSVM for industrial soft sensing with big data and cloud implementation. *IEEE Trans. Ind. Inform.* **2019**, 1. [CrossRef]
17. Greenland, S. Modeling and variable selection in epidemiologic analysis. *Am. J. Public Health* **1989**, *79*, 340–349. [CrossRef] [PubMed]
18. Hui, Z.; Hastie, T. Addendum: Regularization and variable selection via the elastic net. *J. R. Stat. Soc.* **2005**, *67*, 768.
19. Raftery, A.E.; Dean, N. Variable selection for model-based clustering. *J. Am. Stat. Assoc.* **2017**, *101*, 168–178. [CrossRef]
20. Wang, J.L.; Zhang, J.; Wang, X.X. A data driven cycle time prediction with feature selection in a semiconductor wafer fabrication system. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 173–182. [CrossRef]
21. Mcnitt-Gray, M.F.; Huang, H.K.; Sayre, J.W. Feature selection in the pattern classification problem of digital chest radiograph segmentation. *IEEE Trans. Med. Imaging* **1995**, *14*, 537. [CrossRef]
22. Le, Y.; Zhiqiang, G. Variable selection for nonlinear soft sensor development with enhanced binary differential evolution algorithm. *Control Eng. Pract.* **2018**, *72*, 68–82.
23. Zhiqiang, G.; Zhihuan, S. A comparative study of just-in-time-learning based methods for online soft sensor modeling. *Chemom. Intell. Lab. Syst.* **2010**, *104*, 306–317.
24. Sun, K.; Liu, J.; Kang, J.L.; Jang, S.S.; Wong, S.H.; Chen, D.S. Development of a variable selection method for soft sensor using artificial neural network and nonnegative garrote. *J. Process Control* **2014**, *24*, 1068–1075. [CrossRef]
25. Frénay, B.; Doquire, G.; Verleysen, M. Is mutual information adequate for feature selection in regression? *Neural Netw.* **2013**, *48*, 1–7. [CrossRef] [PubMed]
26. Battiti, R. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. Neural Netw.* **1994**, *5*, 537–550. [CrossRef] [PubMed]
27. Soofi, E.S.; Retzer, J.J. Information importance of explanatory variables. In *IEE Conference in Honor of Arnold Zellner: Recent Developments in the Theory, Method and Application of Entropy Econometrics*; IEE: Washington, DC, USA, 2003.
28. Hanchuan, P.; Fuhui, L.; Chris, D. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
29. Estévez, P.A.; Michel, T.; Perez, C.A.; Zurada, J.M. Normalized mutual information feature selection. *IEEE Trans. Neural Netw.* **2009**, *20*, 189–201. [CrossRef] [PubMed]
30. Bennasar, M.; Hicks, Y.; Setchi, R. Feature selection using joint mutual information maximisation. *Expert Syst. Appl.* **2015**, *42*, 8520–8532. [CrossRef]
31. Novovicová, J.; Somol, P.; Haindl, M.; Pudil, P. Conditional mutual information based feature selection for classification task. In *Iberoamerican Congress on Pattern Recognition*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4756, pp. 417–426.
32. Zhang, H.; Sun, G. Feature selection using tabu search method. *Pattern Recognit.* **2002**, *35*, 701–711. [CrossRef]
33. Pachecoaab, J. A variable selection method based on tabu search for logistic regression models. *Eur. J. Oper. Res.* **2009**, *199*, 506–511. [CrossRef]
34. Osman, I.H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann. Oper. Res.* **1993**, *41*, 421–451. [CrossRef]
35. Lin, W.M.; Cheng, F.S.; Tsay, M.T. An improved tabu search for economic dispatch with multiple minima. *IEEE Power Eng. Rev.* **2007**, *22*, 70. [CrossRef]
36. Enrique, R.; Josep María, S. Performing feature selection with multilayer perceptrons. *IEEE Trans. Neural Netw.* **2008**, *19*, 431–441.

37. Peterson, L.E. K-nearest neighbor. *Scholarpedia* **2009**, *4*, 1883. [CrossRef]
38. Kraskov, A.; Stögbauer, H.; Grassberger, P. Estimating mutual information. *Phys. Rev. E Stat. Nonlinear Soft Matter Phys.* **2004**, *69*, 066138. [CrossRef] [PubMed]
39. Fukunaga, K. *Introduction to Statistical Pattern Recognition*; Elsevier: Amsterdam, The Netherlands, 2013.
40. Khaksar, W.; Hong, T.S.; Khaksar, M.; Motlagh, O. A fuzzy-tabu real time controller for sampling-based motion planning in unknown environment. *Appl. Intell.* **2014**, *41*, 870–886. [CrossRef]
41. Hongxing, W.; Wang, B.; Wang, Y.; Shao, Z.; Chan, K.C.C. Staying-alive path planning with energy optimization for mobile robots. *Expert Syst. Appl.* **2012**, *39*, 3559–3571.
42. Faris, H.; Aljarah, I.; Mirjalili, S. Improved monarch butterfly optimization for unconstrained global search and neural network training. *Appl. Intell.* **2017**, *48*, 445–464. [CrossRef]
43. Pan, C.C.; Bai, J.; Yang, G.K.; Wong, S.H.; Jang, S.S. An inferential modeling method using enumerative PLS based nonnegative garrote regression. *J. Process Control* **2012**, *22*, 1637–1646. [CrossRef]
44. Friedman, J.H. Multivariate adaptive regression splines. *Ann. Stat.* **1991**, *19*, 1–67. [CrossRef]