processes

MDPI

Article

Online Monitoring of Flowmeter Anomaly in Tobacco Production Process Using Sliding Window Recursive Lasso

Ziyi Guan [1], Suijun Liu [2], Ying Liu [2,*], Ting Cui [3], Linchao Yang [2], Jinhui Cai [1], Bin Liu [2], Yuhao Liu [2] and Jinming Li [2]

1 College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China; p21020854037@cjlu.edu.cn (Z.G.); caijinhui@cjlu.edu.cn (J.C.)
2 Nanyang Cigarette Factory, China Tobacco Henan Industrial Co. Ltd., Nanyang 473007, China
3 Technology Center, China Tobacco Henan Industrial Co. Ltd., Zhengzhou 450000, China
* Correspondence: hnnylyzzuli@163.com; Tel.: +86-15660017029

Abstract: Ensuring the accuracy of flow measurement is crucial to promoting high-quality cigarette production. In order to monitor the working status of flowmeters, this paper proposes an anomaly detection method based on the sliding-window recursive Lasso (Least absolute shrinkage and selection operator), which is able to track the changes in flowmeter operating conditions by self-adapting model parameters based on observed measurements. Due to the frequent mode switch and high sampling frequency of flow data, this paper introduces the sliding-window strategy to remove the effect of outdated data and accelerate the optimization. The tracking errors are used as a measure of anomaly and different thresholds are introduced based on the operating manual of cigarette production, which are used to distinguish between mode switch and flowmeter anomalies. The method's effectiveness is verified by detecting flowmeter anomalies in a real cigarette production line. The mean absolute error (MAE) is 8.1479 and the root mean squared error (RMSE) is 2.8544, which outperforms methods such as Lasso and the ridge regression.

Keywords: anomaly detection; recursive Lasso; sliding-window; flowmeter measurement

Citation: Guan, Z.; Liu, S.; Liu, Y.; Cui, T.; Yang, L.; Cai, J.; Liu, B.; Liu, Y.; Li, J. Online Monitoring of Flowmeter Anomaly in Tobacco Production Process Using Sliding Window Recursive Lasso. *Processes* **2023**, *11*, 1519. https://doi.org/10.3390/pr11051519

Academic Editor: Alexander Novikov

Received: 12 March 2023
Revised: 4 May 2023
Accepted: 6 May 2023
Published: 16 May 2023

1. Introduction

The production process of cigarettes is highly complex, and the blending process of tobacco with flavoring is a key subprocess that determines the quality of shredded cigarette products. This process significantly affects the smoking taste of the product and the release of volatile components in the smoke. The flavoring flowmeter is used to measure the volume of flavoring that is fed into the tobacco, which consists of different components like honey and perfumes.It is one of the most important sources of basic data acquisition in the traceability chain of shredded tobacco production, and is directly related to the accuracy of adding ingredients, flavoring, and controlling moisture content. The flavoring flowmeter is widely used in key process nodes of cigarette production.

As the flavoring flowmeter is used online, it will inevitably suffer from performance degradation like other instrumentations. Therefore, it is necessary to perform effective online anomaly monitoring during the tobacco manufacturing process to ensure that the measurement performance of the flavoring flowmeter is always within the allowable accuracy range. In traditional practice, the flavoring flowmeter is regularly removed from the production line for inspection and recalibration [1]. However, this practice has significant negative impacts as it introduces a halt in production. Additionally, the interval of recalibration is challenging to determine. Hence, performing online monitoring of the flowmeter using the data collected from the process is highly desirable.

In order to perform online monitoring of the flowmeter measurements, Sun et al. (2017) [2] proposed a self-diagnosis method by analyzing the vibration frequency of a Roots type flowmeter, and diagnosed the fault by inspecting whether the relationship between the instantaneous flow measurement and the vibration frequency was consistent. Zhao et al. (2021) [3] proposed an anomaly detection method based on the functional relationship between the pressure loss of the turbine gas flowmeter and the gas usage, and judged the working state of the flowmeter by the deviation degree of the pressure loss ratio. Wang et al. (2021) [4] developed a flowmeter anomaly detection method using a non-singleton Type 3 fuzzy logic system (FLS) to detect anomaly by comparing the measured terminal estimate signals. Sun et al. (2009) [5] established the relationship between the vortex energy ratio and the flow conditions, and proposed an anomaly detection method using the vortex energy ratio as the state diagnostic index of the vortex flowmeter. However, these methods typically require the installation of additional sensors or instrumentation, which is difficult and not applicable in processes like cigarette production.

Alternatively, researchers have attempted to apply multivariate statistical methods [6–9] to detect sensor failures. For instance, Tang et al. (2018) [6] proposed an anomaly detection and self-correction method for ultrasound flowmeter based on the particle swarm optimized support vector machine. Zhang et al. (2010) [7] introduced an anomaly detection method for flow accumulator using the least squares support vector machine, which detects anomaly by comparing the predicted value with the real-time output of the flowmeter. Chen et al. (2006) [10] performed wavelet analysis on high-frequency flow measurements to detect anomaly in flowmeter. Yang et al. (2006) [11] constructed a flow database using measurements obtained under normal operating conditions and identified the operating status of flowmeter by comparing the sum of squared errors of online measurements with the standard values. Despite the research progress, several issues related to online monitoring still exists: (i) Practical production processes are time-varying and involve frequent operational mode switches. An online monitoring method must be able to adapt to changes in the operational conditions.; (ii) The high sampling frequency of practical production requires the online monitoring method to be computationally efficient.

In this paper, an anomaly detection method for flavoring flowmeter in cigarette production based on the recursive Lasso algorithm [12] is proposed, which uses a sliding-window strategy to enhance modeling efficiency. By re-training the recursive Lasso using the latest flow measurements and removing outdated measurements, the sliding-window recursive Lasso can accurately track online flow measurements, and the resulting tracking errors can be used to detect anomalies in the flowmeter. The method also introduces error thresholds based on the cigarette production process knowledge, in order to differentiate between changes in process mode and flowmeter anomalies. The combination of sliding window strategy and online recursion offers significant advantages over traditional approaches, providing improved reliability in tracking accuracy and computational efficiency.

This rest of the paper is organized as follows. Section 2 introduces the sliding-window recursive Lasso method, with emphasis on how to update the parameter in an efficient way. In Section 3, an online anomaly monitoring method is developed based on the sliding-window recursive Lasso approach. In Section 4, the proposed method is applied to a real production line in a cigarette factory, which shows that the proposed method can correctly identify flowmeter anomaly. Finally, the conclusions of this work are presented Section 4.

2. Methodologies

This section describes the recursive Lasso used in this paper, with emphasis on how to update the parameters whenever new samples are available. Section 2.1 briefly introduces the classic Lasso technique, Section 2.2 describes how to recursively update the model parameters of Lasso, and Section 2.3 further modifies the recursive Lasso by introducing a sliding-window strategy, and proposes an anomaly detection method based on it.

2.1. Lasso Regression

The Lasso (Least Absolute Shrinkage and Selection Operator) regression [13] is a shrinkage estimation technique that can handle collinearity resulting from numerous influential factors. The Lasso method reduces the regression coefficients associated with irrelevant variables to zero, making it an effective variable selection tool. However, like any statistical method, the Lasso has specific assumptions that need to be satisfied for the results to be accurate. Some of the fundamental assumptions of the Lasso technique are as follows:

(1) Linearity: Lasso assumes that the relationship between the predictor variables and the response variable is linear.
(2) Independence: Lasso assumes that the observations are independent of each other. In other words, the value of one observation should not influence the value of another observation.
(3) Homoscedasticity: Lasso assumes that the variance of the errors is constant across all levels of the predictor variables. This is known as homoscedasticity.
(4) Normality: Lasso assumes that the errors are normally distributed.
(5) No multicollinearity: Lasso assumes that there is no perfect or near-perfect linear relationship between any pair of predictor variables. This is known as multicollinearity.

In this section, recursive Lasso is used to address the anomaly detection problem of flowmeter measurements by considering an autoregressive model. Hence a time-series problem is considered here. Assume the time-series problem involve $n-1$ time delays, the n-th sample y_n can be related to the previous $n-1$ samples. Let y_i be the i-th sample for $i = 1, 2, \ldots, n-1$, the linear relationship between y_n and its previous $n-1$ samples can be modeled by the following Lasso model.

$$\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\left\| y_n - \sum_{i=1}^{n-1} y_{n-i}\alpha_i \right\|_2^2 + \mu\|\boldsymbol{\alpha}\|_1 \quad (1)$$

Here $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_{n-1})^T$ is the regression coefficient vector, and the regression model is $y_n = \sum_{i=1}^{p} y_{n-i}\alpha_i + \varepsilon_i$, where ε_i is the noise. The $\|\cdot\|_1$ term denotes the l_1 penalty function and μ is the penalty parameter. The first term in Equation (1) is the least squares loss function, while the second term is the penalty function that shrinks the regression coefficients to obtain a sparse solution. The penalty parameter μ determines how much weight is put on the sparsity of the regression coefficients. The larger the value of μ, the stronger the sparsity is, and more zero elements can be obtained in $\boldsymbol{\alpha}$.

In order to efficiently solve the Lasso regression problem in Equation (1), Efron et al. [14] proposed the LARS (least angle regression) algorithm. The approximate solution $\hat{\boldsymbol{\alpha}}$ can be estimated as follows:

$$\hat{\boldsymbol{\alpha}} = \left(\mathbf{Y}^{\mathbf{T}}\mathbf{Y} + \mu\mathbf{W}^{-}\right)^{-1}\mathbf{Y}^{\mathbf{T}} y_n \quad (2)$$

Here $\mathbf{W} = \mathrm{diag}(|\hat{\alpha}_1|, \ldots, |\hat{\alpha}_{n-1}|)$ is a diagonal matrix, and $\mathbf{W}^{-}$ is the generalized inverse of matrix $\mathbf{W}$. The LARS algorithm's time complexity increases significantly with the number of data samples, making it unsuitable for online applications. To accelerate the calculation, the effective set method is often used, which reduces the computation by updating a subset of variables at each iteration.

It should be noted that the optimization problem in Equation (1) is not differentiable due to the presence of the l_1 norm, making it a non-smooth concave optimization problem. Thus, the global minimum in $\boldsymbol{\alpha}$ is only guaranteed if the objective function at $\boldsymbol{\alpha}$ contains a zero vector. The first-order derivative of the approximate solution $\hat{\boldsymbol{\alpha}}$ can be expressed as shown in Equation (3):

$$\partial\|\hat{\boldsymbol{\alpha}}\|_1 = \left\{ \mathbf{v} \in \mathbb{R}^m : \left\{ \begin{array}{c} v_i = \mathrm{sgn}(\hat{\alpha}_i), if|\hat{\alpha}_i| > 0 \\ v_i \in [-1,1], if|\hat{\alpha}_i| = 0 \end{array} \right\} \right\} \quad (3)$$

Therefore, the optimal approximate solution of Lasso can be written as:

$$\mathbf{Y}^{\mathbf{T}}(\mathbf{Y}\hat{\boldsymbol{\alpha}} - y_n) + \mu_{n-1}\mathbf{v} = 0, v_{n-1} \in \boldsymbol{\partial}\|\hat{\boldsymbol{\alpha}}\|_1 \quad (4)$$

If the indices of all non-zero elements in $\hat{\boldsymbol{\alpha}}$ are defined as an "effective-set", for the calculated $\hat{\boldsymbol{\alpha}}$ and $\boldsymbol{v}$, ordered by the arrangement of non-zero elements in front and zero elements in the back, we can get $\hat{\boldsymbol{\alpha}} = \left(\hat{\boldsymbol{\alpha}}_1^{\mathrm{T}}, 0^{\mathrm{T}}\right)^{\mathrm{T}}$, $\mathbf{v} = \left(\mathbf{v}_1^T, \mathbf{v}_2^T\right)^T$, and $\mathbf{Y}$ can be divided into $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)$. So that the optimal solution can be rewritten as follows.

$$\begin{cases} \hat{\boldsymbol{\alpha}}_1 = (\mathbf{Y_1}^{\mathbf{T}}\mathbf{Y}_1)^{-1}(\mathbf{Y_1}^{\mathbf{T}} y_n - \mu_n \mathbf{v}_1) \\ -\mu_n \boldsymbol{v}_2 = \mathbf{Y_1}^{\mathbf{T}}\left(\mathbf{Y}_1 \hat{\mathbf{ff}}_1 - y_n\right) \end{cases} \tag{5}$$

It is worth noting that $\boldsymbol{\alpha}$ can be computed in closed form if the "effective set" and the signs of the coefficients within the eigenvector $\boldsymbol{\alpha}$ are known.

2.2. Recursive Lasso

In industrial processes, sensor data often exhibits time-varying behavior due to changes in the underlying operating conditions. Static methods, such as Lasso, may not be able to effectively track these changes over time. Therefore, in this section, we use the recursive Lasso algorithm [12] to update the model parameters so that the predictions of the model remain reliable over time. The recursive Lasso algorithm is an extension of the original Lasso, it enables online and recursive feature selection and parameter estimation. Specifically designed for data that changes over time, recursive Lasso is well equipped to handle time-varying characteristics. It works by adding new data points to the existing dataset and re-estimating the Lasso coefficients using a homotopy method. By updating the model parameters in real-time as new data becomes available, the recursive Lasso method ensures that the model remains accurate and reliable over time. Additionally, the recursive Lasso algorithm assumes that the data follows a linear model, the noise is additive and has a normal distribution with zero mean and constant variance, the number of relevant features remains constant or increases over time, and that the data has a sparse representation. If these assumptions are satisfied, the recursive Lasso algorithm can effectively update the model parameters and feature selection in real-time, making it a valuable tool for tracking time-varying data in industrial processes. Please see Figure 1.

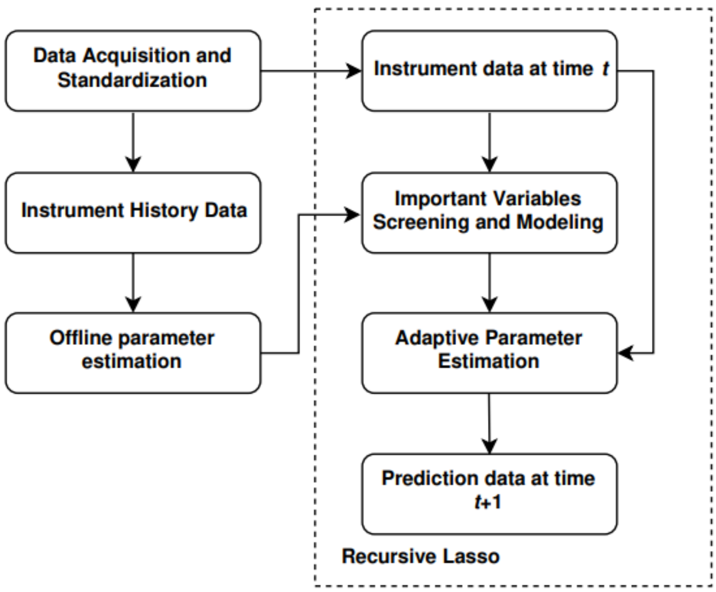


Figure 1. Flow chart of Recursive Lasso algorithm.

Given the Lasso solution $\hat{\boldsymbol{\alpha}}^{(n-1)}$ of Lasso at the $n-1$-th time, the recursive Lasso algorithm can update the model to obtain a new solution $\hat{\boldsymbol{\alpha}}^{(n)}$ using the n data samples and predict y_{n+1} at time $n+1$, enabling the algorithm to track changes in operational conditions recursively. To describe the optimization objective for a general case with delay of p, we define the data vector as $\mathbf{z}_n = (y_n, y_{n-1}, \ldots, y_{n-p})$ and represent the data matrix as $\mathbf{Z}_n = (z_1, z_2, \ldots, z_n)^T$. Here, $\mathbf{Y}_{n-1} = (y_1, y_2, \ldots, y_{n-1})^T$ is the response vector. The optimization objective for recursive Lasso is given as follows:

$$\hat{\boldsymbol{\alpha}}^{(n)}\left(t, \mu^{(n)}\right) = \underset{\boldsymbol{\alpha}}{argmin} \frac{1}{2} \left\| \begin{pmatrix} \mathbf{Y}_{n-1} \\ t y_n \end{pmatrix} - \begin{pmatrix} \mathbf{Z}_{n-1} \\ t \mathbf{z}_n \end{pmatrix} \boldsymbol{\alpha}^{(n)} \right\|_2^2 + \mu^{(n)} \left\| \boldsymbol{\alpha}^{(n)} \right\|_1 \tag{6}$$

Here, $\mu^{(n)}$ represents the penalty item parameter at time n, and $t \in [0, 1]$ is an identifier for the model update. If $t = 0$, the regression coefficient $\hat{\boldsymbol{\alpha}}^{(n-1)}$ has not started to be updated. When $0 < t < 1$, it means that the model is fitting a new data point. When $t = 1$, it means that the fit of the model is complete, and the regression coefficient $\hat{\boldsymbol{\alpha}}^{(n-1)}$ has been updated to $\hat{\boldsymbol{\alpha}}^{(n)}$. The algorithm then prepares to collect data at time $n + 1$ and starts updating at the next time.

The update process of formula Equation (6) can be summarized by the following steps.

STEP1 : $\left(\hat{\boldsymbol{\alpha}}^{(n-1)}(0, \mu^{(n-1)}) \to \hat{\boldsymbol{\alpha}}^{(n-1)}(0, \mu^{(n)})\right)$. If $t = 0$, use the minimum angle regression algorithm to update the penalty parameter $\mu^{(n-1)}$ to $\mu^{(n)}$;

STEP2: $\left(\hat{\boldsymbol{\alpha}}'^{(n-1)}(0, \mu^{(n)}) \to \hat{\boldsymbol{\alpha}}^{(n)}(1, \mu^{(n)})\right)$: If $\mu = \mu^{(n)}$, calculate the value of $\hat{\boldsymbol{\alpha}}$ after t changes from 0 to 1.

In STEP2, it is first necessary to prove that $\hat{\boldsymbol{\alpha}}(t, \mu^{(n)})$ is a smooth function of segments with respect to t. To make the notation easier, let $\boldsymbol{\alpha}(t) = \hat{\boldsymbol{\alpha}}(t, \mu^{(n)})$. When $0 \leq t \leq 1$, the "effective set" at time n is obtained by STEP1 and the signs of the non-zero coefficients in $\boldsymbol{\alpha}$ remain the same, making solution $\boldsymbol{\alpha}(t)$ of Lasso smooth. If at $t^* \in (0, 1)$ the "effective set" changes, then it is called a turning point. At $t = t^*$, the "effective set" and the sign of the coefficients in $\boldsymbol{\alpha}$ are updated and remain fixed until the next turning point is reached. Keep iterating this process until $t = 1$, and the final solution $\boldsymbol{\alpha}(t)$ will be obtained.

According to the "effective set" obtained in STEP1, we can divide $\boldsymbol{\alpha}^{(0)}$ into two sets: the set of non-zero coefficients and the set of zero coefficients. Let us denote them as $\boldsymbol{\alpha}^T = (\boldsymbol{\alpha}_1{}^T, 0^T)$, where $\boldsymbol{\alpha}_1{}^T$ corresponds to the non-zero coefficients and 0^T corresponds to the zero coefficients. We also define $\mathbf{v}^T = (\mathbf{v}_1{}^T, \mathbf{v}_2{}^T) \in \partial|\boldsymbol{\alpha}(0)|_1$, where $\partial|\boldsymbol{\alpha}(0)|_1$ is the subdifferential of the l_1-norm of $\boldsymbol{\alpha}(0)$, and $\mathbf{v}_1$ and $\mathbf{v}_2$ correspond to the subgradients of the non-zero and zero coefficients, respectively.

Lemma 1. *Assume that for all $1 \leq i \leq n$, $\alpha_{1i} \neq 0$, $|v_{2i}| < 1$, there exists $t^* > 0$, and for all $t \in [0, t^*)$, the "effective set" of Equation (6) and the sign of the internal coefficient of $\boldsymbol{\alpha}$ are the same as $\boldsymbol{\alpha}^{(0)}$.*

Proof. The optimal condition of Equation (6) is as follows:

$$\mathbf{Z}_{n-1}^T(\mathbf{Z}_{n-1}\boldsymbol{\alpha} - \mathbf{Y}_{n-1}) + t^2 \mathbf{z}_n(\mathbf{z}_n^T \boldsymbol{\alpha} - y_n) + \mu^{(n)} \boldsymbol{\omega} = 0 \tag{7}$$

Here $\boldsymbol{\omega} = \partial\|\boldsymbol{\alpha}\|_1$, if t is small enough, there exists a solution $\boldsymbol{\alpha}^T = (\boldsymbol{\alpha}_1(t)^T, 0^T)$ and $\boldsymbol{\omega}(t)^T = (v_1{}^T, \boldsymbol{\omega}_2(t)^T) \in \partial\|\boldsymbol{\alpha}(t)\|_1$, that satisfy the optimal condition. □

We can partition the data vector $\mathbf{z}_n$ into two parts based on the "effective-set", that is $\mathbf{z}_n = (\mathbf{z}_{n,1}^T, \mathbf{z}_{n,2}^T)$ and rewrite the optimality condition as follows.

$$\begin{cases} \mathbf{Z}_{n-1,1}^T(\mathbf{Z}_{n-1,1}\boldsymbol{\alpha}_1(t) - \mathbf{Y}_{n-1}) + t^2 \mathbf{z}_{n,1}(\mathbf{z}_{n,1}^T \boldsymbol{\alpha}_1(t) - y_n) + \mu^{(n)} \mathbf{v}_1 = 0 \\ \mathbf{Z}_{n-1,2}^T(\mathbf{Z}_{n-1,1}\boldsymbol{\alpha}_1(t) - \mathbf{Y}_{n-1}) + t^2 \mathbf{z}_{n,2}(\mathbf{z}_{n,1}^T \boldsymbol{\alpha}_1(t) - y_n) + \mu^{(n)} \boldsymbol{\omega}_2(t) = 0 \end{cases} \tag{8}$$

where $\boldsymbol{\alpha}_1(t)$ of the first equation,

$$\boldsymbol{\alpha}_1(t) = (\mathbf{Z}_{n-1,1}^T \mathbf{Z}_{n-1,1} + t^2 \mathbf{z}_{n,1} \mathbf{z}_{n,1}^T)^{-1} (\mathbf{Z}_{n-1,1}^T \mathbf{Y}_{n-1} + t^2 y_n \mathbf{z}_{n,1} - \mu \mathbf{v}_1) \tag{9}$$

It can be seen that $\boldsymbol{\alpha}_1(t)$ is a continuous function of t, because $\boldsymbol{\alpha}_1^{(0)} = \boldsymbol{\alpha}_1$, and all elements in $\boldsymbol{\alpha}_1$ are strictly greater than 0. Thus, for all $t < t_1^*$, there exists a t such that all elements within $\boldsymbol{alpha}_1(t)$ remain sign-invariant for the positive sign. Likewise, we can get:

$$-\mu\boldsymbol{\omega}_2(t) = \mathbf{Z}_{n-1,2}^T(\mathbf{Z}_{n-1,1}\boldsymbol{\alpha}_1(t) - \mathbf{Y}_{n-1}) + t^2\mathbf{z}_{n,2}(\mathbf{z}_{n,1}^T\boldsymbol{\alpha}_1(t) - y_n) \tag{10}$$

Similarly, $\boldsymbol{\omega}_2(t)$ is a continuous function of t. Because $\boldsymbol{\omega}_2(0) = \mathbf{v}_2$, there exists a t_2^* such that for all $t < t_2^*$, and the absolute values of all elements in $\boldsymbol{\omega}_2(t)$ are strictly less than 1. Thus we can obtain $t^* = min(t_1^*, t_2^*)$ to get the desired result. In summary, $\boldsymbol{\alpha}(t)$ is smooth until t reaches the turning point. There are two conditions for t to reach the turning point: one of the elements in $\boldsymbol{\alpha}_1(t)$ becomes 0, or one of the elements in $\boldsymbol{\omega}_2(t)$ reaches 1 in absolute value. Now we derive how to calculate the turning point.

Let $\tilde{\mathbf{Z}} = \begin{pmatrix} \mathbf{Z}_{n-1} \\ \mathbf{z}_n^T \end{pmatrix}$, $\tilde{\mathbf{Y}} = \begin{pmatrix} \mathbf{Y}_{n-1} \\ y_n \end{pmatrix}$, and divide $\tilde{\mathbf{Z}}$ into $\tilde{\mathbf{Z}} = (\tilde{\mathbf{Z}}_1, \tilde{\mathbf{Z}}_2)$ according to the "effective-set". Then rewrite Equation (10) using the Sherman Morrison formula [15]:

$$\boldsymbol{\alpha}_1(t) = \tilde{\boldsymbol{\alpha}}_1 - \frac{(t^2-1)\bar{e}}{1+\beta(t^2-1)}\mathbf{u} \tag{11}$$

Here, $\tilde{\boldsymbol{\alpha}}_1 = (\tilde{\mathbf{Z}}_1^T\tilde{\mathbf{Z}}_1)^{-1}(\tilde{\mathbf{Z}}_1^T\tilde{\mathbf{Y}} - \mu v_1)$, $\bar{e} = z_{n,1}^T\tilde{\boldsymbol{\alpha}}_1 - y_{\boldsymbol{n}}$, $\beta = \mathbf{z}_{n,1}^T(\tilde{\mathbf{Z}}_1^T\tilde{\mathbf{Z}}_1)^{-1}\mathbf{z}_{n,1}$ and $\mu = (\tilde{\mathbf{Z}}_1^T\tilde{\mathbf{Z}}_1)^{-1}\mathbf{z}_{n,1}$. Let the value of t be t_{1i} when $\alpha_{1i}(t) = 0$ one can get:

$$t_{1i} = (1 + (\frac{\bar{e}\mu_i}{\tilde{\alpha}_{1i}} - \beta)^{-1})^{\frac{1}{2}} \tag{12}$$

For the case that the absolute value of $\boldsymbol{\omega}_2(t)$ reaches 1 and we can obtain the following equation.

$$\begin{cases} \mathbf{z}_{n,1}^T\boldsymbol{\alpha}_1(t) - y_n = \frac{\bar{e}}{1+\beta(t^2-1)} \\ \tilde{\mathbf{Z}}_1\boldsymbol{\alpha}_1(t) - \tilde{\mathbf{Y}} = \tilde{e} - \frac{(t^2-1)\bar{e}}{1+\beta(t^2-1)}\tilde{\mathbf{Z}}_1\mathbf{u} \end{cases} \tag{13}$$

where $\tilde{e} = \tilde{\mathbf{Z}}_1\tilde{\boldsymbol{\alpha}}_1 - \mathbf{Y}$, rewrite Equation (10) as:

$$-\mu\boldsymbol{\omega}_2(t) = \tilde{\mathbf{Z}}_2^T\tilde{e} + \frac{(t^2-1)\bar{e}}{1+\beta(t^2-1)}(\mathbf{z}_{n,2} - \tilde{\mathbf{Z}}_2^T\tilde{\mathbf{Z}}_1\mathbf{u}) \tag{14}$$

Let c_i be the i-th column of $\tilde{\mathbf{Z}}_2$ and $z^{(i)}$ be the i-th element of $\mathbf{z}_{n,2}$. Only when the following equation holds,

$$\left| c_i^T\tilde{e} + \frac{(t^2-1)\bar{e}}{1+\beta(t^2-1)}(z^{(i)} - c_i^T\tilde{\mathbf{Z}}_1\mathbf{u}) \right| = \mu \tag{15}$$

the absolute value of the i-th element in $\boldsymbol{\omega}_2(t)$ will become 1.

Let $t_{2,j}^+$ (or $t_{2,j}^-$) be the value of t when $\boldsymbol{\omega}_{2j}(t) = 1$ (or $\boldsymbol{\omega}_{2j}(t) = -1$), the following can be obtained.

$$\begin{cases} t_{2,j}^+ = (1 + (\frac{\bar{e}(x^{(j)} - c_j^T\tilde{\mathbf{Z}}_1\mathbf{u})}{-\mathbf{u} - c_j^T\tilde{e}})^{-1})^{\frac{1}{2}} \\ t_{2,j}^- = (1 + (\frac{\bar{e}(x^{(j)} - c_j^T\tilde{\mathbf{Z}}_1\mathbf{u})}{\mathbf{u} - c_j^T\tilde{e}})^{-1})^{\frac{1}{2}} \end{cases} \tag{16}$$

So the turning point is $t' = min\{min_i t_{1i}, min_j t_{2,j}^+, min_j t_{2,j}^-\}$.

In summary, the update procedures of the recursive Lasso can be obtained as Algorithm 1.

Algorithm 1 : Lasso online update (adding observations).

STEP1: calculate $\boldsymbol{\alpha}^{(n-1)} = \boldsymbol{\alpha}(0, \mu_{n-1})$ to $\boldsymbol{\alpha}^{(n)} = \boldsymbol{\alpha}(1, \mu_n)$.

STEP2: Initialize the non-zero coefficients of $\boldsymbol{\alpha}(0, \mu_n)$ to the "effective set", let $\mathbf{v} = sgn(\boldsymbol{\alpha}(0, \mu_n))$. Let $\mathbf{v}_1$ and $z_{n,1}$ be the subvectors of $\mathbf{v}$ and z_n according to the "effective set", and $\tilde{\mathbf{Z}}_1$ be a submatrix of $\tilde{\mathbf{Z}}$ whose columns are the "effective set". initialization $\tilde{\boldsymbol{\alpha}}_1 = (\tilde{\mathbf{Z}}_1^T \tilde{\mathbf{Z}}_1)^{-1}(\tilde{\mathbf{Z}}_1^T \tilde{\mathbf{Y}} - \mu \mathbf{v}_1)$. Initialize turning point $t' = 0$.

STEP3: Calculate the next turning point t'. If the turning point is smaller than the previous turning point or if the turning point is greater than 1, skip to STEP5. If the i-th element in $\boldsymbol{\alpha}_1(t')$ becomes 0, choose case 1. Otherwise, if the absolute value of the j-th element in $\boldsymbol{\omega}_2(t')$ reaches 1, choose case 2.

CASE1: ① The i-th element in $\boldsymbol{\alpha}_1(t')$ becomes 0;
② remove i from the "active set";
③ set $\mathbf{v}_i$ to 0.

CASE2: ① The absolute value of the j-th element of $\boldsymbol{\omega}_2(t')$ reaches 1;
② Add j to the "valid set";
③ If the element reaches 1 (or -1), set $\mathbf{v}_j$ to 1 (or -1).

STEP4: Update $\mathbf{v}_1$, $\tilde{\mathbf{Z}}_1$ and $z_{n,1}$ according to the updated "effective set";
Update $\tilde{\boldsymbol{\alpha}}_1 = (\tilde{\mathbf{Z}}_1^T \tilde{\mathbf{Z}}_1)^{-1}(\tilde{\mathbf{Z}}_1^T \tilde{\mathbf{Y}} - \mu \mathbf{v}_1)$.

STEP5: Computes the final result when $t = 1$, where the value of $\boldsymbol{\alpha}^{(n)}$ is given by the active set of $\tilde{\boldsymbol{\alpha}}_1$.

2.3. The Sliding-Window Strategy for Anomaly Detection

The frequent mode switching of cigarettes production operating conditions and high sampling frequency of flowmeter requires the model to efficiently and accurately track the data. To achieve this, we employ a sliding-window strategy [16] that removes the oldest samples and includes the latest samples in the updating procedures. The flow chart of the sliding-window strategy is shown in Figure 2.

In Figure 2, assume the solution $\alpha^{(n)}$ at time n has been obtained, when a new data point is obtained, the oldest sample is discarded and the new observation can be included and the recursive Lasso method can be used to obtain the new model parameters. Assuming $\mathbf{z}_1 = (y_0, y_{-1}, \ldots, y_{-p})$, $\mathbf{Z} = (z_2, \ldots, z_n)^T$ $\mathbf{Y} = (y_2, y_3, \ldots, y_n)^T$, the objective function to be optimized is given by:

$$\boldsymbol{\alpha}(t, \mu) = \underset{\boldsymbol{\alpha}}{argmin} \frac{1}{2} \left\| \begin{pmatrix} ty_1 \\ Y \end{pmatrix} - \begin{pmatrix} tz_1 \\ Z \end{pmatrix} \boldsymbol{\alpha} \right\|_2^2 + \mu \|\boldsymbol{\alpha}\|_1 \tag{17}$$

The entire update path becomes $\boldsymbol{\alpha}^{(n)} = \boldsymbol{\alpha}(1, \mu_n)$ to $\boldsymbol{\alpha}^{(n')} = \boldsymbol{\alpha}(0, \mu_{n'})$. According to this update path, the update method can be divided into for two steps:

The update process of Equation (17) can be summarized as the following steps.

STEP1: $\hat{\boldsymbol{\alpha}}^{(n)}(1, \mu^{(n)}) \to \hat{\boldsymbol{\alpha}}'^{(n)}(1, \mu^{(n')})$ When $t = 1$, use the minimum angle regression algorithm [14] to update the penalty item parameter $\mu^{(n)}$ to $\mu^{(n')}$;

STEP2: $\hat{\boldsymbol{\alpha}}^{(n)}(1, \mu^{(n')}) \to \hat{\boldsymbol{\alpha}}^{(n)}(0, \mu^{(n)})$ When $\mu = \mu^{(n')}$, calculate t from 1 to 0.

Once the model update is finished, it is now possible to use the model the track the flowmeter observations. Since the flowmeter is checked regularly to ensure its accuracy, we can obtain a set of unevenly sampled check points (which are the actual values of the flowmeter). Denote the check points to $\bar{Y} = (\bar{y}_t, \bar{y}_{2t}, \ldots \bar{y}_{nt})$. Then the optimization problem can be rewritten as follows.

$$\boldsymbol{\alpha}(t,\mu) = \underset{\boldsymbol{\alpha}}{argmin}\frac{1}{2}\left\|\begin{pmatrix} ty_1 \\ Y \end{pmatrix} - \begin{pmatrix} tz_1 \\ Z \end{pmatrix}\boldsymbol{\alpha}\right\|_2^2 + \mu\|\boldsymbol{\alpha}\|_1 + \lambda\sum_{i=1}^{n}(y_i - \bar{y}_{it}) \tag{18}$$

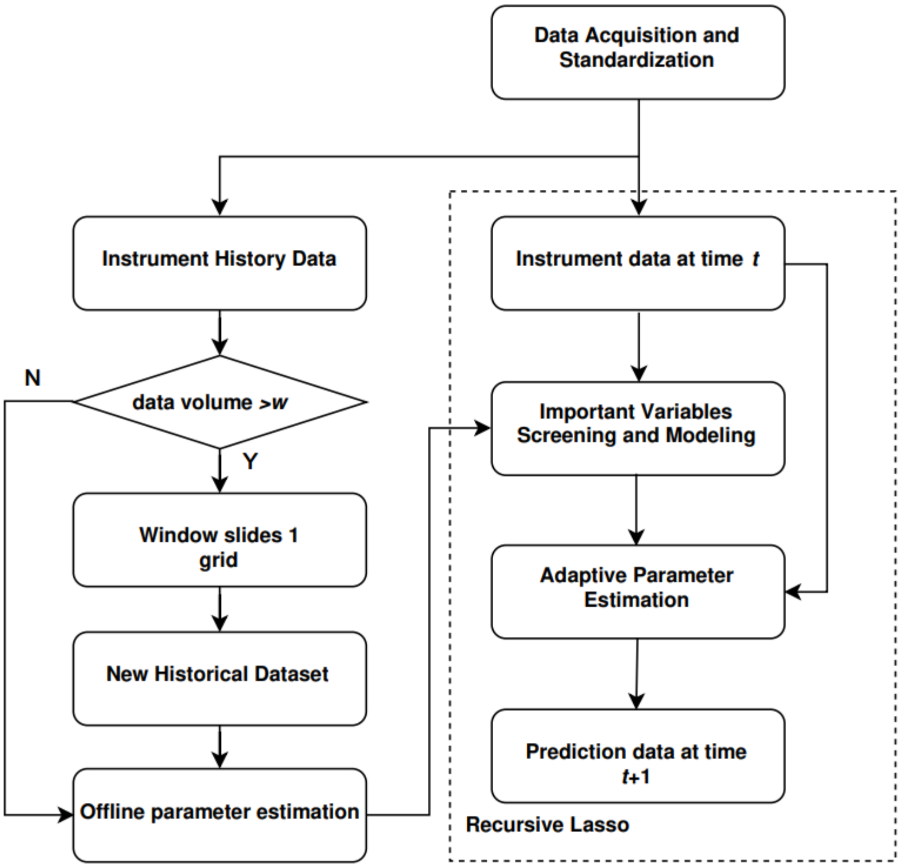


Figure 2. Flow chart of Recursive Lasso algorithm under sliding window.

In order to detect anomaly in the process, we consider the residual absolute value $\hat{\boldsymbol{\varepsilon}}$ between the predicted value $\hat{\boldsymbol{y}}$ and the flowmeter measurement $\bar{\boldsymbol{y}}$ to evaluate the status of the flowmeter.If the residual absolute value exceeds a certain threshold, it indicates that there might be an anomaly in the process, and the flowmeter needs to be checked. The threshold can be determined based on the statistical properties of the residual errors or by setting it empirically. Once an anomaly is detected, a notification can be sent to the operator or maintenance team to take action.

$$\hat{\boldsymbol{\varepsilon}} = |\boldsymbol{y} - \hat{\boldsymbol{y}}| \tag{19}$$

According to practical settings of the tobacco factory, if the residual $\hat{\boldsymbol{\varepsilon}}$ exceeds a certain threshold, then operational mode switch or flowmeter anomaly is detected. In addition, a higher threshold can be used to distinguish between switch in process mode and flowmeter anomaly. Once an anomaly in the flowmeter has occurred, the operators are then required for further inspection.

3. Application Results and Discussion

This section presents the application results of the proposed anomaly detection method on a dataset collected from a tobacco factory in central China. The data was collected at the cigarette preliminary processing stage, which processes the tobacco raw materials (strips and stems) into cut tobacco for rolling. It is mainly composed of three process sections, namely lamina pre-treatment section, cut lamina making section and casing and flavoring section, involving 9 working units and 11 flow control points.

Under actual working conditions, the working conditions of preliminary processing stage constantly changes according to different production tasks. Therefore, it is required to use a model that can adapt to change of working conditions. We apply the proposed sliding-window recursive Lasso method to detect anomalies in flowmeter data collected from a humidification unit. The dataset consists of 12,771 samples collected over a period of 30 h, during which the operational modes of the unit switched 4 times. An anomaly in the flowmeter occurred, resulting in a congestion of flows. The proposed method is applied to identify the anomaly and used root mean squared error (RMSE) and mean absolute error (MAE) [17] values as indicators to compared with Lasso [18] and Ridge [19,20]. Details of the dataset and experimental setup are provided in the methodology section.

In order to obtain an initial model for the recursive Lasso, it is necessary to train the model using a small part of data before online application. Therefore, a small set of 316 samples are used for initial model construction and the length of sliding window is set as 5. To show how the recursive Lasso behaves under stable conditions, a total of 1265 samples are considered and the prediction results are shown in Figure 3.

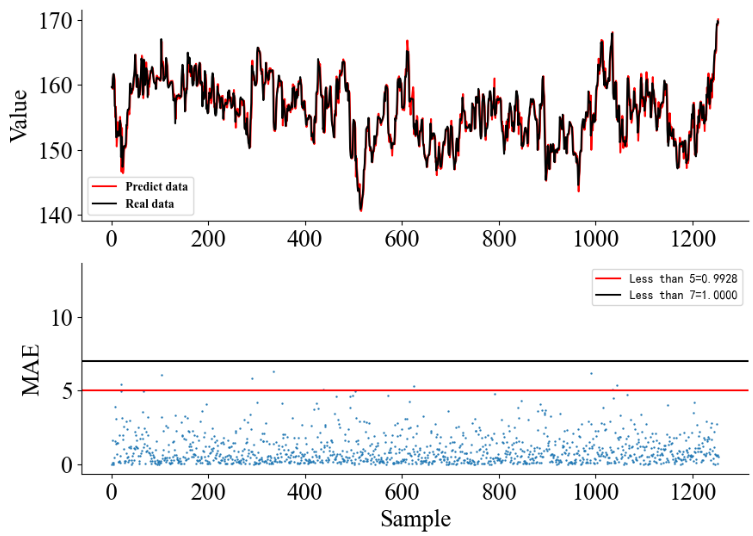


Figure 3. The prediction results of sliding-window recursive Lasso under steady flow state.

In Figure 3, it can be seen from the upper plot that the predicted flow values track the real flow values very well and the results are further verified by the absolute value of residuals in the lower plot. It can be seen that about 99.28% of the absolute prediction residuals are below 5kg/h, and 100% of the absolute prediction residuals are below 7 kg/h. This indicates that when the system is in a stable state, the tracking error of the sliding window is expected to be no more than 7 kg/h.

Next, the 12,771 samples under full working conditions from 2020/10/06 08:31:08 to 2020/10/07 14:21:18 are considered. First, an initial Lasso model is constructed using the first 2554 samples and the remaining 10,217 samples are used for online prediction. Figure 4 shows the prediction results of different models for the flow data at the conditioning point of the cut stem casing unit. Lasso, the ridge regression, and sliding-window recursive Lasso all use past values to predict current values. Therefore, when an abnormality occurs, the predictions will deviate from the actual values.From Figure 4 it is evident that the sliding-window recursive Lasso outperforms Lasso and the ridge regression in predicting the flow data. The predicted values of the sliding-window recursive Lasso model better follow the actual values compared to the other two methods, indicating its superior performance in capturing the underlying patterns and trends in the data. In contrast, Lasso and the ridge regression show a relatively larger deviation from the actual values, indicating their limited ability to capture the complex dynamics of the system.

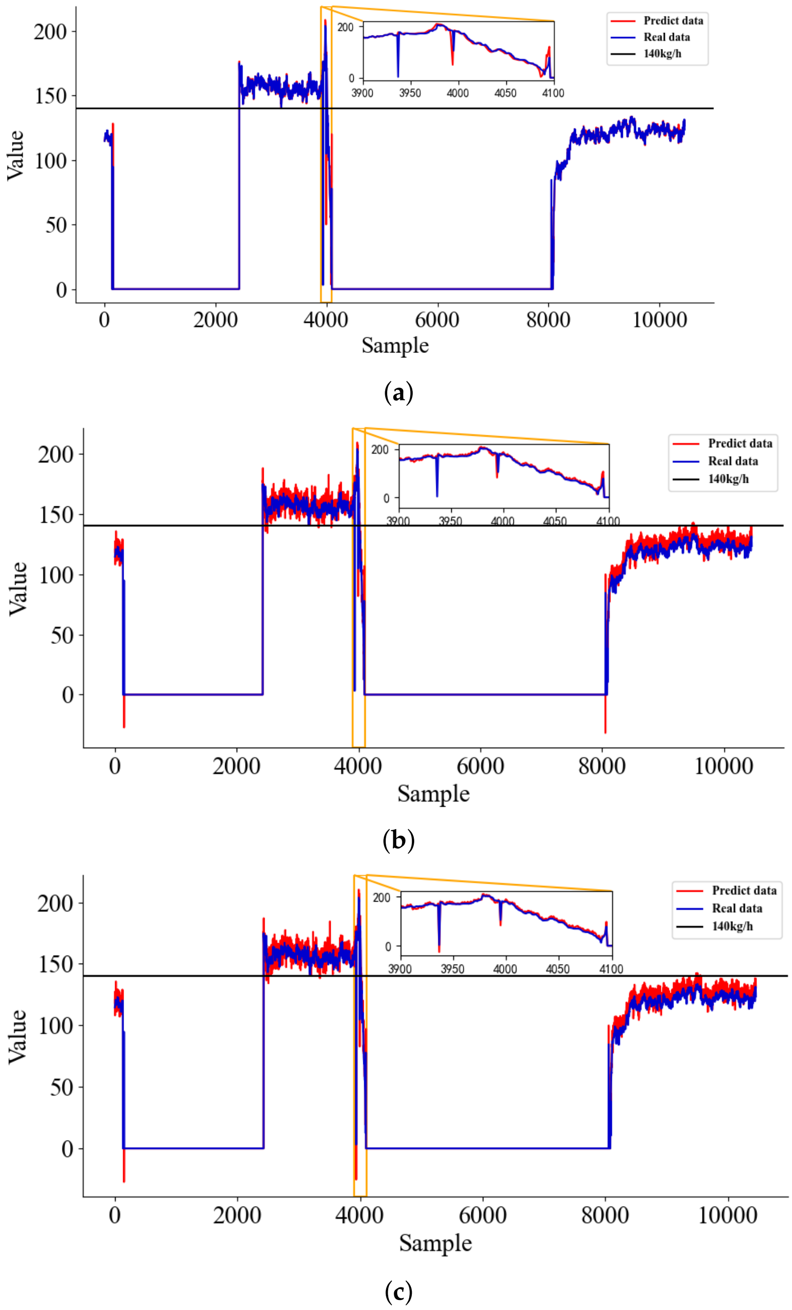
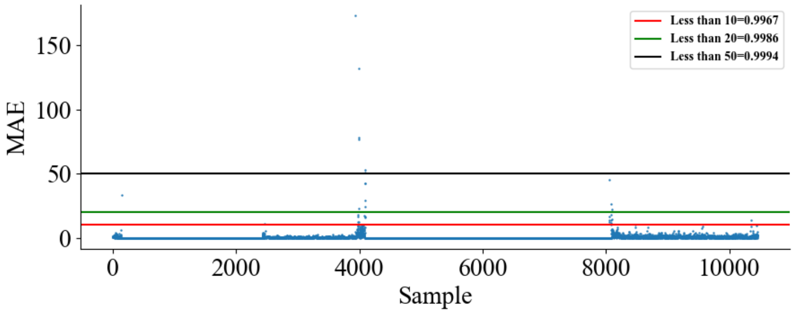
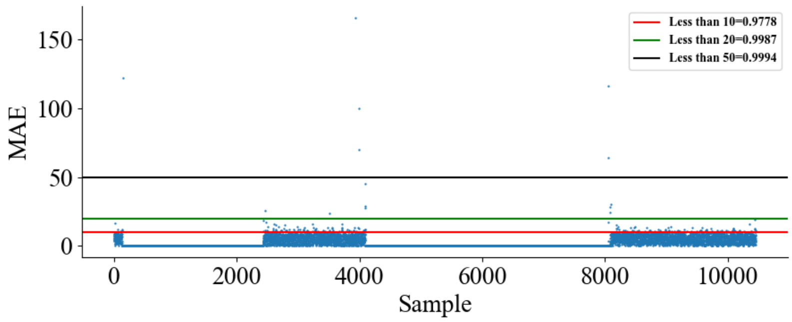


Figure 4. The prediction results under full working conditions (the red lines represent prediction, the blue lines represent actual values and the black line represents the flow indicator line of 140 kg/h for a kind of special tobacco stem products). (**a**) Prediction results of sliding-window recursive Lasso. (**b**) Prediction results of Lasso. (**c**) Prediction results of the ridge regression.

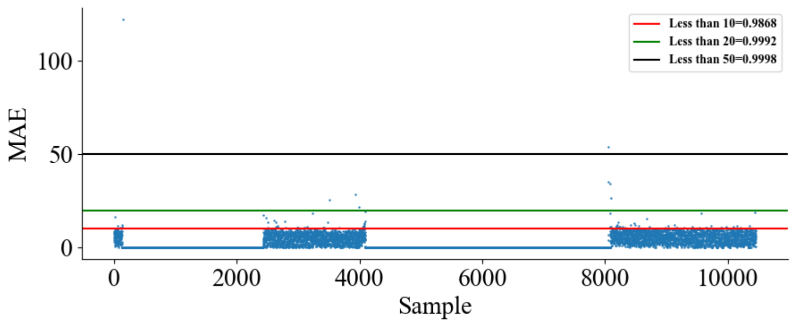
Based on the analysis of the absolute values of the tracking errors shown in Figure 5, the proposed method used three thresholds to distinguish between mode switches and flowmeter anomalies. The first threshold, indicated by the red line (10 kg/h), and the second threshold, indicated by the green line (20 kg/h), was used to identify abnormal conditions such as mode switches or flowmeter anomalies. It can be seen that all four mode switches exceed the 10 kg/h threshold, which is consistent with real-world practice. The third threshold, indicated by the black line (50 kg/h), was used to determine whether the abnormal conditions are flowmeter anomalies. As in the case of mode switches, the sliding-window recursive Lasso can adapt to the changes quickly, while in the case of flowmeter anomaly, it is difficult to track the real values in an accurate way. Hence exceeding of the third threshold almost surely indicates there is an anomaly in the flow measurements. This is in accordance with the practical situation. The sliding-window recursive Lasso achieved the highest detection accuracy with the lowest false alarm rate, followed by Lasso and the ridge regression, as shown in the results presented in Figure 5 The sliding-window recursive Lasso was able to accurately detect all four mode switches and the flowmeter anomaly that occurred in the 3900–4100 sampling points. While Lasso was able to identify both mode switches and the flowmeter anomaly, it could not distinguish between them and thus had a higher false alarm rate. On the other hand, the ridge regression only identified two of the four mode switches and mistakenly treated them as anomalies, resulting in the lowest detection accuracy with the highest false alarm rate.



(**a**)



(**b**)



(**c**)

Figure 5. Anomaly detection results under flowmeter anomaly(the red line represents the first threshold (10 kg/h), the green line represents second threshold (20 kg/h) and the black line represents the third threshold (50 kg/h). (**a**) Anomaly detection results of sliding-window recursive Lasso. (**b**) Anomaly detection results of Lasso. (**c**) Anomaly detection results of the ridge regression.

The Table 1 presents a comparison of the performance of three different methods for detecting anomalies in flowmeter data collected from a humidification unit. It shows the mean absolute error (MAE) and root mean squared error (RMSE) values for each method. The Ringe method had an MAE of 16.3249 and an RMSE of 4.0404, the Lasso method had an MAE of 22.3342 and an RMSE of 4.7259, while the sliding-window recursive Lasso method had an MAE of 8.1479 and an RMSE of 2.8544. Both MAE and RMSE were significantly decreased compared to the other two methods, indicating that the sliding-window recursive Lasso method outperformed the other two methods in detecting anomalies in the flowmeter data. These findings are consistent with the results shown in Figure 5, further supporting the effectiveness of the sliding-window recursive Lasso method.

Table 1. Comparison of detection accuracy between Lasso algorithm, sliding-window recursive Lasso algorithm and the ridge regression.

Method	MAE	RMSE
Ridge	16.3249	4.0404
Lasso	22.3342	4.7259
Sliding-window Recursive Lasso	8.1479	2.8544

4. Conclusions

In this paper, an anomaly detection method based on sliding window recursive Lasso is proposed for tobacco flowmeters. First, the initial regression coefficients are calculated using the traditional lasso regression method to 'warm up' the recursive algorithm. Secondly, when new data are obtained, a sliding window strategy is used to incorporate the effect of new data and eliminate that of outdated data. Then, the model parameters are updated in real-time using online recursion based on the newly sampled data, so that when the industrial process changes, the model can still be adjusted according to the changed sampling points, and the model prediction results are always reliable. Finally, the absolute difference between the predicted and measured values at the verification point can be used as the index for flowmeter anomaly detection and pattern recognition, allowing for efficient and accurate detection of anomalies in the flowmeter. The proposed method introduces a sliding window strategy that effectively eliminates the impact of outdated data. By adaptively updating the model parameters in real-time based on new sampled data, the model is well adapted to the changing situation of modern industrial processes. The proposed method introduces a sliding window strategy that effectively eliminates the impact of outdated data. By adaptively updating the model parameters in real-time based on new sampled data, the model is well adapted to the changing situation of modern industrial processes. This innovative approach offers a significant advantage over conventional methods such as Lasso and the ridge regression. Experimental results on real-world data from a tobacco factory demonstrate the effectiveness of the proposed method, achieving remarkable results with a mean absolute error (MAE) of 8.1479 and a root mean square error (RMSE) of 2.8544. The combination of the sliding window strategy and online recursion ensures that the model prediction results are always reliable and responsive to changes in the industrial process, making the proposed method highly suitable for anomaly detection in flowmeters.

Author Contributions: Conceptualization, Z.G., Y.L. (Ying Liu) and J.C.; methodology, Z.G. and J.C.; validation, L.Y. and J.L.; formal analysis, L.Y and J.L; investigation, S.L., B.L. and Y.L. (Yuhao Liu); resources, S.L.; Data curation, B.L.; writing—original draft, Z.G.; writing—review & editing, Y.L. (Ying Liu) and J.C.; supervision, T.C. and J.C.; project administration, Y.L. (Yuhao Liu) and T.C.; funding acquisition, T.C. and Y.L. (Yuhao Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Technology Projects of China Tobacco He'nan Industrial Co., Ltd (A202053 and RH202005).

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

References

1. Pike, P.; Pennycook, K. *Commissioning of BEMS: Code of Practice*; Building Services Research and Information Association: Bracknell, UK, 1992.
2. Sun, Y.; Liu, X.; Chen, X.; Sun, Q.; Zhao, J. Research and application of a fault self-diagnosis method for roots flowmeter based on wsn node. *Wirel. Pers. Commun.* **2017**, *95*, 2315–2330. [CrossRef]
3. Ran, Z.; Liang, W. Research on diagnosis of faulty working turbine gas flowmeter. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 3588–3592.
4. Wang, J.; Tavoosi, J.; Mohammadzadeh, A.; Mobayen, S.; Asad, J.H.; Assawinchaichote, W.; Vu, M.T.; Skruch, P. Non-singleton type-3 fuzzy approach for flowmeter fault detection: experimental study in a gas industry. *Sensors* **2021**, *21*, 7419. [CrossRef] [PubMed]
5. Sun, Z.; Zhang, H. Application of empirical mode decomposition based energy ratio to vortex flowmeter state diagnosis. *J. Cent. South Univ. Technol.* **2009**, *16*, 154–159. [CrossRef]
6. Tang, X.; Xie, X.; Fan, B.; Sun, Y. A fault-tolerant flow measuring method based on pso-svm with transit-time multipath ultrasonic gas flowmeters. *IEEE Trans. Instrum. Meas.* **2018**, *67*, 992–1005. [CrossRef]
7. Zhang, H.; Zhang, X.; Zhang, S. Investigation on flow instrument fast fault detection based on LSSVM predictor. In Proceedings of the 2010 Sixth International Conference on Natural Computation, Yantai, China, 10–12 August 2010; Volume 3, pp. 1271–1275.
8. Chen, W.; Zeng, J.; Li, Y.; Luo, S. Improved pca-based fault isolation using sparse group lasso. In Proceedings of the 2020 IEEE 9th Data Driven Control and Learning Systems Conference (DDCLS), Liuzhou, China, 20–22 November 2020; pp. 156–161.
9. Cascianelli, S.; Crocetti, F.; Costante, G.; Valigi, P.; Fravolini, M.L. Data-based design of robust fault isolation residuals using lasso optimization. In Proceedings of the 2019 International Conference on Control, Automation and Diagnosis (ICCAD), Grenoble, France, 2–4 July 2019; pp. 1–6.
10. Chen, Y.; Hao, X.; Zhang, G.; Wang, S. Flow meter fault isolation in building central chilling systems using wavelet analysis. *Energy Convers. Manag.* **2006**, *47*, 1700–1710. [CrossRef]
11. Yang, R.; Satou, Y.; Tateno, S.; Matsuyama, H. Fault diagnosis of flow meters utilizing historical data. In Proceedings of the 2006 SICE-ICASE International Joint Conference, Busan, Republic of Korea, 18–21 October 2006; pp. 4703–4708.
12. Garrigues, P.; Ghaoui, L. An homotopy algorithm for the lasso with online observations. In Proceedings of the 21st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 8–10 December 2008; Advances in neural information processing systems; Volume 21.
13. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Methodol.* **1996**, *58*, 267–288. [CrossRef]
14. Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R. Least angle regression. *Ann. Stat.* **2004**, *32*, 407–499. [CrossRef]
15. Sherman, J.; Morrison, W.J. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Stat.* **1950**, *21*, 124–127. [CrossRef]
16. Saeed, H.A.; Wang, H.; Peng, M.; Hussain, A.; Nawaz, A. Online fault monitoring based on deep neural network & sliding window technique. *Prog. Nucl. Energy* **2020**, *121*, 103236.
17. Jadon, A.; Patil, A.; Jadon, S. A comprehensive survey of regression based loss functions for time series forecasting. *arXiv* **2022**, arXiv:2211.02989.
18. Chen, S.; Peng, M.; Xiong, H.; Wu, S. An anomaly detection method based on lasso. *Clust. Comput.* **2019**, *22*, 5407–5419. [CrossRef]
19. Hoerl, A.E.; Kennard, R.W. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics* **1970**, *12*, 55–67. [CrossRef]
20. Galal, M.A.; Hussein, W.M.; Abdelkawy, E.E. Satellite battery sensor values prediction using bayesian ridge regression models. *IOP Conf. Ser. Mater. Sci. Eng.* **2019**, *610*, 012012. [CrossRef]