processes

MDPI

Article

Fault Detection and Identification of Blast Furnace Ironmaking Process Using the Gated Recurrent Unit Network

Hang Ouyang [1], Jiusun Zeng [1,*] and Yifan Li [1] and Shihua Luo [2]

[1] College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China; sutouyangh@sina.com (H.O.); liyifan@cjlu.edu.cn (Y.L.)

[2] School of Statistics, Jiangxi University of Finance and Economics, Nanchang 330013, China; luoshihua@aliyun.com

* Correspondence: jszeng@cjlu.edu.cn; Tel.: +86-139-6800-6265

Received: 22 February 2020; Accepted: 23 March 2020; Published: 27 March 2020

Abstract: It is of critical importance to keep a steady operation in the blast furnace to facilitate the production of high quality hot metal. In order to monitor the state of blast furnace, this article proposes a fault detection and identification method based on the multidimensional Gated Recurrent Unit (GRU) network, which is a kind of recurrent neural network and is highly effective in handling process dynamics. Comparing to conventional recurrent neural networks, GRU has a simpler structure and involves fewer parameters. In fault detection, a moving window approach is applied and a GRU model is constructed for each process variable to generate a series of residuals, which is further monitored using the support vector data description (SVDD) method. Once a fault is detected, fault identification is performed using the contribution analysis. Application to a real blast furnace fault shows that the proposed method is effective.

Keywords: gated recurrent unit; support vector data description; time sequence prediction; fault detection and identification

1. Introduction

Maintaining the blast furnace system at a stable status is critical to ensure efficient production of high-quality blast furnace hot metal [1]. Therefore, condition monitoring of the blast furnace ironmaking process becomes a significant issue. During the operation of blast furnace ironmaking process, different kinds of faults may happen, such as hanging, low stockline and abnormal gas flow. If the faults cannot be detected and identified in time and accurately, it may lead to loss in production rate or even a significant accident.

The problem of fault detection and diagnosis for blast furnace ironmaking process is a long lasting and well research topic. Traditional methods like expert knowledge and fuzzy logic have been well developed in different kinds of expert systems [2]. However, constructing and maintaining an up-to-date knowledge base is difficult. Alternatively, classification-based algorithms like support vector machine have been applied to diagnose faults in blast furnaces [3]. Liu et al. [4] proposed a novel strategy based on cost-conscious least squares support vector machine (LS-SVM) to achieve rapid diagnosis of blast furnace faults. An et al. [5] proposed a support vector machine for multiple classification to diagnose blast furnace faults. The main assumption of classification-based methods is that sufficient faulty samples can be collected, which is often not true in a real blast furnace. More recently, multivariate statistical methods became popular in the monitoring of blast furnaces. For example, Vanhatalo applied the principal component analysis (PCA) to monitor the status of an experimental blast furnace [6]. A two-stage PCA is considered to deal with multi-modal distribution

in blast furnace data [7]. Shang et al. [8] developed a recursive transformed component statistical analysis (RTCSA)-based algorithms to monitor incipiently happened faults in the iron-making process. In addition, other kinds of PCA-based approaches have been introduced to monitor process faults, such as robust PCA [9] and convex hull-based PCA [10].

In order to deal with process dynamics, Zeng et al. applied a state space model to extract residuals from the process data and used the support vector data description (SVDD) to detect blast furnace faults [11]. Also, Vanhatalo and Kulahci [12] considered the impact of autocorrelation to statistical methods like PCA. Dynamic principal component analysis (DPCA) [13,14] and dynamic linear discriminant analysis (DLDA) [15] are also used to handle dynamic processes. From the above analysis, it can be seen that how to handle process dynamics has become an important task in fault detection and diagnosis of blast furnace.

In this paper, a new process monitoring method based on the GRU network [16] is considered to detect and identify process faults in blast furnace. The GRU network is a new type of recurrent neural network (RNN). Comparing to conventional RNN methods like long-short term memory network, it has comparable capability to handle process dynamics, however with a simpler structure and fewer parameters. In fault detection, a GRU neural network is used to make prediction for each process variable, so that the process dynamics can be filtered and a series of residuals can be generated. The generated residuals are then monitored using the support vector data description (SVDD) method [11]. Faulty variables are then identified by inspecting the deviation of the residuals from normal operation condition (NOC). The benefits of the proposed method can be summarized as: (i) the introduction of GRU network can fully capture the dynamic characteristics of the blast furnace data; (ii) faulty variables can be identified by investigating the residual of each variable, which greatly simplifies subsequent fault diagnosis task.

2. Methodologies

This section describes the methodologies applied in fault detection and identification of blast furnace system. Section 2.1 briefly introduces the GRU network, which is an extension of the LSTM network. Section 2.2 describes the SVDD classifer.

2.1. GRU Neural Network

GRU is a type of recurrent neural network. The main difference between RNN and feed-forward artificial neural network is in their structure. In a feed-forward artificial neural network, signals travel from the inputs to outputs and the flow of information is in the forward direction only. Since there is no backward/feedback flow, the name of "feed-forward" is justified. In contrast, an RNN allows feedback from output to input and hence it is called "recurrent". In addition, the output of the previous time step/state in RNN will be used as the input of the next time step, which is different from feed-forward neural network that considers fixed length input and fixed length output only. With this kind of recurrent structure, RNN can be used to learn the characteristics of the time series and make predictions. A widely used RNN is the LSTM network, which is very suitable to capture long-term dependencies and also able to avoid the vanishing gradient problem. As an improvement of LSTM, GRU network inherits its advantages, whilst having an optimized structure and fewer parameters, resulting in lower computation load and better generalization ability.

2.1.1. The Structure of LSTM Cell

LSTM [17] was originally proposed in 1997, in order to solve the vanishing gradient problem faced by RNN [18]. The main difference between LSTM and standard RNN network is the handling of long-term dependencies. In the RNN network, each cycle involves only the last state and the current input. Because each prediction only involves the state at the last moment, the RNN can only establish a dependency relationship between states in a short time. In contrast, LSTM can establish dependencies between states at arbitrary long intervals, so they are called "Long-Short Term

Memory network". In addition, the LSTM has a cell state update process similar to the conveyor belt structure. The old cell state will remain on the conveyor belt until it needs to be forgotten by structure called "gate". Through this conveyor belt structure, LSTM can take long-term memory from the conveyor belt at any time for learning the characteristics of time series and make predictions. An LSTM unit consists of a cell, an input gate, a forget gate and an output gate. The cell is used to record state values at different time intervals and the three gates are used to control the flow of the information. The introduction of three gates enables LSTM to keep, utilize, or discard a state when necessary.

Let $\mathbf{x}_t$ denote a data sample at the tth time instance, $\mathbf{C}_{t-1}$ denote the cell value and $\mathbf{h}_{t-1}$ the hidden state of each cell at the $t-1$th time instance. The information of previous time is stored in $\mathbf{C}_{t-1}$ and $\mathbf{h}_{t-1}$. The input gate regulates to what extent a new value $\mathbf{x}_t$ is transferred into the cell, the forget gate controls to what extent $\mathbf{C}_{t-1}$ remains in the cell and the output gate regulates to what extent $\mathbf{C}_{t-1}$ is used to calculate the output activation. The structure of a standard LSTM cell is shown in Figure 1.

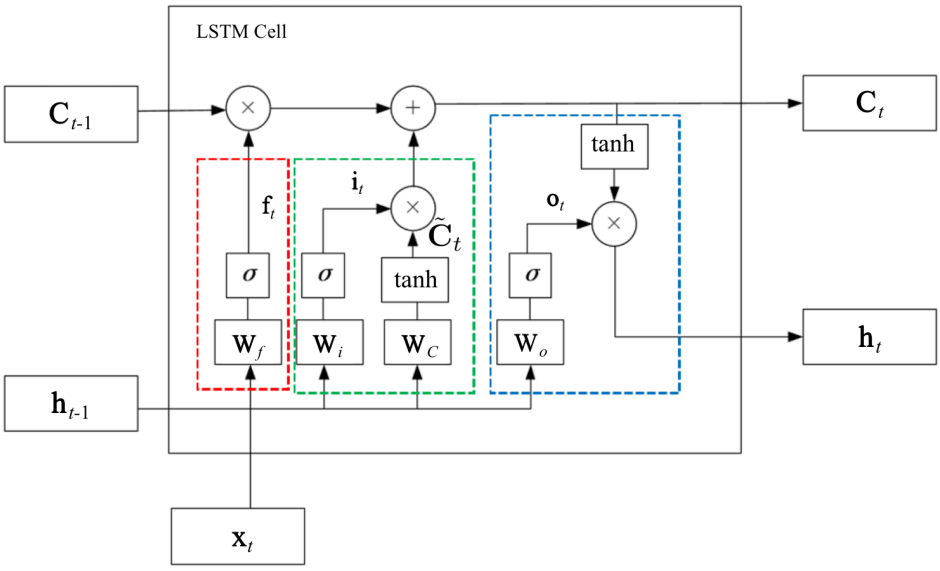

Figure 1. Structure diagram of LSTM cell.

In Figure 1, the green box, blue box and red box correspond to the input gate, the output gate and the forget gate respectively. The mathematic formulation of the forget gate is described as:

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f\right) \tag{1}$$

where $\mathbf{W}_f$ is the weight matrix of the forget gate; σ is the sigmoid activation function; $\mathbf{b}_f$ is the bias vector for the forget gate; $[\mathbf{h}_{t-1}, \mathbf{x}_t]$ is a vector that merge the previous cell state vector $\mathbf{h}_{t-1}$ and the input vector $\mathbf{x}_t$ at the current moment. The input gate is used to decide what information will be saved in the cell value. On the other hand, the input gate can be described mathematically as follows.

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right) \tag{2}$$

$$\tilde{\mathbf{C}}_t = \tanh\left(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right) \tag{3}$$

Here, $\mathbf{W}_i$ is the weight matrix in the input gate, $\mathbf{b}_i$ is the bias vector. The input gate adds new information generated by the current input to the cell value, and creates new memories: $\mathbf{i}_t$ and $\tilde{\mathbf{C}}_t$. The current state $\mathbf{C}_t$ is updated based on the previous cell value $\mathbf{C}_{t-1}$, the new memories $\mathbf{i}_t$ and $\tilde{\mathbf{C}}_t$ as follows.

$$\mathbf{C}_t = \mathbf{f}_t \cdot \mathbf{C}_{t-1} + \mathbf{i}_t \cdot \tilde{\mathbf{C}}_t \tag{4}$$

Finally, the hidden state h_t is updated in the output gate as:

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right) \tag{5}$$

$$\mathbf{h}_t = \mathbf{o}_t \cdot \tanh\left(\mathbf{C}_t\right) \tag{6}$$

where $\mathbf{W}_o$ is the weight matrix in the output gate. In this way, the cell value $\mathbf{C}_t$ and hidden state $\mathbf{h}_t$ can be updated whenever a new sample $\mathbf{x}_t$ is available.

2.1.2. The Structure of GRU Cell

The GRU is a refined version of LSTM with a simpler structure [19]. The main difference between GRU and LSTM is in the process of forgetting and updating cell values. In the LSTM network, update of cell values are controlled by two gates, the forget gate and the input gate. Since two gate structures are required, the structure of LSTM is relatively complex. Compared to LSTM, GRU controls both the forgetting coefficient and the update coefficient for the output with one single update gate, so it involves fewer matrix multiplication calculations. Through this simplification, the GRU can retain the functions of the LSTM and reduce network training time. More specifically, it consists of an update gate and a reset gate, which reduces the number of parameters to only one fourth of the LSTM. The reset gate determines how much previous memory is retained and the update gate determines how much new information needs to be combined with the previous memory. The structure of GRU cell is shown in Figure 2.

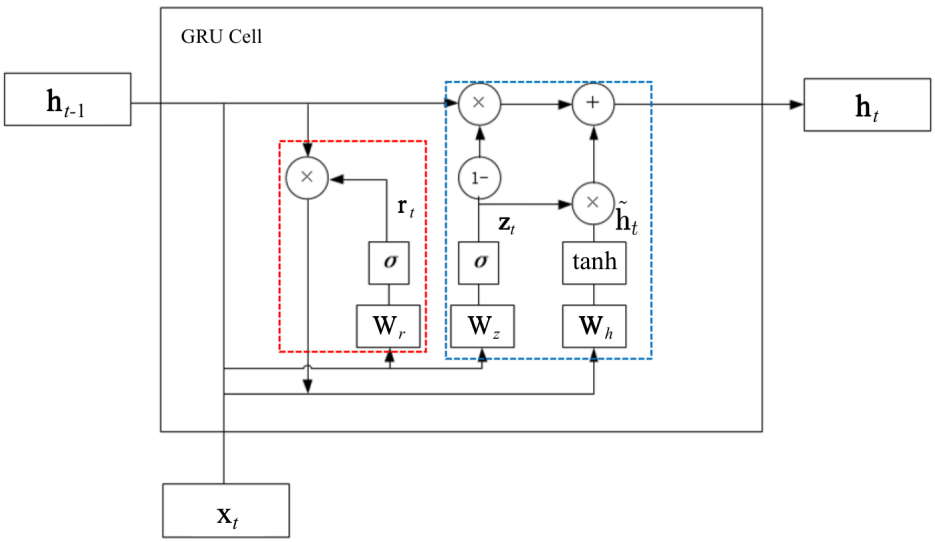

Figure 2. Structure diagram of Gated Recurrent Unit (GRU) cell.

In contrast to LSTM, GRU has only 2 gate functions. The update gate is shown in the blue box in Figure 2 and the reset gate is shown in the red box. The forward transfer formulations of GRU can be calculated as follows.

$$\mathbf{r}_t = \sigma\left(\mathbf{W}_r \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \tag{7}$$

$$\mathbf{z}_t = \sigma\left(\mathbf{W}_z \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t]\right) \tag{8}$$

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{W}_{\tilde{h}} \cdot [\mathbf{r}_t \cdot \mathbf{h}_{t-1}, \mathbf{x}_t]\right) \tag{9}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \cdot \mathbf{h}_{t-1} + \mathbf{z}_t \cdot \tilde{\mathbf{h}}_t \tag{10}$$

where $\mathbf{r}_t$ is the reset gate determining how much information in the previous state cell should be forgotten; $\mathbf{z}_t$ is the update gate determining how much information should be brought to the next cell; $\tilde{\mathbf{h}}_t$ is the intermediate state; $\mathbf{h}_t$ is hidden state. For the update gate, a greater value of $\mathbf{z}_t$ means that

more new information is brought to the next cell. For the reset gate, a greater value means that more information from the former cell may be ignored [16].

2.2. Support Vector Data Description

SVDD is a kernel method, which maps the data samples into the high-dimensional feature space through a non-linear mapping. In the high-dimensional feature space, a compact hypersphere with the minimum radium while covering the maximum number of data samples is obtained by solution of an optimization problem. The SVDD is generally used in anomaly detection. If a new sample is mapped inside the hypersphere, it is regarded as a normal sample, otherwise it is faulty.

Given a data set $\left\{\mathbf{x}_i \in \mathcal{R}^d, i = 1, \cdots, N\right\}$ and assume $\mathbf{a} \in \mathcal{R}^d$ the center of the hypersphere, R is the radius of the hypersphere, the following objective function can be obtained for SVDD.

$$\begin{cases} F\left(R, \mathbf{a}, \xi_i\right) = R^2 + C\sum_{i=1}^{N} \xi_i \\ \left\|\mathbf{x}_i - \mathbf{a}\right\|^2 \leq R^2 + \xi_i \end{cases} \tag{11}$$

Here, ξ_i is the relaxation factor, and C is the penalty parameter. In Equation (11), ξ_i satisfies $\xi_i \geq 0, \forall i$. The above optimization problem can be transformed as follows using the Lagrangian multipliers.

$$L\left(R, \mathbf{a}, \alpha, \gamma_i, \xi_i\right) = R^2 + C\sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \gamma_i \xi_i - \sum_{i=1}^{N} \alpha_i \left[R^2 + \xi_i - \left(\left\|\mathbf{x}_i\right\|^2 - 2\mathbf{a}\mathbf{x}_i + \left\|\mathbf{a}\right\|^2\right)\right] \tag{12}$$

where γ_i and α_i is the Lagrange multiplier and they satisfy $\alpha_i \geq 0, \gamma_i \geq 0$. Differentiate Equation (12) with respect to R, $\mathbf{a}$ and ξ_i and make it equal to 0, the following holds:

$$\begin{cases} \frac{\partial L}{\partial R} = 0 \\ \frac{\partial L}{\partial \mathbf{a}} = 0 \\ \frac{\partial L}{\partial \xi_i} = 0 \end{cases} \Longrightarrow \begin{cases} \sum_{i=1}^{N} \alpha_i = 1 \\ \mathbf{a} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \\ C - \alpha_i - \gamma_i = 0 \end{cases} \tag{13}$$

Combining Equation (13) to Equation (12) one can obtain:

$$L = \sum_{i=1}^{N} \alpha_i \left(\mathbf{x}_i \cdot \mathbf{x}_i\right) - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \tag{14}$$

where α_i is the support vector and $0 \leq \alpha_i \leq C$. Generally, kernel function K is used to calculate whether the distance between the new sample $\mathbf{y} \in \mathcal{R}^d$ and the center of the hypersphere is less than the radius R^2:

$$D^2\left(\mathbf{y}\right) = K\left(\mathbf{y} \cdot \mathbf{y}\right) - 2\sum_{i=1}^{N} \alpha_i K\left(\mathbf{y} \cdot \mathbf{x}_i\right) + \sum_{i,j=1}^{N} \alpha_i \alpha_j K\left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \leq R^2 \tag{15}$$

The kernel term $K\left(\mathbf{x}_i \cdot \mathbf{x}_j\right)$ is commonly used to replace the inner product $\left(\mathbf{x}_i \cdot \mathbf{x}_j\right)$, which is the Gaussian kernel here:

$$K\left(\mathbf{x}_i \cdot \mathbf{x}_j\right) = \exp\left(-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma^2}\right) \tag{16}$$

3. Fault Detection and Identification Strategy

In order to detect and identify a process fault, it is essential to characterize the normal operating condition (NOC). Hence, a training dataset collected under normal operational condition is used to construct the GRU neural network. The GRU neural network generates model residuals, which is further used to construct monitoring statistics using SVDD. As described earlier, the GRU model is capable of extracting the spatial and temporal signatures in the data that are important

for characterizing complex ironmaking process. The general framework for fault detection and identification based on GRU-SVDD is described in detail in the following subsections.

3.1. Fault Detection

In order to detect a process fault, it is required to train a model based on the NOC data. In the ironmaking process, this involves training a GRU with multiple time series to model temporal dynamics and correlations between process variables.

The GRU model is trained on historical normal data. Specifically, the GRU model uses the past information captured by its cell value and current observation to predict the next observation. Assume a training set $\left\{\mathbf{x}_i \in \mathcal{R}^d, i = 1, \cdots, N\right\}$ is collected under NOC, a moving window approach can be applied, with the window length being $n, n \ll N$. Take the first window as an example, the structure of a two-layer GRU is shown in Figure 3.

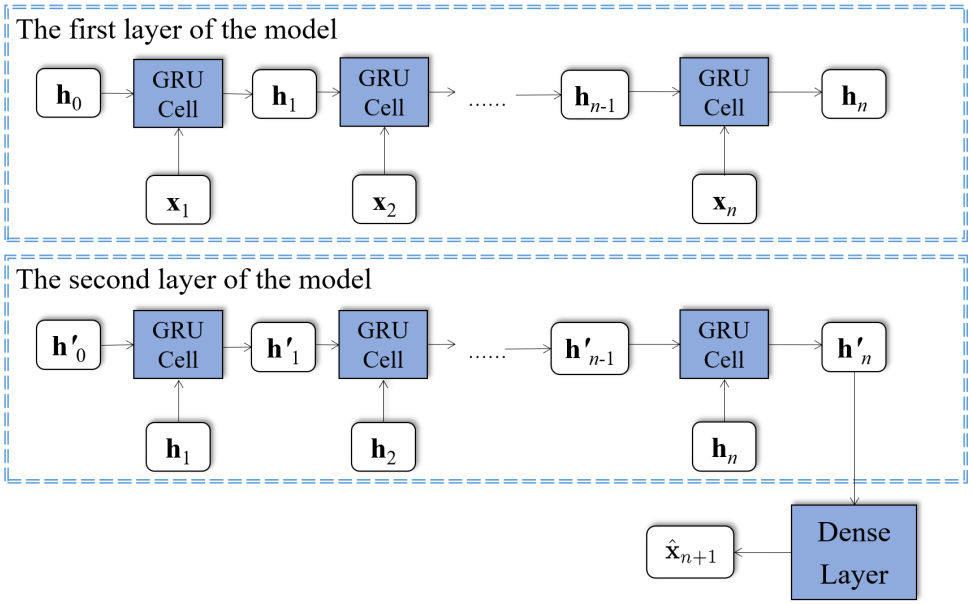

Figure 3. Structure of two layers GRU.

Here, $\mathbf{h}_i \in \mathcal{R}^{d_h}$ denotes the hidden state of the first layer at the ith time, $\mathbf{h}_i^{'} \in \mathcal{R}^{d_h^{'}}$ denotes the hidden state of the second layer and $\hat{\mathbf{x}}_{n+1} \in \mathcal{R}^d$ is the predicted value. The hidden state $\mathbf{h}_i$ of the first layer becomes the input to the second layer of GRU model. The final output is then obtained using the dense layer as follows.

$$\hat{\mathbf{x}}_{n+1} = \mathbf{W}^{'}\mathbf{h}_n^{'} + \mathbf{b}^{'} \tag{17}$$

Here $\hat{\mathbf{x}}_{n+1}$ is the prediction of $\mathbf{x}$ at the $n+1$th time instance, $\mathbf{W}^{'}$ is the weight matrix of the dense layer, $\mathbf{b}^{'}$ is the bias term. A GRU model with more layers can also be used, for the sake of simplicity, however two-layer GRU is considered here.

The model parameters can be trained based on the $N - n + 1$ windows. Once the model parameters are estimated, estimation of model output $\hat{\mathbf{x}}_{n+1}$ can be predicted from the past n samples. A series of residuals can be obtained as $\mathbf{e}_i = |\hat{\mathbf{x}}_i - \mathbf{x}_i|, i = n+1, \cdots, N$. The residual series obtained from GRU under NOC is then fed into the SVDD to estimate the parameters, namely the center $\mathbf{a}$ and the radius R of the hypersphere. Whenever a new sample is available, the residuals obtained from the GRU can be fed into the SVDD to calculate the the squared distance D^2 according to Equation (15). If D^2 is greater than R^2, it is faulty, otherwise it is normal.

3.2. Fault Identification

Once a fault is detected, the next goal is to identify which variables are the most affected and contribute most to the monitoring statistics. Assume a fault was detected between time t_1 and t_2, let $\mathbf{e}_i = (e_i^1, e_i^2, \cdots, e_i^d)$ denote the residual vector for the l process variables, $i = t_1, \cdots, t_2$. The normalized residuals E_i^l can be used to evaluate the impact of the fault on each variable as:

$$E_i^l = \frac{e_i^l - \mu^l}{\sigma^l} \tag{18}$$

where μ^l is the mean value and σ^l is the standard deviation of the GRU residuals of the NOC training data.

For a clearer exhibition, the deviation E_i^l of each variable is accumulated to get the total contribution rate $CR^l = \sum_{i=t}^{N} E_i^l$. With the contribution rate obtained, operators can know which variables are most sensitive to the process fault. Also, operators can use the contribution plots to identify which kind of fault has occurred.

For completeness, the overall flowchart is summarized in Figure 4, including both the offline training stage (left) and the online monitoring stage (right).

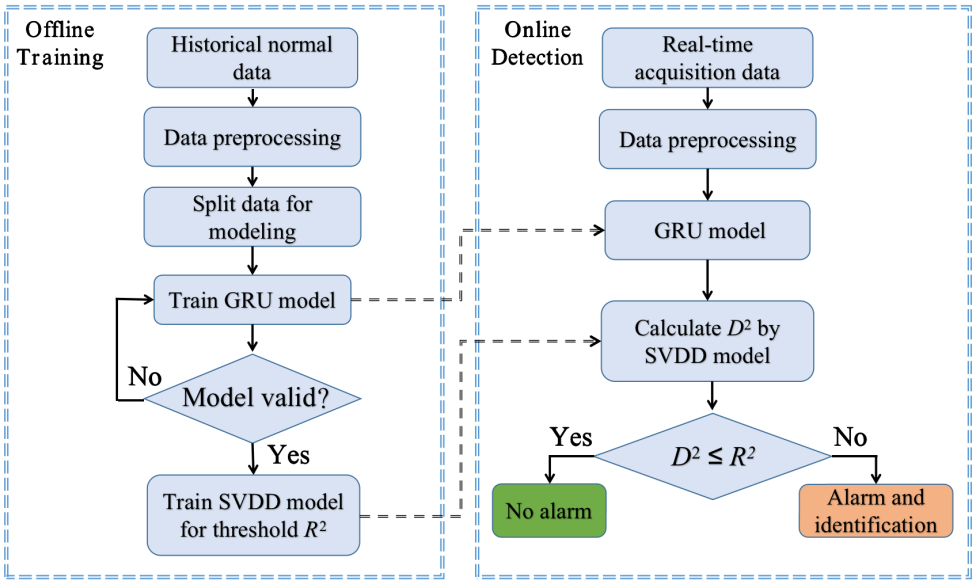

Figure 4. Flowchart of the GRU-support vector data description (SVDD)-based fault detection and identification methodology.

The offline monitoring stage can be summarized as follows:

1. Obtain historical NOC data;
2. Remove extreme values and normalize the training data to have a zero mean and unit variance.
3. Set initial parameters of GRU model and train the model;
4. If the GRU model is valid, the GRU residuals will be fed into the SVDD model, and the threshold R^2 of D^2 statistic is obtained.

The online detection stage can be summarized as follows:

1. Collect online samples;
2. Normalize the online samples;
3. Use the GRU model trained in the offline process to make prediction and get the residuals;
4. Calculate the D^2 statistic using SVDD;

5. Determine whether to alarm by comparing the D^2 statistic and the threshold R^2. If D^2 is greater than R^2, the process is faulty, otherwise it is normal.
6. If the process is faulty, isolate and identify which variables are most severely affected.

4. Application Studies

This section presents the application results of the proposed GRU-based fault detection and identification method to the datasets collected from a blast furnace (with the inner volume of 2500 m^3) in China. Two case studies are studied, with Section 4.1 introduces the application of GRU-based fault detection and identification method to a hanging fault and Section 4.2 presents the application results to a fault involving fluctuation in molten iron temperature.

4.1. Case 1: Hanging Fault

The hanging fault happens in the upper part of the blast furnace, the fault caused a severe drop in the quantity of blast u_1 and pressure of blast u_3, which subsequently resulted in an abnormal change in the composition of flue gas u_5, u_6, u_7. For the purpose of model training, 2000 samples are collected under the normal operation condition and a faulty dataset containing 400 samples is considered. The sampling interval of the data is 20 min. A total of 7 process variables are considered and listed in Table 1. For comparison, the LSTM-SVDD and PCA-SVDD [20] methods are considered.

Table 1. The input variables for Case 1.

No.	Variable
u_1	quantity of blast
u_2	temperature of blast
u_3	pressure of blast
u_4	the quantity of oxygen blasted
u_5	CO concentration in top gas
u_6	CO_2 concentration in top gas
u_7	H_2 concentration in top gas

4.1.1. Residual Generation Using the GRU Network

In order to reduce the impact of extreme values in the process data, the Hampel filter [21] is used to process the training set before feeding into the GRU network. During the training of GRU network, the mean square error loss function and 'Adam' optimizer are used [22]. The length of moving window n is set as 99 by trial and error. The number of hidden states in the first layer d_h and the second layer $d_h^{'}$ are determined in a similar way. Figure 5 shows the modelling errors of the GRU network for u_1 under different combinations of d_h and $d_h^{'}$.

Considering both the modelling error and structure complexity, the fourth combination in Figure 5 is used so that $d_h = 32$ and $d_h^{'} = 200$. Also, to prevent overfitting, a dropout process is used in the training process by randomly discarding a part of units. Here, the dropout rate of $p_d = 0.2$ is selected. The GRU network uses the past values to predict current values. The predicted values obtained by this model not only contains the past information, but also affected by other related variables. Therefore, when a fault happens, the predictions will deviate from the actual values. The modeling results of the GRU model are shown in Figure 6.

Figure 6 shows that there are some clear changes occurring in several variables (e.g., CO concentration, CO_2 concentration and H_2 concentration). the obtained residuals are then fed into the SVDD model to perform fault detection.

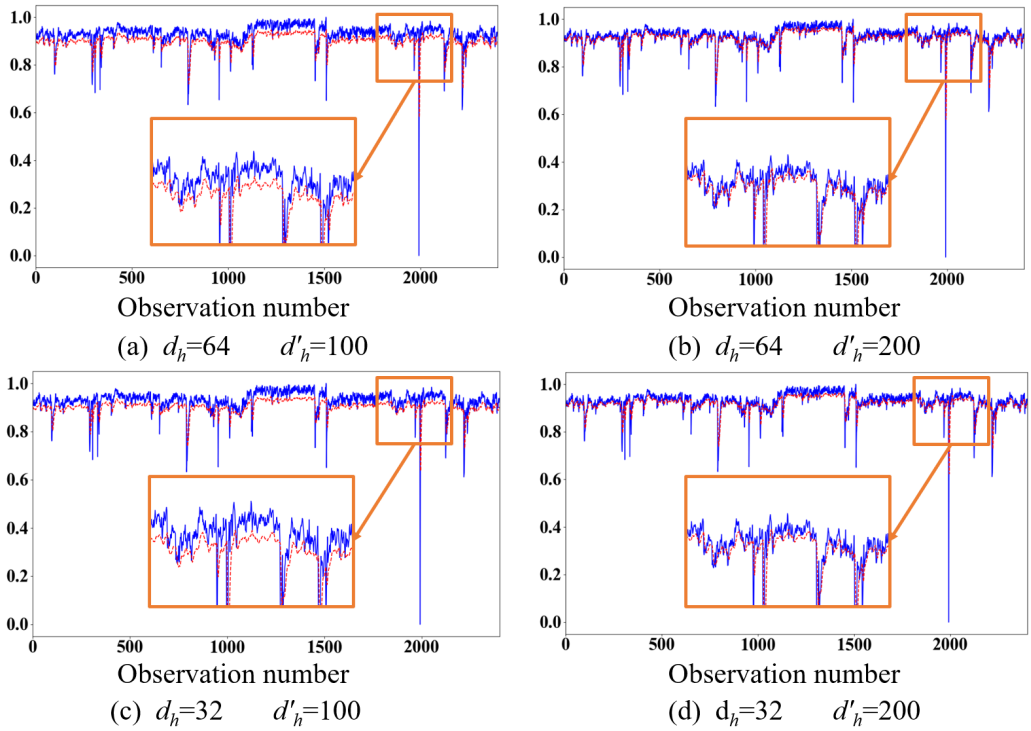

Figure 5. Prediction results for u_1 using models with different parameter settings (the red line corresponding to predictions, the blue line corresponding to true values). (**a**) Prediction results for u_1 with $d_h = 64$ and $d'_h = 100$; (**b**) Prediction results for u_1 with $d_h = 64$ and $d'_h = 200$; (**c**) Prediction results for u_1 with $d_h = 32$ and $d'_h = 100$; (**d**) Prediction results for u_1 with $d_h = 32$ and $d'_h = 100$.

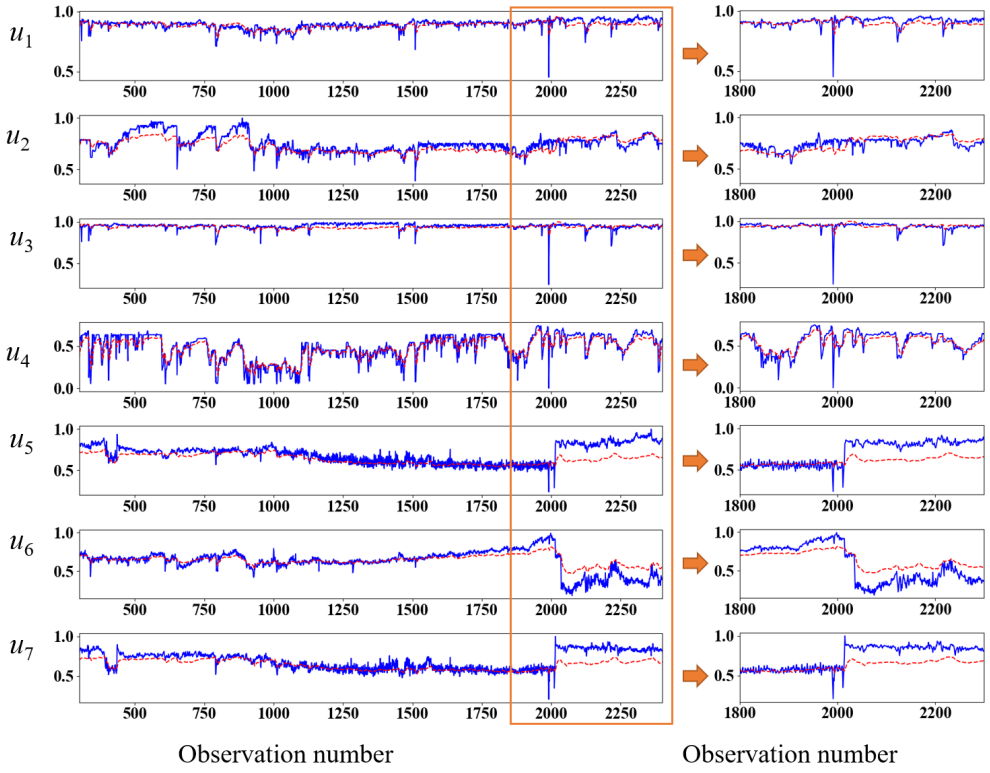

Figure 6. The prediction results of the GRU model in Case 1 (the red lines represent prediction and the blue lines represent actual values).

4.1.2. Fault Detection and Identification

From the previous subsection, the GRU model can be used to generate residuals. As is shown in the Figure 6, there is an obvious change after the 2000th sample. In order to detect this change, SVDD is used here. The parameters of SVDD are set as $\sigma = 10$, $C = 0.01$. With 99% of confidence limit, the monitoring results using SVDD are shown in Figure 7a.

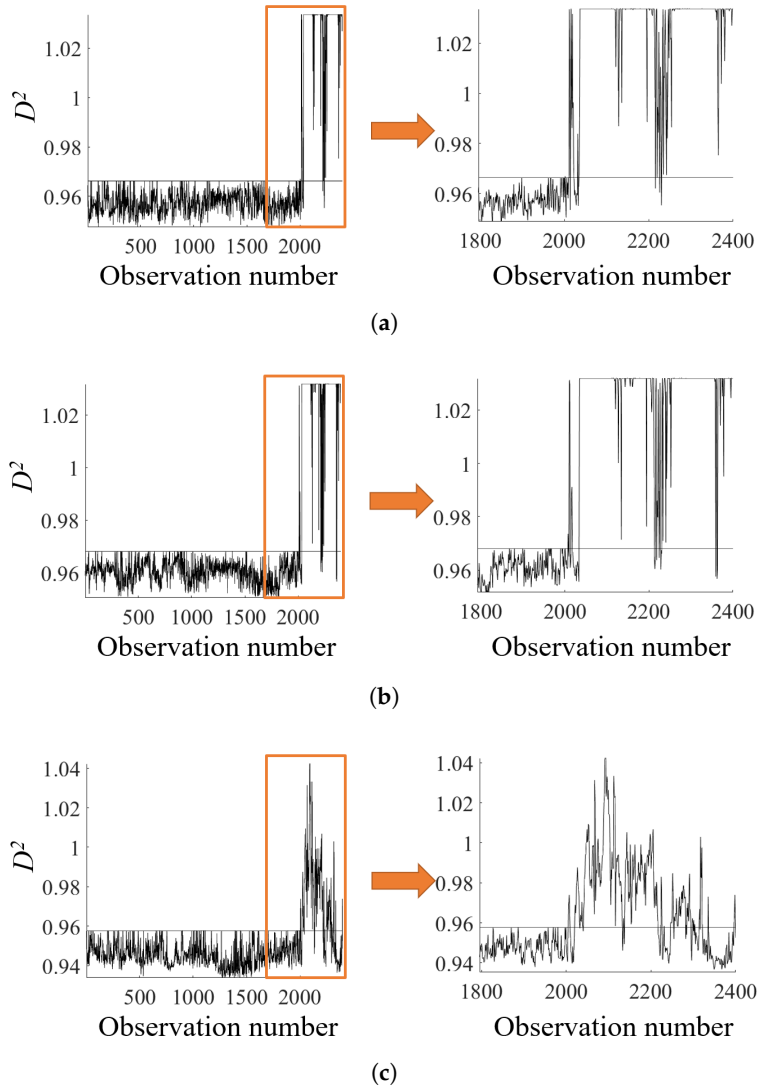

Figure 7. Monitoring results for the hanging fault. (**a**) Monitoring results of GRU-SVDD, (**b**) Monitoring results of LSTM-SVDD, (**c**) Monitoring results of principal component analysis (PCA)-SVDD.

From Figure 7a it can be seen that significant violation of the confidence limit can be observed, indicating that there is a fault happening in the blast furnace system. This is in accordance with the fact that the last 400 samples correspond to a hanging fault. For comparison, the monitoring results using LSTM-SVDD and PCA-SVDD are shown in Figure 7b,c. The LSTM network has the same structure and parameters as GRU-SVDD. For PCA-SVDD, PCA is first performed on the training data and SVDD is used to detect the residual subspace. The number of principal components retained for PCA is 3. Comparing Figure 7a–c, it can be seen that GRU-SVDD and LSTM-SVDD has higher sensitivity than PCA-SVDD. The detection rates of the three methods are shown in Table 2.

Table 2. Comparison of detection rate for different methods.

Methods	Detection Rate
$D^2_{GRU-SVDD}$	93.52%
$D^2_{LSTM-SVDD}$	92.27%
$D^2_{PCA-SVDD}$	72.82%

Table 2 confirms the finding that GRU-SVDD and LSTM-SVDD have better detection rates. Considering the simpler structure of GRU, obviously GRU-SVDD is a better method. After the hanging fault is detected, fault identification is then performed based on the GRU residuals. Figure 8 shows the sample by sample GRU residuals, with deeper color indicating greater residuals.

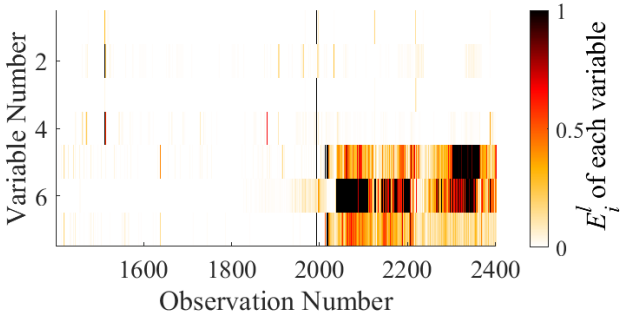

Figure 8. The sample by sample normalized GRU residuals in Case 1.

For a clearer inspection, Figure 9 presents the accumulated normalized GRU residuals. Figure 9 shows that the hanging fault has significant impact on the concentration of flue gas, with the most significant change happening in the CO_2, CO, H_2 concentration. This will lead experienced operators to inspect the gas flow and see whether there is any kind of hanging fault happening in the system.

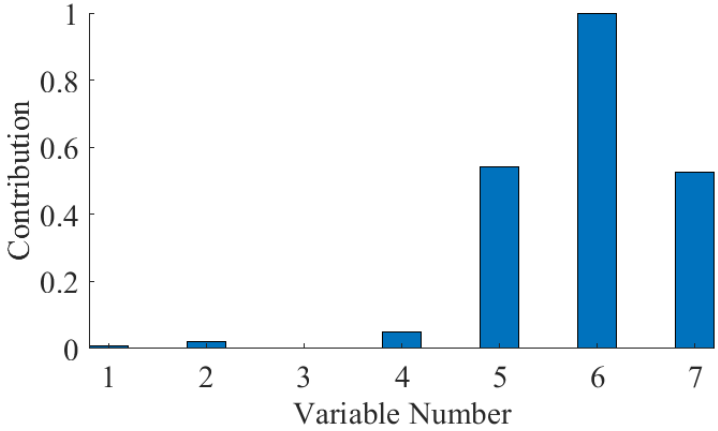

Figure 9. The fault contribution rate for each variable in Case 1.

4.2. Case 2: Abnormal Molten Iron Temperature

In this subsection, a faulty condition from the same blast furnace is considered. The fault involves an abnormal fluctuation of the molten iron temperature, which caused the operators to adjust the quantity of blast u_1 as well as the temperature of blast u_2, resulting in change in a series of variables. In the later stage, the fault was corrected, however the temperature of blast was kept at a relatively low level for the sake of safety. Similar to the hanging fault, 2000 samples were collected under the normal operating conditions for model training, and a faulty dataset containing 1000 samples is considered. The fault involves an abnormal molten iron temperature, which caused reduction in blast quantity,

blast temperature and fluctuation in a series of variables related to the gas flow. This time, 10 process variables are considered and listed in Table 3.

Table 3. The input variables for Case 2.

No.	Variable
u_1	quantity of blast
u_2	temperature of blast
u_3	pressure of blast
u_4	quantity of oxygen blasted
u_5	temperature of cold blast
u_6	top pressure
u_7	CO concentration in top gas
u_8	CO_2 concentration in top gas
u_9	H_2 concentration in top gas
u_{10}	pressure of cold blast

Comparing to Table 1, it can be seen that three additional variables, the temperature of cold blast (u_5), the top pressure (u_6) and the pressure of cold blast (u_{10}) are also included. it should be noted some of them are redundant variables (u_5 and u_{10}) that are highly related with other variables. The purpose for introducing these variables is to show the capability of the proposed method in dealing with variable redundancy.

Similar to Section 4.1, the proposed GRU method is applied, with the same parameter values. And the prediction results for the 10 variables are shown in Figure 10. For a clearer exhibition, only the 1000 faulty samples are presented. It can be seen that for the first 200 samples, the prediction accuracy is acceptable. After that, an obvious fluctuation can be observed and the prediction accuracy deteriorated. After the 2450th sample, the prediction accuracy for all variables except u_2 return normal.

After the predictions are obtained, SVDD is applied and the monitoring results are shown in Figure 10b. It can be seen that the fault was successfully detected since significant number of violations can be observed after the 2250th sample. This again indicates the good capability for the proposed method in fault detection. It should be noted that violations can still be observed even after the fault was corrected after 2450th sample. This can be explained, as to avoid further fault, the operators decided to reduce the temperature of blast(u_2), which caused the violations. This can be confirmed by the subsequent fault identification results in Figure 11.

In Figure 11, the first plot involves the fault identification results from samples from 2250 to 2450, while the second plot involves the identification results for samples from 2450 till 3000. As can be seen from the first plot of Figure 11, it can be clearly seen all variables except u_4 have significant contribution to the fault, indicating a significant anomaly arises. This is expected, as to correct the fault, both the quantity of blast and the temperature of blast are reduced, resulting in changes in other variables. From the second plot, it can be seen that after the fault was corrected, the contribution of other variables reduced significantly while that of u_2 remains. This is in accordance with our previous analysis that the operators reduced the temperature of blast to avoid further fault. The application results of the second faulty case also confirmed the performance of the proposed method.

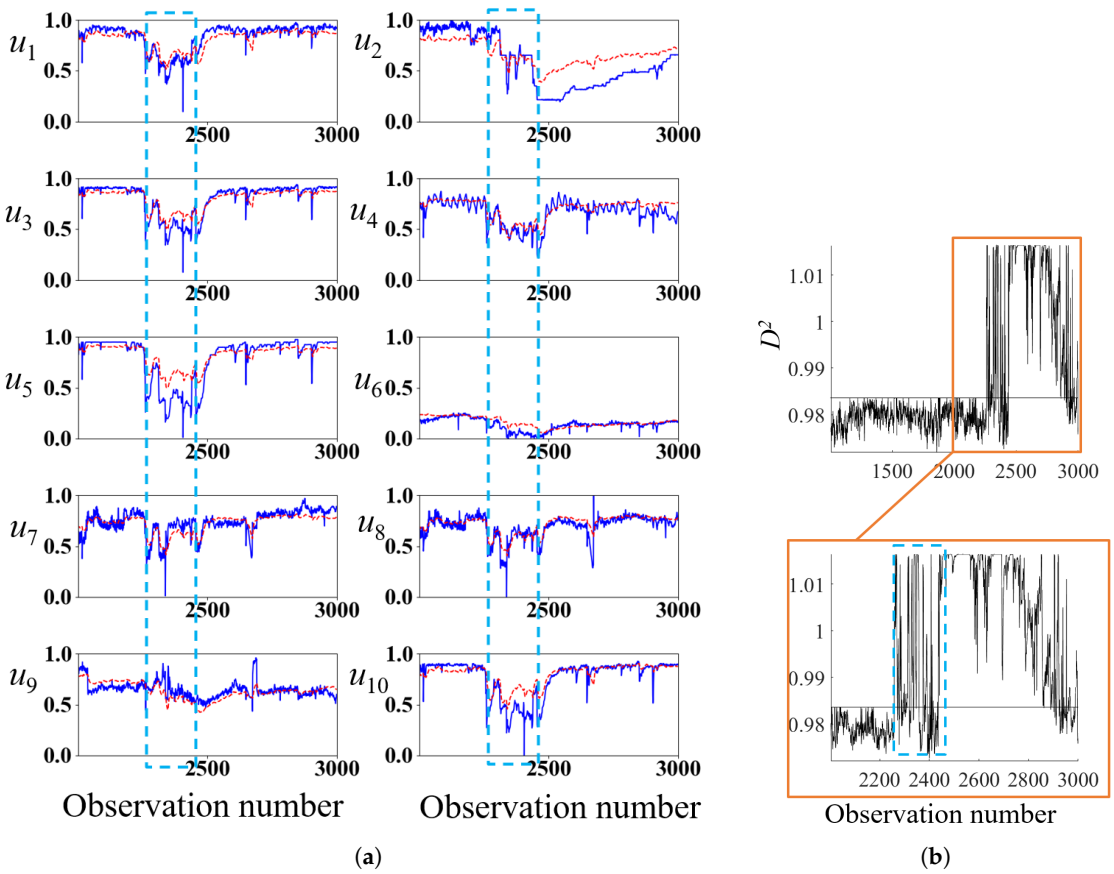

Figure 10. Prediction results using GRU and monitoring results using SVDD for Case 2. (**a**) Prediction results using GRU, (**b**) Monitoring results using SVDD.

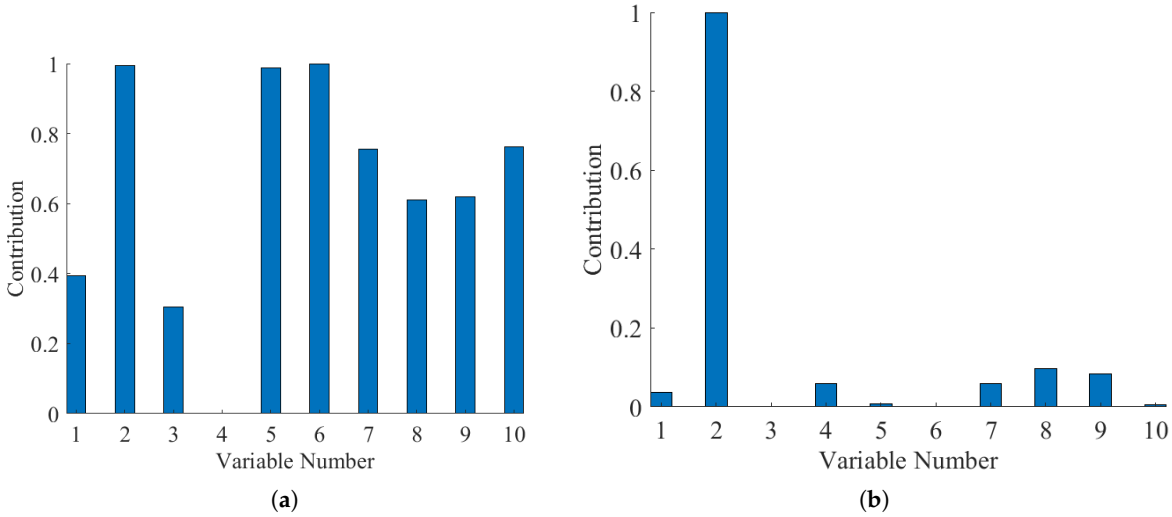

Figure 11. The fault contribution rate for each variable in Case 2. (**a**) The Contribution of different variables in observations 2250 to 2450, (**b**) The Contribution of different variables in observations 2450 to 3000.

5. Conclusions

This paper introduces a fault detection and identification method for blast furnace ironmaking process based on the GRU network and SVDD. The GRU model is capable of handling multi-dimensional inputs to make predictions for future inputs. The residuals between the actual

inputs and predictions are then monitored using SVDD. A fault identification method is further developed by inspecting the accumulated normalized residuals. The proposed method is tested on a hanging fault observed in a real blast furnace in China. Application results show that the proposed GRU-SVDD model can successfully detect the hanging fault. Compared with the PCA-SVDD model, GRU-SVDD has a higher detection rate. The method proposed in this article is very suitable for monitoring systems with strong dynamics and non-Gaussianity.

Author Contributions: Conceptualization, J.Z. and Y.L.; methodology, H.O. and J.Z.; validation, H.O. and Y.L.; formal analysis, H.O.; investigation: H.O.; resources, J.Z. and S.L.; data curation, J.Z.; writing—original draft preparation, H.O.; writing—review and editing, J.Z.; visualization, H.O.; supervision, J.Z. and S.L.; project administration, Y.L.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to thank financial support from National Natural Science Foundation of China (Grant Nos. 61673358 and 61973145).

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

References

1. Amano, S.; Takarabe, T.; Nakamori, T. Expert system for blast furnace operation at Kimitsu works. *ISIJ Int.* **1990**, *30*, 105–110. [CrossRef]
2. Liao, S. Expert system methodologies and applications–a decade review from 1995 to 2004. *Exp. Syst. Appl.* **2005**, *28*, 93–103. [CrossRef]
3. Tian, H.; Wang, A. A Novel Fault Diagnosis System for Blast Furnace Based on Support Vector Machine Ensemble. *ISIJ Int.* **2010**, *50*, 738–742. [CrossRef]
4. Liu, L.; Wang, A.; Sha, M. Multi-class classification methods of cost-conscious LS-SVM for fault diagnosis of blast furnace. *Ind. J. Iron Steel Res. Int.* **2011**, *18*, 17–23. [CrossRef]
5. An, R.; Yang, C.; Zhou, Z.; Wang, L. Comparison of Different Optimization methods with support vecto machine for blast furnace multi-fault classification. *IFAC-PapersOnLine* **2015**, *48*, 1204–1209.
6. Vanhatalo, E. Multivariate process monitoring of an experimental blast furnace. *Qual. Reliab. Eng. Int.* **2010**, *26*, 495–508. [CrossRef]
7. Zhang, T.; Ye, H.; Wang, W. Fault diagnosis for blast furnace ironmaking process based on two-stage principal component analysis. *ISIJ Int.* **2014**, *54*, 2334–2341. [CrossRef]
8. Shang, J.; Chen, M.; Zhang, H. Increment-based recursive transformed component statistical analysis for monitoring blast furnace iron-making processes: An index-switching scheme. *Control Eng. Pract.* **2018**, *77*, 190–200. [CrossRef]
9. Pan, Y.; Yang, C.; An, R. Robust principal component pursuit for fault detection in a blast furnace process. *Ind. Eng. Chem. Res.* **2017**, *57*, 283–291. [CrossRef]
10. Zhou, B.; Ye, H.; Zhang, H. Process monitoring of iron-making process in a blast furnace with PCA-based methods. *Control Eng. Pract.* **2016**, *47*, 1–14. [CrossRef]
11. Cai, J.; Zeng, J.; Luo, S. A state space model for monitoring of the dynamic blast furnace system. *ISIJ Int.* **2012**, *52*, 2194–2199. [CrossRef]
12. Vanhatalo, E.; Kulahci, M. Impact of autocorrelation on principal components and their use in statistical process control. *Qual. Reliab. Eng. Int.* **2015**, *32*, 1483–1500. [CrossRef]
13. Dong, Y.; Qin, S.J. A novel dynamic pca algorithm for dynamic data modeling and process monitoring. *J. Process Control.* **2018**, *67*, 1–11. [CrossRef]
14. Chiang, L.; Braatz, R.; Russell, E.L. *Fault Detection and Diagnosis in Industrial Systems*; Springer Science & Business Media; Berlin, Germany, 2002; Volume 44, 197–198.
15. Qin, S. Survey on data-driven industrial process monitoring and diagnosis. *Annu. Rev. Control* **2012**, *36*, 220–234. [CrossRef]
16. Wang, J.; Yan, J.; Li, C. Deep heterogeneous GRU model for predictive analytics in smart manufacturing: Application to tool wear prediction. *Comput. Ind.* **2019**, *111*, 1–14. [CrossRef]
17. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
18. Funahashi, K.; Nakamura, Y. Approximation of dynamical systems by continuous time recurrent neural networks. *Neural Netw.* **1993**, *6*, 801–806. [CrossRef]

19. Kim, P.; Lee, D.; Lee, S. Discriminative context learning with gated recurrent unit for group activity recognition. *Pattern Recognit.* **2018**, *76*, 149–161. [CrossRef]
20. Li, G.; Hu, Y.; Chen, H. An improved fault detection method for incipient centrifugal chiller faults using the PCA-R-SVDD algorithm. *Comput. Sci.* **2014**, *116*, 104–113. [CrossRef]
21. Pearson, R.; Neuvo, Y.; Astola, J.; Gabbouj, M. Generalized Hampel filters. *EURASIP J. Adv. Signal Process.* **2016**, *87*. [CrossRef]
22. Chang, Z.; Zhang, Y.; Chen, W. Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. *Energy* **2019**, *187*. [CrossRef]

Sample Availability: Samples of the compounds are available from the authors.