

Article

# Data-Driven-Based Intelligent Alarm Method of Ultra-Supercritical Thermal Power Units

Xingfan Zhang, Lanhui Ye, Cheng Zhang and Chun Wei \*

College of Information Engineering, Zhejiang University of Technology, Hangzhou 310014, China; 2112103148@zjut.edu.cn (X.Z.); 221122030410@zjut.edu.cn (L.Y.); 211123030146@zjut.edu.cn (C.Z.)

\* Correspondence: chunwei18@zjut.edu.cn

**Abstract:** In order to ensure the safe operation of the ultra-supercritical thermal power units (USCT-PU), this paper proposes an intelligent alarm method to enhance the performance of the alarm system. Firstly, addressing the issues of slow response and high missed alarm rate (MAR) in traditional alarm systems, a threshold optimization method is proposed by integrating kernel density estimation (KDE) and convolution optimization algorithm (COA). Based on the traditional approach, the expected detection delay (EDD) indicator is introduced to better evaluate the response speed of the alarm system. By considering the false alarm rate (FAR), and EDD, a threshold optimization objective function is constructed, and the COA is employed to obtain the optimal alarm threshold. Secondly, to address the problem of excessive nuisance alarms, this paper reduces the number of nuisance alarms by introducing an adaptive delay factor into the existing system. Finally, simulation results demonstrate that the proposed method significantly reduces the MAR and EDD, improves the response speed and performance of the alarm system, and effectively reduces the number of nuisance alarms, thereby enhancing the quality of the alarms.

**Keywords:** USCTPUs; intelligent alarm; false alarm rate; missed alarm rate; convolution optimization



**Citation:** Zhang, X.; Ye, L.; Zhang, C.; Wei, C. Data-Driven-Based Intelligent Alarm Method of Ultra-Supercritical Thermal Power Units. *Processes* **2024**, *12*, 889. <https://doi.org/10.3390/pr12050889>

Academic Editors: Michael C. Georgiadis, Gongzhuang Peng and Shenglong Jiang

Received: 15 March 2024

Revised: 21 April 2024

Accepted: 25 April 2024

Published: 28 April 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Compared to traditional thermal power units, ultra-supercritical thermal power units (USCTPUs) exhibit higher thermal conversion efficiency and lower coal consumption, which can significantly reduce the pollution emissions of power plants, thereby enhancing the energy-saving and environmental protection effects of thermal power plants [1]. However, due to the complex system structure and increased coupling among equipment, USCTPUs are prone to experiencing abnormal operating conditions. The occurrence of malfunctions during operation not only reduces the production efficiency and economic benefits of the power plant, but can also lead to serious safety accidents [2]. In order to mitigate the adverse effects of malfunctions and prevent unit shutdown or equipment damage caused by malfunctions, it is necessary to utilize an alarm system to monitor critical equipment and parameters.

In the current industrial power generation field, alarm systems are widely used due to their critical role. However, the alarm system of USCTPUs faces issues such as slow system response, missed alarms, and excessive nuisance alarms. The main cause of slow system response and increased missed alarms is the improper setting of alarm thresholds. When the alarm threshold set by the system is large, it delays the generation of alarm signals, resulting in a slow system response and an increase in missed alarms, thus increasing the risk of equipment damage due to abnormal operating conditions. The excessive number of nuisance alarms is primarily due to the lack of a delay factor in traditional alarm systems. Due to the complexity of USCTPUs, there are strong correlations among alarm variables. The repeated occurrence of alarm signals for a specific monitoring parameter may be influenced by other parameters and does not necessarily indicate a fault. This leads to a

significant number of nuisance alarms in the alarm information, which severely interferes with the judgment of operators [3,4]. Therefore, an efficient and intelligent alarm system is crucial for ensuring the secure and steady running of USCTPUs.

Currently, the commonly used method for optimizing alarm systems is to optimize and design alarm thresholds. The optimization and design methods for alarm thresholds can be categorized into model-based [5], experience-based [6], and data-driven approaches [7,8]. Reference [9] proposes a method for designing alarm thresholds applicable to condensate water pumps in thermal power plants and constructs a model estimator based on Bayesian filtering to maximize the absolute cumulative error of model parameters, thus obtaining the alarm thresholds, and the effectiveness of the proposed method is verified through experiments. In [10], expert knowledge is combined with data featuring reliability characteristics to build an optimized model for alarm threshold based on interval evidence inference, and the effectiveness of this method is demonstrated through experimentation. Unlike the limitations of the above modeling and knowledge-based approaches, data-driven methods do not require the establishment of complex mechanistic models or extensive empirical knowledge; they only necessitate the analysis of alarm data, making them simple and efficient, and widely applied in industrial fields. In [11], a multivariable alarm threshold optimization method was proposed to address the issue of high false alarm rate (*FAR*) resulting from unreasonable alarm threshold settings, and simulation results demonstrate that this method effectively reduces *FAR*. Reference [12] introduces a variable alarm threshold optimization method based on correlation and cluster analysis, establishing an objective function concerning *FAR* and missed alarm rate (*MAR*) with variable weight coefficients, and employs a numerical optimization algorithm (NOA) to solve the objective function. Reference [13] uses standardized Euclidean distance to cluster and group the variables, calculates the variable weights by entropy weighting method, establishes the threshold optimization objective function with variable weights, and finally uses ant colony optimization algorithm to solve the optimal alarm threshold, and the experimental results show that this method can effectively reduce the *FAR*. Reference [14] initially groups parameter variables, calculates original weights, establishes an optimization objective function concerning *FAR*, *MAR*, and original weights, and then utilizes particle swarm optimization algorithm (PSO) to solve the objective function; experimental results indicate that the proposed method reduces the number of alarms by 37.8% compared to the initial situation. Reference [15] proposes a dynamic variable alarm threshold optimization algorithm suitable for non-stationary processes under different operating conditions, significantly reducing *FAR* and ensuring the reliability of the alarm system. Reference [16] presents a process alarm threshold optimization method based on satisfaction optimization, constructing a weighted objective function for *FAR* and *MAR*, and solving the optimal threshold through NOA, further enhancing the alarm system performance. Reference [17] utilizes non-parametric statistical methods based on windowing and testing to adjust the Hotelling  $T^2$  statistic and Q statistic, thereby generating second-level control limits for alarm thresholds, and the experimental results demonstrate that using these two methods can effectively reduce the *FAR*. Reference [18] addresses the lack of the ability in traditional alarm threshold optimization methods to adjust thresholds online, offering an adaptive optimization method for simplifying multi-layer Bayesian network alarm thresholds based on active transfer entropy, achieving a reduction in the sum of the average *FAR* from 28.5% to 19.7%. The aforementioned studies have further optimized alarm thresholds, to some extent reducing *FAR*; however, they overlook the issue that simply reducing *FAR* may lead to an increase in *MAR*, posing significant safety risks to production. Additionally, they neglect the problem of slow response in the alarm system. Furthermore, due to the absence of delay factors in traditional alarm systems, there is an issue of excessive nuisance alarms, preventing substantial improvement in the performance of alarm systems.

To address the issues of slow response speed and high *MAR* in traditional alarm systems, an optimization method for alarm thresholds is proposed in this paper. Firstly, the data under both normal and abnormal operating conditions were fitted into probabil-

ity density functions (PDF) using the kernel density estimation (KDE) approach. Then, based on the traditional threshold optimization methods, the detection delay expectation (*EDD*) to measure the agility of the response to the alarm system is considered, and the threshold optimization objective function is established with the *FAR*, *MAR*, and *EDD* as the optimization evaluation indexes. The convolution optimization algorithm (COA) is used to solve the problem, and the optimal alarm threshold is obtained. Furthermore, to address the issue of a high number of nuisance alarms, an adaptive alarm delay method is proposed. Based on the analysis of the alarm time interval data, an adaptive delay factor is introduced into the existing system to eliminate nuisance alarms. Finally, the effectiveness of the proposed methods is validated using historical operational data from 1000 MW and 660 MW USCTPU. The results demonstrate that the proposed method significantly reduces the *MAR* and *EDD*, improves the response speed and performance of the alarm system, effectively reduces the number of nuisance alarms, and enhances the quality of alarms.

The rest of the paper is organized as follows: Section 2 briefly introduces the relevant theoretical knowledge of KDE and COA. Section 3 presents the proposed alarm threshold optimization method and adaptive alarm delay method in detail. Section 4 applies the alarm threshold optimization method to 1000 MW and 660 MW USCTPU for case analysis and conducts an instance analysis of the adaptive alarm delay method to verify the performance of the proposed methods. Section 5 provides a summary of the entire paper.

## 2. Fundamental Theories

### 2.1. KDE

KDE is developed based on the statistical histogram, which differs from the traditional histogram estimation which yields a continuous PDF. In practical applications, KDE can estimate unknown variables by performing weighted averaging over fixed windows of data, thereby revealing the probability density distribution pattern. The estimated density function minimizes the integrated mean square error between the estimated and true density functions of the variables. Additionally, KDE allows for the study of distribution characteristics based on the data, making it suitable for data with arbitrary distribution properties. Therefore, selecting KDE to compute the PDF of the data in this paper facilitates the calculation of probability expressions for performance metrics [19]. It can be expressed as follows:

$$\hat{H}(x) = \frac{1}{mu} \sum_{p=1}^m K\left(\frac{x - x_p}{u}\right) \quad (1)$$

where  $x$  is the estimation point,  $x_p$  is the monitoring data point,  $m$  is the data length,  $K(\cdot)$  represents the kernel function,  $u$  is the width of the window, and  $\hat{H}(x)$  is the estimated value of the probability density function of the current monitoring data point. In KDE, the kernel function and the window width collectively decide on the precision of KDE. In this paper, we choose the Gaussian kernel function, and the window width  $u$  is obtained by minimizing the mean integrated square error (*MISE*) [20]. The definition of *MISE* is as follows:

$$\begin{aligned} MISE[\hat{H}(x)] &= E\left(\int [\hat{H}(x) - H(x)]^2 dx\right) \\ &= (mu)^{-1} \left[ \int K^2(s) ds \right] + \frac{\left[ \int s^2 K(s) ds \right]^2}{4} \cdot \int \left[ \frac{d^2}{dx^2} H(x) \right]^2 dx \cdot u^4 + o\left(\frac{1}{mu} + u^4\right) \end{aligned} \quad (2)$$

where  $H(x)$  is the real worth of the probability density function at the monitoring point, and  $o(\cdot)$  denotes a higher-order infinitesimal term. After neglecting the infinitesimal term in (2), the asymptotic mean integrated square error (*AMISE*) is obtained [21], and it can be expressed as follows:

$$AMISE[\hat{H}(x)] = (mu)^{-1} \left[ \int K^2(s) ds \right] + \frac{\left[ \int s^2 K(s) ds \right]^2}{4} \cdot \int \left[ \frac{d^2}{dx^2} H(x) \right]^2 dx \cdot u^4 \quad (3)$$

When the derivative of *AMISE* is equal to 0, its value reaches the minimum. The optimal window width  $u_b$  is given by the following [22]:

$$u_b = \sqrt[5]{\frac{\int K^2(s)ds}{m[\int s^2K(s)ds]^2 \cdot \int [\frac{d^2}{dx}H(x)]^2 dx}} \quad (4)$$

Finally, employing the Silverman's rule of thumb [23] to simplify the calculation of (4), the  $u_b$  can be expressed as follows:

$$u_b = S \times \left(\frac{4}{3} \times \frac{1}{m}\right)^{\frac{1}{5}} \quad (5)$$

where  $S$  is the standard deviation of the data.

## 2.2. COA

### 2.2.1. Convolutional Search Process

Step 1: Update the longitudinal convolution location. Firstly, define a longitudinal convolution as follows [24]:

$$X_1^t = X^t * [2 \times \text{rand}(k, 1) - I_a] \quad (6)$$

where  $\text{rand}(k,1)$  is a  $k \times 1$  matrix whose elements are random numbers between 0 and 1,  $k$  is the size of the convolution kernel,  $I_a$  is a  $k \times 1$  matrix whose elements are all one,  $t$  is the current iteration number,  $*$  denotes convolution operation,  $X^t$  is the location vector of the population in generation  $t$ , and  $X_1^t$  is the location vector of the population in generation  $t$  after updating the longitudinal convolution location.

Then, compare the *fitness* values of  $X_t$  and  $X_1^t$ , and optimally replace the larger individual location in  $X_t$  as shown below:

$$X_b^t = \begin{cases} X_{1b}^t, & \text{fitness}_{X_{1b}^t} < \text{fitness}_{X_b^t} \\ X_b^t, & \text{fitness}_{X_{1b}^t} \geq \text{fitness}_{X_b^t} \end{cases} \quad (7)$$

where  $X_b^t$  is the location of the  $b$ -th individual,  $X_{1b}^t$  is the location of the  $b$ -th individual after updating the longitudinal convolution location, and *fitness* is an adaptation function.

Step 2: Update the horizontal convolution location. Firstly, define a horizontal convolution as follows:

$$X_2^t = X^t * [2 \times \text{rand}(1, k) - I_c] \quad (8)$$

where  $\text{rand}(1,k)$  is a  $1 \times k$  matrix whose elements are random numbers between 0 and 1,  $I_c$  is a  $1 \times k$  matrix whose elements are all one, and  $X_2^t$  is the location vector of the population in generation  $t$  after updating the horizontal convolution location.

$$X_b^t = \begin{cases} X_{2b}^t, & \text{fitness}_{X_{2b}^t} < \text{fitness}_{X_b^t} \\ X_b^t, & \text{fitness}_{X_{2b}^t} \geq \text{fitness}_{X_b^t} \end{cases} \quad (9)$$

where  $X_{2b}^t$  is the  $b$ -th location of the individual after updating the horizontal convolution location.

Step 3: update the matrix convolution location. Firstly, define a matrix convolution as follows:

$$X_3^t = X^t * [2 \times \text{rand}(k, k) - I_n] \quad (10)$$

where  $\text{rand}(k,k)$  is a  $k \times k$  matrix whose elements are random numbers between 0 and 1,  $I_n$  is a  $k \times k$  matrix whose elements are all one, and  $X_3^t$  is the location vector for the population of the  $t$ -th generation after updating the matrix convolution location.

Then, compare the *fitness* values of  $X^t$  and  $X_{3b}^t$ , and optimally replace the larger individual location in  $X^t$  as shown below:

$$X_b^t = \begin{cases} X_{3b}^t, & \text{fitness}_{X_{3b}^t} < \text{fitness}_{X_b^t} \\ X_b^t, & \text{fitness}_{X_{3b}^t} \geq \text{fitness}_{X_b^t} \end{cases} \quad (11)$$

where  $X_{3b}^t$  is the location of the  $b$ -th individual after updating the matrix convolution location.

Step 4: Update the integrated location. Add  $X_1^t$ ,  $X_2^t$ , and  $X_3^t$  together according to proportional weights, as shown below:

$$X_4^t = \frac{e_1 \times X_1^t + e_2 \times X_2^t + e_3 \times X_3^t}{e_1 + e_2 + e_3} \quad (12)$$

where  $X_4^t$  is the location vector for the population of the  $t$ -th generation after updating the integrated location, and  $e_1$ ,  $e_2$ , and  $e_3$  are random numbers between 0 and 1.

Then, compare the *fitness* values of  $X^t$  and  $X_{4b}^t$ , and optimally replace the larger individual location in  $X^t$  as shown below.

$$X_b^t = \begin{cases} X_{4b}^t, & \text{fitness}_{X_{4b}^t} < \text{fitness}_{X_b^t} \\ X_b^t, & \text{fitness}_{X_{4b}^t} \geq \text{fitness}_{X_b^t} \end{cases} \quad (13)$$

where  $X_{4b}^t$  is the location of the  $b$ -th individual after updating the integrated location.

### 2.2.2. Result Enhancement

Select the best solution  $X_{be}^t$  from  $X^t$ , and apply Gaussian mutation with non-inertial weight to  $X_{be}^t$  in order to avoid falling into local optima during the iteration, as shown below:

$$\hat{X}_{be}^t = \left[ 1 - \left( \frac{t}{t_{\max}} \right)^2 \right] \cdot X^t + \text{randn} \cdot X_{be}^t \quad (14)$$

where  $\hat{X}_{be}^t$  is the best solution after enhanced perturbation,  $t_{\max}$  is the maximum number of iterations, and *randn* is a random number. Finally, compare  $\hat{X}_{be}^t$  and  $X_{be}^t$ , and replace the individual location of  $X_{be}^t$  on a merit basis:

$$X_{be}^t = \begin{cases} \hat{X}_{be}^t, & \text{fitness}_{\hat{X}_{be}^t} < \text{fitness}_{X_{be}^t} \\ X_{be}^t, & \text{fitness}_{\hat{X}_{be}^t} \geq \text{fitness}_{X_{be}^t} \end{cases} \quad (15)$$

## 3. Proposed Intelligent Alarm Method

### 3.1. Calculation of Performance Optimization Metric

For a univariate alarm system, one of the most critical performance metrics is accuracy, which can be evaluated using metrics such as *FAR* and *MAR* [25]. *FAR* refers to the probability of the system issuing an alarm signal when the process variable is operating normally. A high *FAR* can increase the number of alarms, significantly impacting the judgment and efficiency of the operators. *MAR* denotes the probability that the alarm system fails to respond with an alarm when the process variable experiences an anomaly. A high *MAR* can increase the potential risk of equipment failure, posing serious safety hazards to production. Therefore, the index value of *FAR* and *MAR* can reflect the performance of the designed alarm system. In alarm systems, *FAR* and *MAR* are inversely related, and this contradictory relationship is reflected in the receiver operating characteristic (ROC) curve, as shown in Figure 1. Assuming a process variable  $x(t)$  with known PDF for normal and abnormal operating conditions, PDF curve distributions for normal and fault data can be obtained, as shown in Figure 2. Here, the solid line represents the PDF curve distribution of normal operating condition data  $\hat{H}_1(x)$ , while the dashed line represents the PDF curve distribution of abnormal operating condition data  $\hat{H}_2(x)$ .

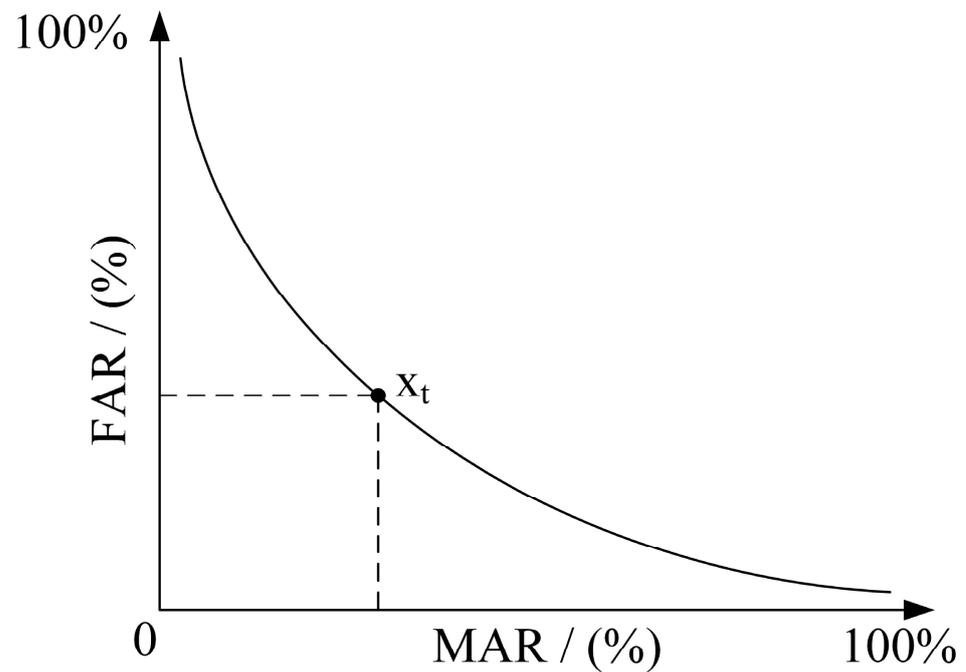


Figure 1. ROC curve.

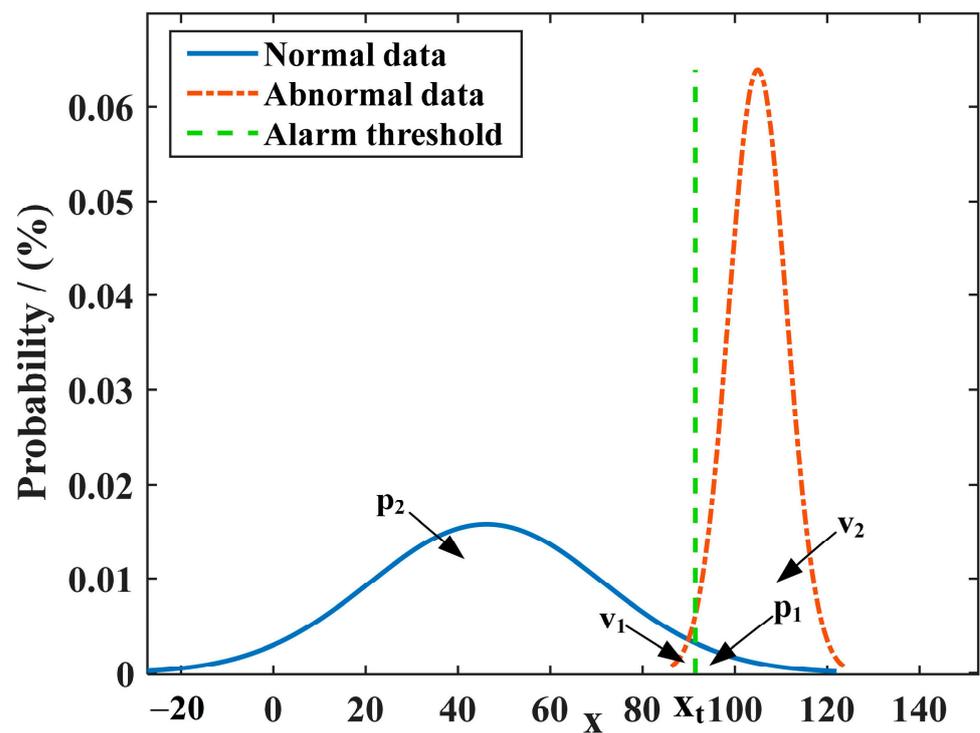


Figure 2. Distribution of PDF of process variables.

According to Figure 2, we can calculate  $p_1$ ,  $p_2$ ,  $v_1$ , and  $v_2$ , where  $p_1 + p_2 = v_1 + v_2 = 1$ . It can be expressed as follows:

$$\begin{cases} p_1 = \int_{X_t}^{+\infty} \hat{H}_1(x) dx \\ p_2 = \int_{-\infty}^{X_t} \hat{H}_1(x) dx \end{cases} \quad (16)$$

$$\begin{cases} v_1 = \int_{-\infty}^{x_t} \hat{H}_2(x) dx \\ v_2 = \int_{x_t}^{+\infty} \hat{H}_2(x) dx \end{cases} \quad (17)$$

Based on the definitions of *FAR* and *MAR* [25], the *FAR* and *MAR* can be expressed as follows:

$$FAR = p_1 = \int_{x_t}^{+\infty} \hat{H}_1(x) dx \quad (18)$$

$$MAR = v_1 = \int_{-\infty}^{x_t} \hat{H}_2(x) dx \quad (19)$$

Apart from accuracy, agility is another important performance metric for alarm systems. Agility is a measure of the response speed of the alarm system, typically characterized by *EDD*. Detection delay refers to the time difference between the occurrence of an anomaly and the triggering of the alarm state by the alarm system. A smaller value indicates a higher sensitivity and faster response speed of the alarm system. Therefore, minimizing the *EDD* as much as possible is essential to ensure the stable and reliable operation of USCTPUs. Assuming that the moment of failure occurrence is  $t_s$  during operation, and an alarm signal is sent out after  $h$  moments, with the signal being sent at time  $t_a$ , the detection delay at  $h$  moments can be expressed as follows:

$$\begin{cases} p(DD = 0) = p\{t_{al} = t_s\} = v_2 \\ p(DD = 1) = p\{t_{unal} = t_s\} \cdot p\{t_{al} = t_{s+1}\} = v_1 v_2 \\ \vdots \\ p(DD = h) = p\{t_{unal} = t_s\} \dots p\{t_{unal} = t_{s+h-1}\} \cdot p\{t_{al} = t_{s+h}\} = v_2 v_1^h \end{cases} \quad (20)$$

where  $t_{al}$  is the time when an alarm signal is issued, and  $t_{unal}$  is the time when no alarm signal is issued.

Next, find the *EDD* [25], calculated as follows:

$$E(DD) = \sum_{h=0}^{\infty} h \cdot p(DD = h) = \frac{v_1}{v_2} = \frac{MAR}{1 - MAR} \quad (21)$$

### 3.2. Design of Alarm Threshold Optimization Method

In order to find the optimal alarm threshold, the proposed threshold optimization method in this paper takes into account the *FAR*, *MAR*, and *EDD*. With the *FAR*, *MAR*, and *EDD* as optimization indicators, an alarm threshold optimization objective function is established, and the objective is to minimize the objective function using the COA for optimization calculation, thereby obtaining the optimal alarm threshold  $x_c$ . The flowchart of the threshold optimization algorithm is shown in Figure 3, the pseudocode is shown in Algorithm 1, and the objective function can be represented as follows:

$$F(x) = o_1 \frac{FAR}{FAR^*} + o_2 \frac{MAR}{MAR^*} + o_3 \frac{EDD}{EDD^*} \quad (22)$$

where  $o_1$ ,  $o_2$ , and  $o_3$  are the design weighting coefficients, and  $FAR^*$ ,  $MAR^*$ , and  $EDD^*$  are the maximum acceptable *FAR*, *MAR*, and *EDD*.

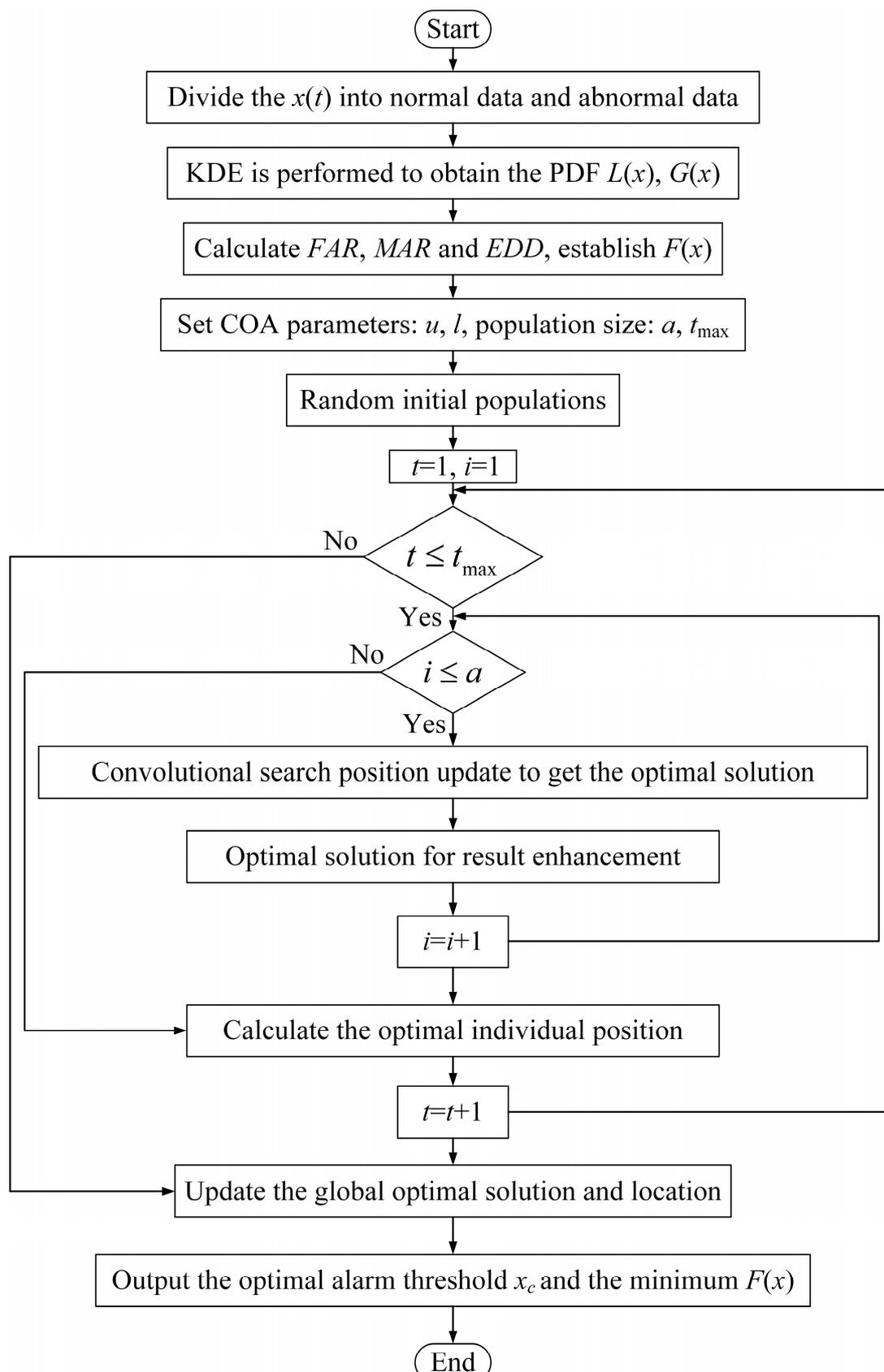


Figure 3. Flow chart of alarm threshold optimization.

Algorithm 1: The pseudocode of the alarm threshold optimization algorithm.

**Algorithm 1: Threshold optimization algorithm**


---

Input: Process variable  $x(t)$   
Parameter:  $x_t, dim, u_b, l_b, t_{max}, a$   
Output:  $FAR_{be}, MAR_{be}, EDD_{be}, x_c$

- 1: normal data = find  $x(t) \leq x_t$ , abnormal data = find  $x(t) > x_t$
- 2: normal PDF = PDF(normal data), abnormal PDF = PDF(abnormal data),
- 3: Function  $F(x) = \text{fun}(x)$
- 4: define  $o_1, o_2, o_3, FAR^*, MAR^*$ , and  $EDD^*$
- 5: compute  $FAR, MAR, EDD$
- 6: construct  $F(x)$
- 7: end
- 8: Initial populations
- 9: repeat
- 10: for  $t = 1:t_{max}$
- 11: for  $i = 1:a$
- 12: vertical convolution position update
- 13: horizontal convolution position update
- 14: square convolution position update
- 15: comprehensive location update
- 16: end for
- 17: calculate the best position
- 18: for  $j = 1:dim$
- 19: solution quality enhancement
- 20: end for
- 21: update the global best location
- 22: end for
- 23: until Convergence
- 24: return  $x_c, FAR_{be}, MAR_{be}, EDD_{be}$

---

**3.3. Design of Alarm Adaptive Delay Algorithm**

The principle of the alarm delay method is that the system starts alarming when a variable exceeds  $d$  consecutive alarm signals, and the system stops alarming when there are  $d$  consecutive non-alarm signals. The alarm adaptive delay method designed in this paper can dynamically adjust the size of the delay factor  $d$  based on the actual alarm data and the mean and variance of the alarm duration. The steps of the alarm adaptive delay method are as follows:

Step 1: Based on the original alarm threshold, the alarm interval data  $E$  processed by the delay factor  $d$  can be obtained, where  $E = [E_1, E_2, E_3, \dots, E_r]$ , and  $E_i$  is the time interval of the  $i$ th alarm ( $i = 1, 2, \dots, r$ ).

Step 2: Based on the alarm interval data  $E$ , calculate the total mean  $E_{avg}$  and the standard deviation  $S_E$  of  $E$ .

$$E_{avg} = \frac{\sum_{i=1}^r E_i}{r} \quad (23)$$

$$S_E = \sqrt{\frac{\sum_{i=1}^r (E_i - E_{avg})^2}{r - 1}} \quad (24)$$

Step 3: Based on the  $E_{avg}$  and  $S_E$ , calculate the coefficient of variation  $CV_d$  for the alarm interval data processed by the delay factor  $d$  [26].

$$CV_d = \frac{S_E}{E_{avg}} \quad (25)$$

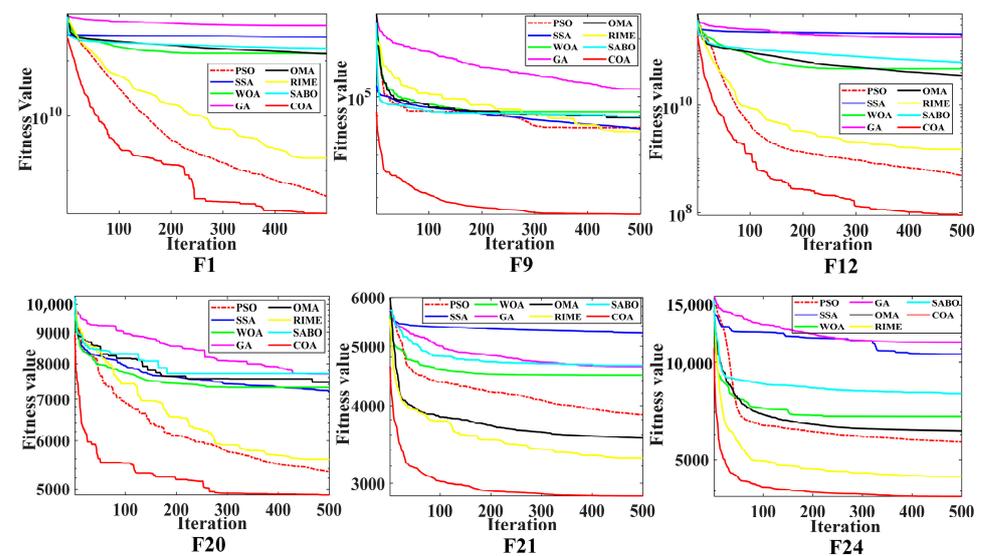
Step 4: Set the standard coefficient of variation  $CV_{st}$  to one, and compare the calculated  $CV_d$  with the  $CV_{st}$ . If  $CV_d > CV_{st}$ , update the delay factor  $d = d + CV_d$ , and return to step 1; if  $CV_d < CV_{st}$ , end the algorithm process, and obtain the final adaptive delay factor.

## 4. Experimental Analysis

### 4.1. Performance Validation of COA

In order to fully evaluate the performance of the COA, this paper compared it with seven intelligent optimization algorithms, including the subtraction-based averaging optimization (SABO) [27], optical microscope algorithm (OMA) [28], rimmed ice optimization (RIME) [29], as well as the classical genetic algorithm (GA), particle swarm optimization (PSO), whale optimization algorithm (WOA), and sparrow search optimization algorithm (SSA). The experimental dataset consisted of 12 test functions from the CEC 2017 benchmark. To make sure that the experiments are fair, the population size for all the algorithms was set to 20, the maximum number of iterations was set to 500, and the dimensionality was set to 100. Moreover, each algorithm was run five times separately, and the average value was taken as the final result to reduce random errors.

To visually observe the performance of the COA and the aforementioned seven optimization algorithms during the iteration process, an analysis of the iterative convergence curves of the eight optimization algorithms on test functions was conducted, as shown in Figure 4. Specifically, results from six challenging functions were selected for demonstration.



**Figure 4.** Iterative convergence curves for six test functions.

Figure 4 clearly demonstrates the variations in the *fitness* values during the optimization iteration process of the COA, SABO, OMA, RIME, GA, PSO, WOA, and SSA algorithms. From the convergence curves in Figure 4, it is evident that the COA consistently reaches the optimal value at the fastest rate during the iteration process of each test function, maintaining a leading convergence speed compared to the other seven algorithms. Furthermore, when reaching the maximum number of iterations, the *fitness* value of the COA is significantly lower than that of the other seven algorithms, indicating that the convergence accuracy of the COA exceeds that of the other algorithms. It can be observed that the COA possesses superior optimization capabilities, as well as stronger abilities in both local and global search.

Furthermore, to further validate the performance of the COA, the Friedman test was employed to compare the statistical differences between each group of results. The rank values were compared to evaluate the performance of the different algorithms, where smaller rank values indicate higher performance [30]. The Friedman rank and ranking results for each algorithm on the test functions are presented in Table 1.

**Table 1.** Friedman rank sum ranking of 12 algorithms on different test functions.

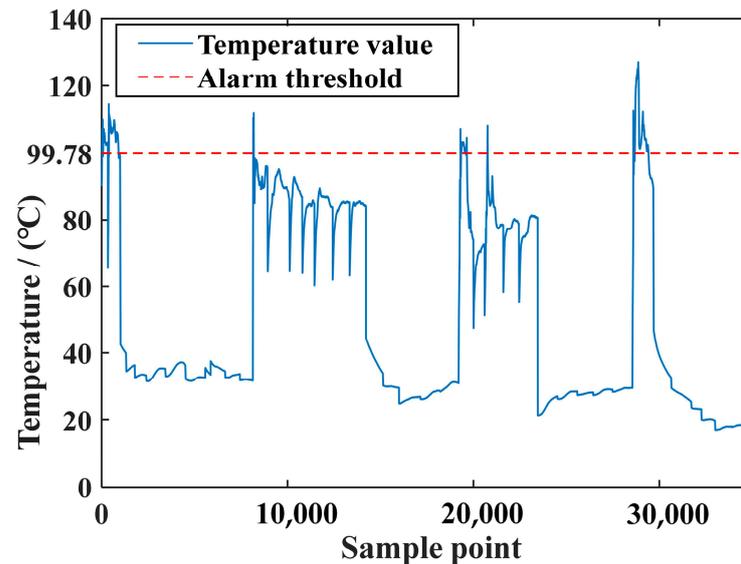
Function	COA	PSO	SSA	WOA	GA	OMA	RIME	SABO	COA Ranking
F1: Shifted and Rotated Bent Cigar Function	1	2	7	4.8	8	4.2	3	6	1
F2: Shifted and Rotated Sums of Different Power Functions	1.2	1.8	7.2	5.2	7.8	4	3	5.8	1
F3: Shifted and Rotated Zakharov Function	1	4.6	4.2	6.6	7.2	3.2	2	7.2	1
F5: Shifted and Rotated Rastrigin's Function	1	2.2	6.4	5.6	8	4.4	2.8	5.6	1
F8: Shifted and Rotated Lunacek BiRastrigin Function	1.2	3.2	7	5.6	7.8	3.8	1.8	5.6	1
F9: Shifted and Rotated Levy Function	1	5	2.6	5.8	7.8	5.4	2.6	5.8	1
F12: Hybrid Function 2 (N = 3)	1	2	7	4.8	8	4.2	3	6	1
F16: Hybrid Function 6 (N = 4)	1.4	1.6	8	6.2	6.8	4	3	5	1
F20: Hybrid Function 6 (N = 6)	1.4	1.6	5.2	6	6.8	6.2	3	5.8	1
F21: Composition Function 1 (N = 3)	1	4	7.4	5.8	6.6	2.8	2.2	6.2	1
F24: Composition Function 4 (N = 4)	1	3.6	7.4	5.2	7.6	3.8	2	5.4	1
F30: Composition Function 10 (N = 3)	1	2	7.2	4.4	7.8	4.8	3	5.8	1
Average	1.1	3	7	5.5	7.5	4.2	2.6	5.9	1

From Table 1, it can be observed that in the Friedman test conducted on each test function, the COA consistently achieved the smallest rank. The average rankings of the 8 algorithms were COA, RIME, PSO, OMA, WOA, SABO, SSA, and GA. The COA ranked first, indicating that the COA holds a statistical and performance advantage over the other algorithms.

#### 4.2. Performance Analysis of Threshold Optimization

##### 4.2.1. Data Validation of 1000 MW USCTPU

To verify the intelligent alarm capability of the proposed alarm threshold optimization method during the normal operation of the unit, experimental data from a 1000 MW-level secondary reheating ultra-supercritical DC boiler system in a certain thermal power plant were collected. The data are the variations in a high-pressure feed pump BV winding temperature over time, with an initial alarm threshold set at 99.78. The sample data curve is shown in Figure 5.



**Figure 5.** High-pressure feed pump BV winding temperature data curve.

Firstly, the process values of the BV winding temperature data of the high-pressure feedwater pump were separately collected under normal and abnormal working conditions based on the alarm threshold, resulting in two groups of data. Then, according to Equation (5), the optimal window widths for the two groups of data were calculated as

$u_{b1} = 3.4$  and  $u_{b2} = 1.4$ , respectively. Subsequently, Equation (1) was used to fit the PDF curves of the normal and abnormal states, as shown in Figure 6.

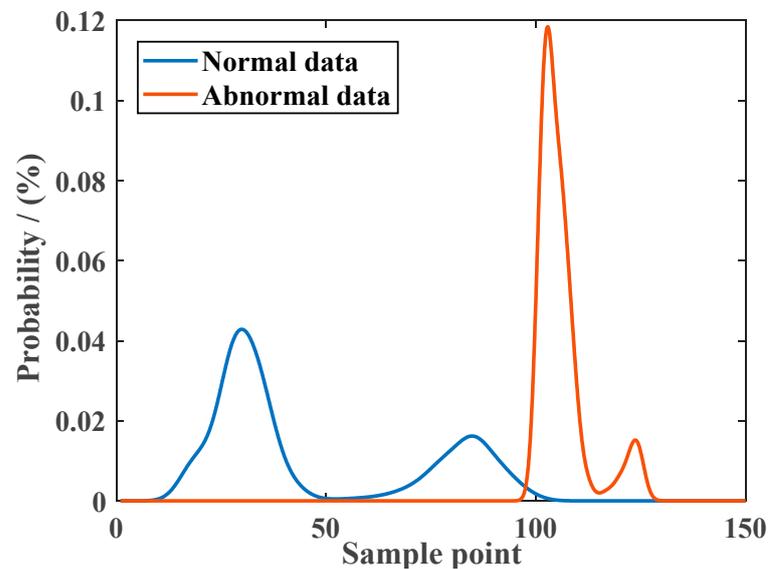


Figure 6. PDF curve distribution.

According to Figure 6, the PDFs of the normal and abnormal state data can be obtained, denoted as  $L(x)$  and  $R(x)$ , respectively. It can be expressed as follows:

$$L(x) = \frac{1}{32753 \times 3.4} \sum_{q=1}^{32753} \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x-x_q}{\sqrt{2} \times 3.4}\right)^2} \quad (26)$$

$$R(x) = \frac{1}{1918 \times 1.4} \sum_{d=1}^{1918} \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x-x_d}{\sqrt{2} \times 1.4}\right)^2} \quad (27)$$

Based on Equations (26) and (27), we can further derive the expressions for the FAR, MAR, and EDD, as expressed below

$$FAR = \int_{x_b}^{+\infty} \frac{1}{32753 \times 3.4} \sum_{q=1}^{32753} \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x-x_q}{\sqrt{2} \times 3.4}\right)^2} dx \quad (28)$$

$$MAR = \int_{-\infty}^{x_b} \frac{1}{1918 \times 1.4} \sum_{d=1}^{1918} \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x-x_d}{\sqrt{2} \times 1.4}\right)^2} dx \quad (29)$$

$$EDD = \frac{\int_{-\infty}^{x_c} \frac{1}{1918 \times 1.4} \sum_{d=1}^{1918} \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x-x_d}{\sqrt{2} \times 1.4}\right)^2} dx}{1 - \int_{-\infty}^{x_c} \frac{1}{1918 \times 1.4} \sum_{d=1}^{1918} \frac{1}{\sqrt{2\pi}} e^{-\left(\frac{x-x_d}{\sqrt{2} \times 1.4}\right)^2} dx} \quad (30)$$

Next, by substituting the expressions for the FAR, MAR, and EDD into Equation (22), we obtain the objective function  $F(x)$  for optimizing the alarm threshold, where the  $FAR^* = 0.02$ ,  $MAR^* = 0.02$ , and  $EDD^* = 0.04$ . The optimization is performed using the COA; at this point, the variable search range is set to [95, 105], and the *fitness* value curve during the optimization process with increasing iteration numbers as well as the curve showing the variation in the objective function values under different alarm thresholds, are illustrated in Figure 7.

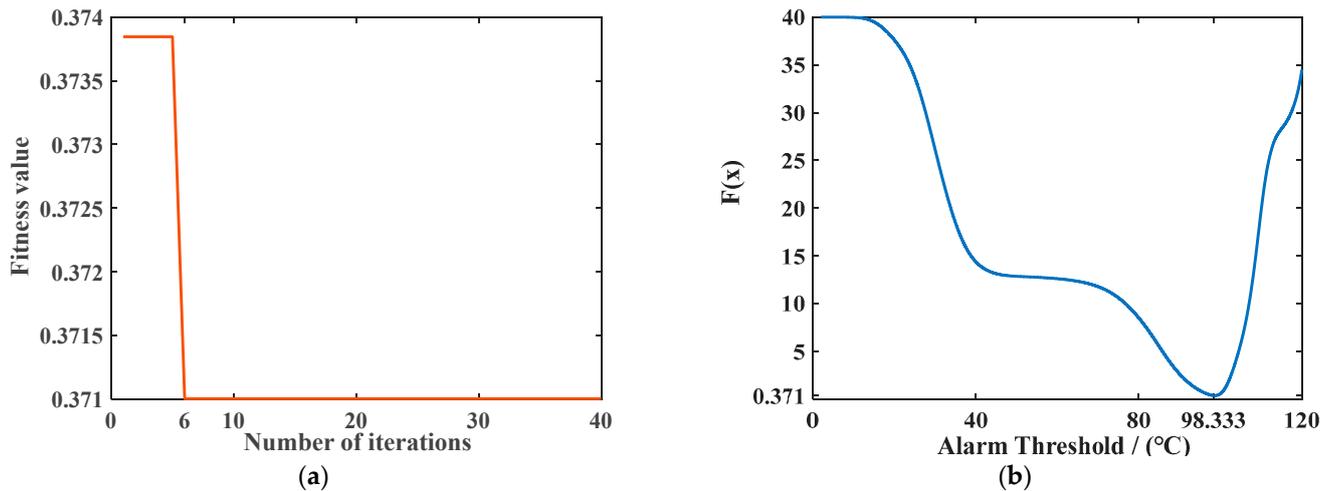


Figure 7. (a) COA iterative convergence curves; (b) curve of change in  $F(x)$ .

According to Figure 7a, it can be observed that when using the COA to optimize the objective function, the minimum value of the function is reached after 6 iterations, with a value of 0.371, which remains unchanged in the subsequent iterations. Additionally, as shown in Figure 7b, the optimal alarm threshold corresponding to the minimum function value is  $x_c = 98.333$ . Furthermore, based on Equations (28)–(30), the optimal values for the  $FAR$ ,  $MAR$ , and  $EDD$  are calculated to be  $FAR = 0.0079$ ,  $MAR = 0.0068$ , and  $EDD = 0.0069$ , respectively.

Finally, in order to better validate the effectiveness of the proposed threshold optimization method in this paper, the threshold optimization method proposed in this paper is compared with the  $3\sigma$  method commonly used in the industry, NOA threshold optimization method, PSO threshold optimization method, and the optimization index results of the original system alarm threshold. The results are shown in Table 2.

Table 2. Comparison of indicator results.

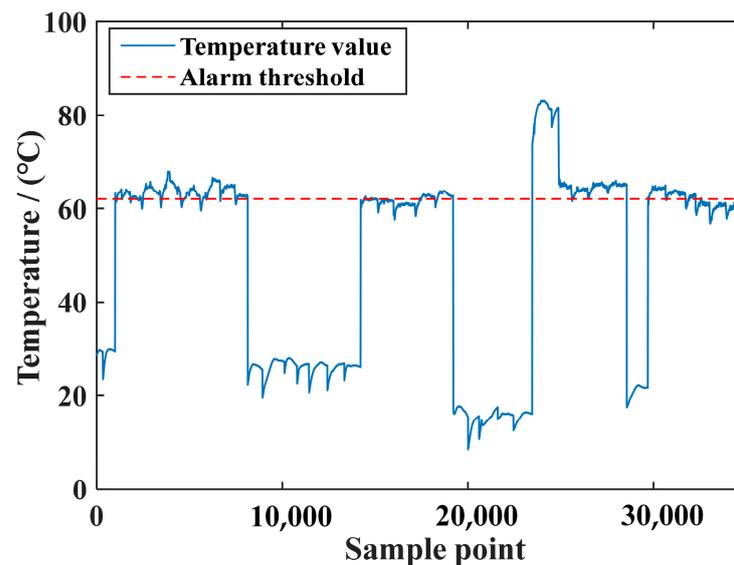
Index	Method	$FAR$	$MAR$	$EDD$	Alarm Threshold
1	Original system	0.0046	0.0423	0.044	99.78
2	$3\sigma$	0.0029	0.0538	0.0568	106.11
3	KDE + NOA	0.0073	0.01	0.0101	98.5888
4	KDE + PSO	0.0072	0.0103	0.0104	98.6119
5	KDE + COA	0.0079	0.0098	0.0069	98.3333

From the results for the first and fifth methods in Table 2, it is evident that compared to the original system before optimization, although the proposed threshold optimization method in this paper has resulted in a slight increase in the  $FAR$  by only 0.0033, the obtained  $FAR$  remains within an acceptable range. Furthermore, the original values of the  $MAR$  and  $EDD$  exceeded the maximum acceptable range, posing safety hazards to production and reducing the response speed of the alarm system. However, after optimization using the proposed method, both the  $MAR$  and  $EDD$  are reduced to approximately one-sixth of their original values, and the optimized results fall within the maximum acceptable range. This achievement demonstrates that the proposed threshold optimization method significantly reduces the  $MAR$  and  $EDD$  without significantly increasing the  $FAR$ , thereby effectively mitigating the safety hazards associated with excessive false negatives and improving the response speed and performance of the alarm system. Although using the  $3\sigma$  method can greatly reduce the  $FAR$ , the  $MAR$  and  $EDD$  are much higher than the threshold optimization method in this paper. Additionally, from the results for the third, fourth, and fifth methods in Table 2, it can be inferred that the utilization of the COA method in this paper further

reduces the *MAR* and *EDD*, indicating that the COA method employed in the optimization process exhibits superior performance.

#### 4.2.2. Data Validation of 660 MW USCTPU

To validate the applicability and scalability of the threshold method on the different types of USCTPUs, experimental data from the 660 MW ultra-supercritical variable-pressure direct current boiler system of a thermal power plant were collected. The data depict the variation in temperature over time in the non-drive section of the high-pressure feedwater pump A. The initial alarm threshold is set at 62, and the sample data curve is shown in Figure 8.



**Figure 8.** High-pressure feed pump A motor non-driven section temperature data curve.

To validate the effectiveness of the threshold optimization method proposed in this paper on the 660 MW USCTPU, the optimization indicators resulting from the threshold optimization method proposed in this paper are compared with those under the original system alarm thresholds, where the  $FAR^* = 0.02$ ,  $MAR^* = 0.02$ , and  $EDD^* = 0.04$ . The final results are shown in Table 3.

**Table 3.** The performance indexes of the threshold optimization method are compared with the original system and the traditional threshold optimization method.

Index	Method	<i>FAR</i>	<i>MAR</i>	<i>EDD</i>	Alarm Threshold
1	Original system	0.12189	0.074827	0.080879	62
2	KDE + COA	0.15697	0.015809	0.016063	61.2357
3	$3\sigma$	0.038	0.1538	0.1818	65.46
4	KDE + NOA	0.155	0.055	0.058	61.465
5	KDE + PSO	0.1546	0.067	0.072	61.33

From Table 3, it can be observed that compared to the results of the original system before optimization, the threshold optimization method proposed in this paper increases the *FAR* by 0.035, but it remains within the maximum acceptable range without affecting the safe operation. However, both the *MAR* and *EDD* exceeded the maximum acceptable range before optimization, posing safety hazards to production. After optimization using the method proposed in this paper, the *MAR* and *EDD* are significantly reduced and are within the maximum acceptable range. Compared with the  $3\sigma$  method, the *MAR* and *EDD* optimized by the threshold method proposed in this paper are 1/10 of the  $3\sigma$  method, indicating that using the threshold method can improve the response speed of the alarm

system. Additionally, from the results for the second, fourth, and fifth methods in Table 3, it can be inferred that the utilization of the COA method in this paper further reduces the *MAR* and *EDD*, indicating that the COA method employed in the optimization process exhibits superior performance. This demonstrates that the proposed threshold optimization method also exhibits a superior performance on the 660 MW USCTPU.

#### 4.2.3. Performance Verification of the Proposed Threshold Optimization Method under System Fault Conditions

To verify the robustness of the threshold method proposed in this paper, it is applied when the system experiences faults. Fault data from a 1000 MW-level secondary reheating ultra-supercritical direct current boiler system were collected. The fault observed was an elevated temperature in the winding 1 of the high-pressure feedwater pump AV. The sample data operational curve is shown in Figure 9.

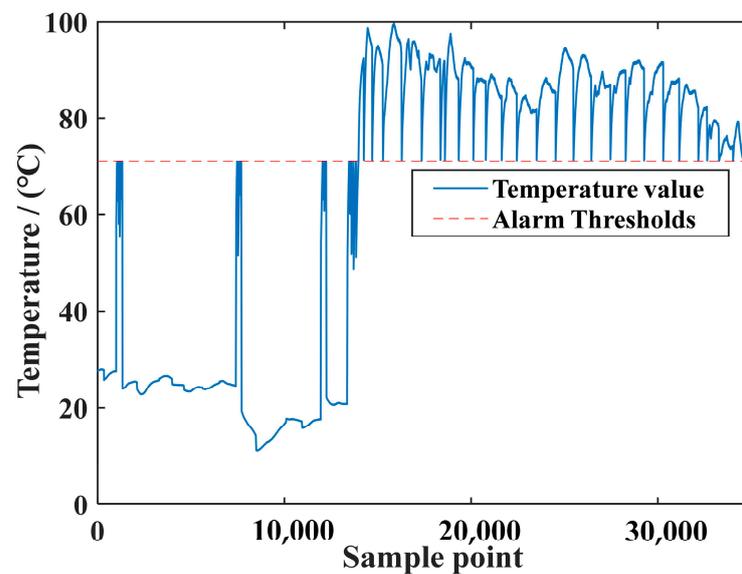


Figure 9. High-pressure feed pump AV winding temperature operating curve.

From Figure 9, it is evident that the system experienced a fault at sample point 13,967, after which the temperature remained above the alarm threshold set in the original system. Subsequently, the threshold optimization method proposed in this paper was applied to optimize the alarm thresholds of the original system. The optimized results were then compared with the optimization indicator results under the original system's alarm thresholds, where the  $FAR^* = 0.02$ ,  $MAR^* = 0.002$ , and  $EDD^* = 0.002$ . The comparative results are presented in Table 4.

Table 4. Comparison of performance indicators under fault conditions.

Index	Method	<i>FAR</i>	<i>MAR</i>	<i>EDD</i>	Alarm Threshold
1	Original system	0.0094	0.004	0.004	71
2	KDE + COA	0.0129	0.00146	0.00146	70.4312
3	$3\sigma$	0.005	0.035	0.036	75.472
4	KDE + NOA	0.011	0.018	0.018	70.465
5	KDE + PSO	0.012	0.0174	0.018	71.3324

From Table 4, it can be observed that after optimizing the alarm thresholds using the method proposed in this paper under fault conditions, the *MAR* and *EDD* are significantly reduced and controlled within the safety range. Although there is an increase in the *FAR*, it is also within the safety range. Overall, the threshold optimization method proposed

in this paper demonstrates good performance even under fault conditions. As shown by the second, third, fourth, and fifth methods in Table 4, the results obtained in this section are similar to those in Sections 4.2.1 and 4.2.3. This indicates that when the system fails, compared with traditional threshold design methods, the proposed threshold method is the optimal solution.

#### 4.2.4. Real-Time Analysis of the Proposed Threshold Optimization Method

To validate the effectiveness of the threshold optimization method on real-time operational data, the algorithm was deployed onto a user interface within an online platform. The threshold optimization algorithm was integrated as a functional module on the user interface, enabling all functionalities of the optimization algorithm. Real-time operational data stored in the Distributed Control System (DCS) were inputted into the threshold optimization module through the database on the online platform to obtain the optimized performance metric results. This integration ensured seamless connectivity with existing alarm systems. The results before and after the optimization were visualized on the platform's graphical interface.

The real-time operational data of the high-pressure main steam temperature from a 1000 MW-level secondary reheating ultra-supercritical direct current boiler system were taken as the research subject. The sampling frequency was set at 1 s, and 9.5 h of real-time operational data were collected. Among them, the alarm threshold is 556 °C. The real-time operational data curve is depicted in Figure 10.

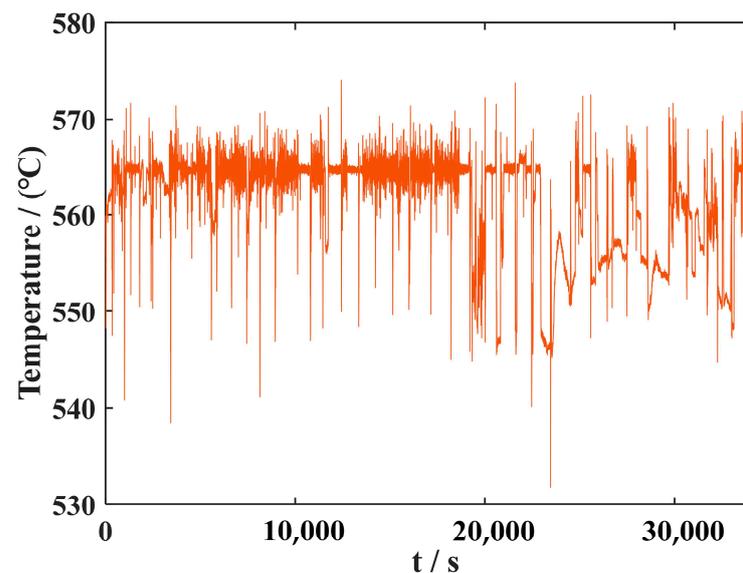
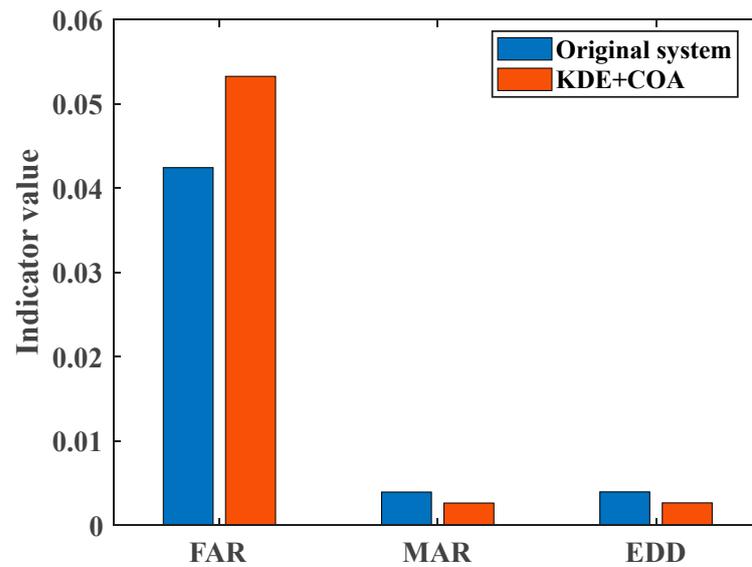


Figure 10. Real-time operation curve of high-pressure main steam temperature.

The performance comparison results of the alarm threshold before and after optimization using the threshold optimization algorithm module proposed in this article are displayed in the graphical configuration of the online platform, as shown in Figure 11. From Figure 11, it can be observed that during real-time operation, the threshold optimization algorithm proposed in this paper still achieves a further reduction in the *MAR* and *EDD* without significantly increasing the *FAR*. This demonstrates the effectiveness of the threshold optimization algorithm proposed in this paper during the real-time operation of the unit.

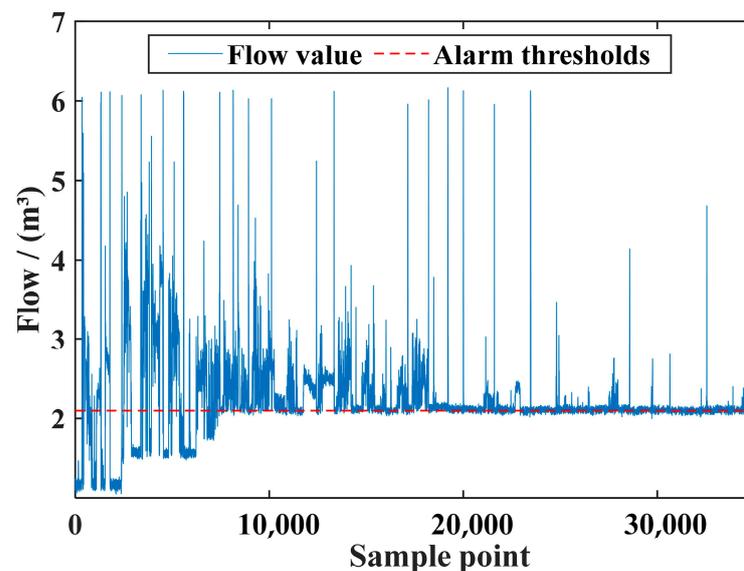


**Figure 11.** Comparison of the performance indicators before and after the lower threshold optimization in real-time running.

#### 4.3. Performance Analysis of Alarm Adaptive Delay Algorithm

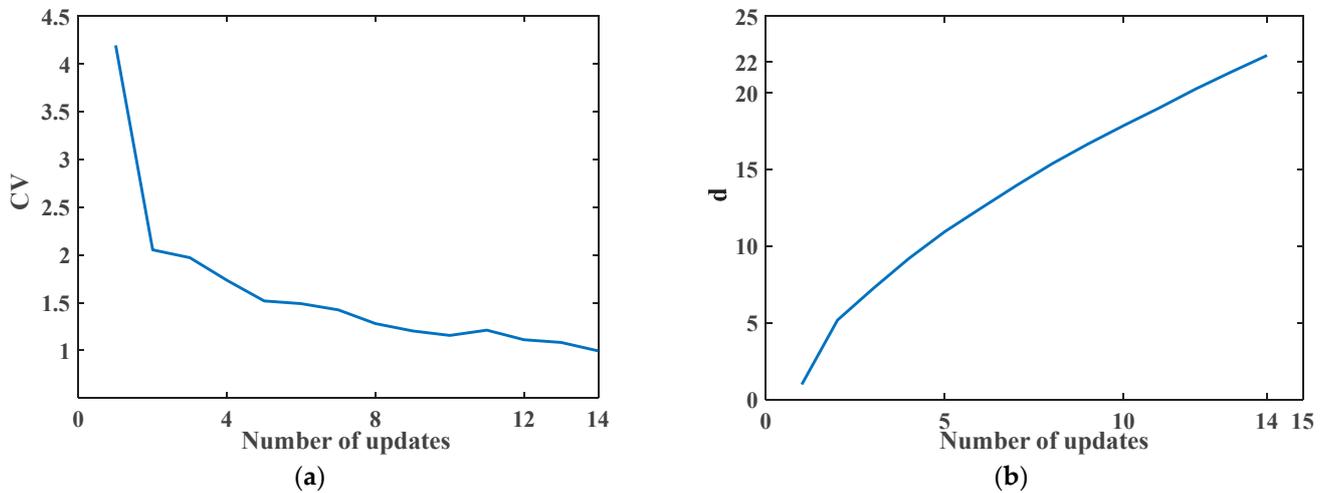
##### 4.3.1. Data Verification of 1000 MW USCTPU

To validate the suppression capability of nuisance alarms during the normal operation of a 1000 WM USCTPU using the proposed adaptive alarm delay method, the reheat desuperheating water flow data in a 1000 MW-level secondary reheat ultra-supercritical DC boiler system were selected as the research subject, with its operational curve depicted in Figure 12. It can be observed from Figure 12 that the flow data contain numerous sample values fluctuating around the alarm threshold. Whenever the sample values of the flow data fluctuate below the alarm threshold, an alarm is triggered, leading to repetitive alarm signals, known as nuisance alarms. These alarms are often not caused by variable anomalies but rather by strong correlations among alarm variables, with the repeated occurrence of alarm signals potentially influenced by other parameters rather than indicating a malfunction. Upon statistical analysis, the original number of alarms reached 1647.



**Figure 12.** Operating data of reheat desuperheating water flow.

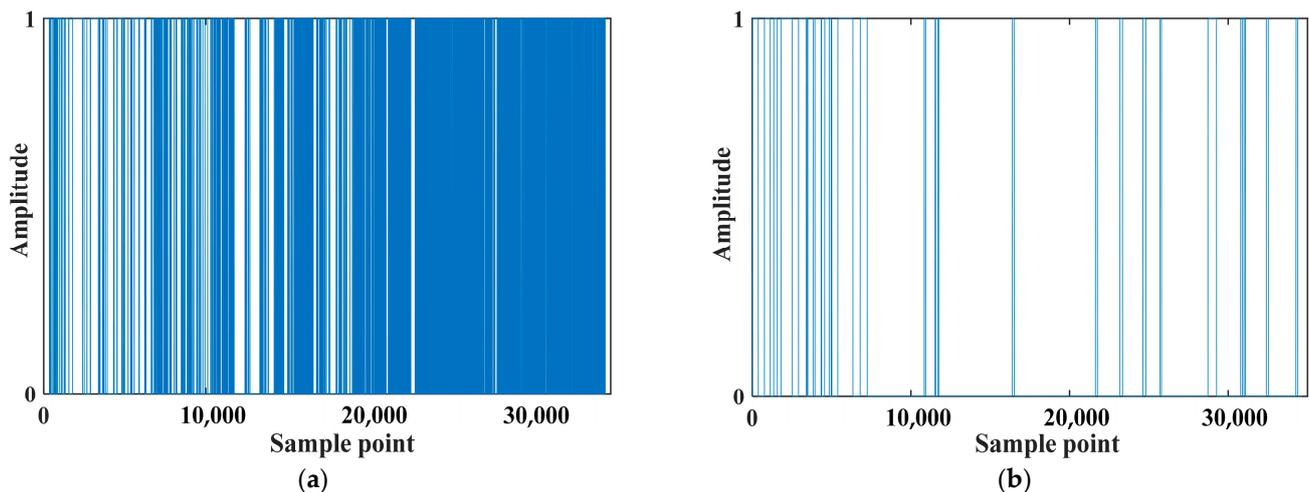
Firstly, based on the pre-set alarm threshold and the original alarm data, the time interval data for each alarm are obtained. Then, the alarm system is optimized using an alarm adaptive delay method according to the alarm time interval data. The changes in the coefficient of variation  $CV$  and the delay factor  $d$  are shown in Figure 13.



**Figure 13.** (a) Update of the coefficient of variation  $CV$ ; (b) updated results for the delay factor  $d$ .

From Figure 13a, it can be seen that the value of the  $CV$  is 0.994 after the 14th update, which is less than the  $CV_{st}$ . This indicates that the setting of the delay factor is reasonable at this point, and the algorithm meets the stopping update condition. Figure 13b shows that the optimal delay factor after the optimization using the alarm adaptive delay method is  $d = 22$ .

In order to visually demonstrate the effect of setting the delay factor on suppressing nuisance alarms, a comparison is made between the alarm signal data before and after setting the delay factor. The alarm signal values are recorded as 1 when an alarm occurs, and 0 otherwise. The results are shown in Figure 14.



**Figure 14.** (a) Alarm signal data before setting delay factor; (b) alarm signal data after setting delay factor.

As shown in Figure 14a, without setting an adaptive delay factor in the traditional alarm system, the number of alarms is 1647, severely affecting the operation of the staff. However, in Figure 14b, due to the reasonable setting of the adaptive delay factor, the number of alarms is reduced to 27, which is 1/61 of the original number of alarms. Therefore,

adding an appropriate delay factor in the alarm system can significantly reduce the number of nuisance alarms and improve the quality of the alarm signals.

#### 4.3.2. Data Verification of 660 MW USCTPU

The expansion data of the gas exhaust expansion section in the 660 MW ultra-supercritical direct current boiler system are used as the research object to verify the effectiveness and universality of the proposed adaptive delayed alarm method on the different types of USCTPU. The operational curves are depicted in Figure 15.

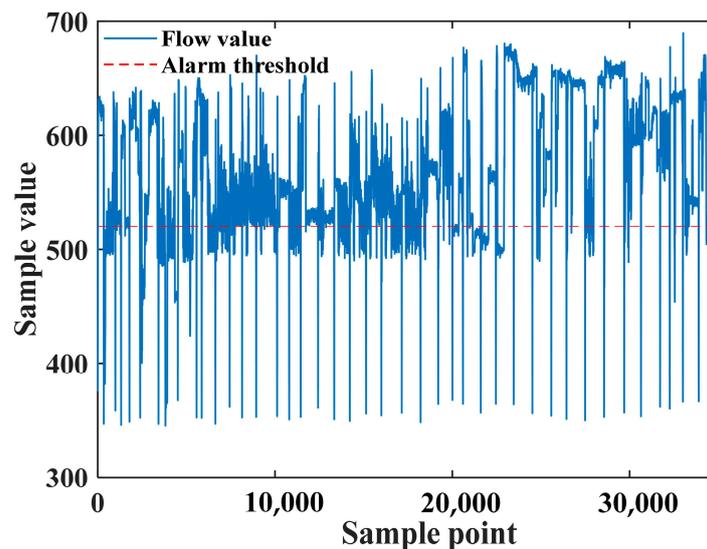


Figure 15. Data of gas exhaust expansion section.

The alarm signal data before and after setting the delay factor are compared, and the results are shown in Figure 16. As shown in Figure 16a, the number of alarms without setting the delay factor is 1121, while after setting the adaptive delay factor, the number of alarms decreases to 41. It can be seen that the adaptive delay alarm method proposed in this paper can greatly reduce the number of disturbance alarms. This also verifies the effectiveness and universality of the proposed delay alarm method on the different types of ultra-supercritical thermal power units.

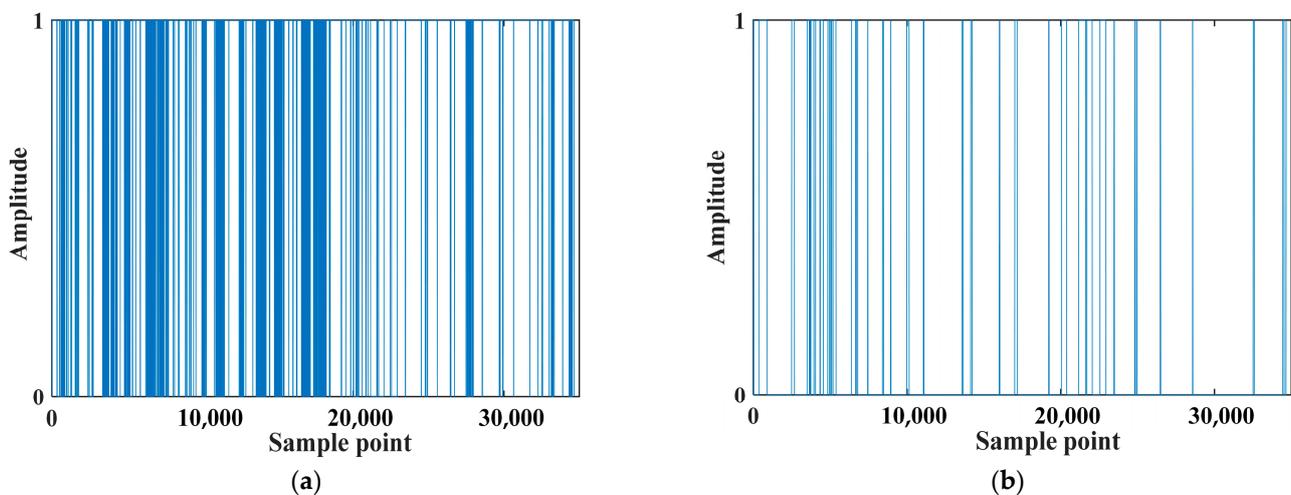
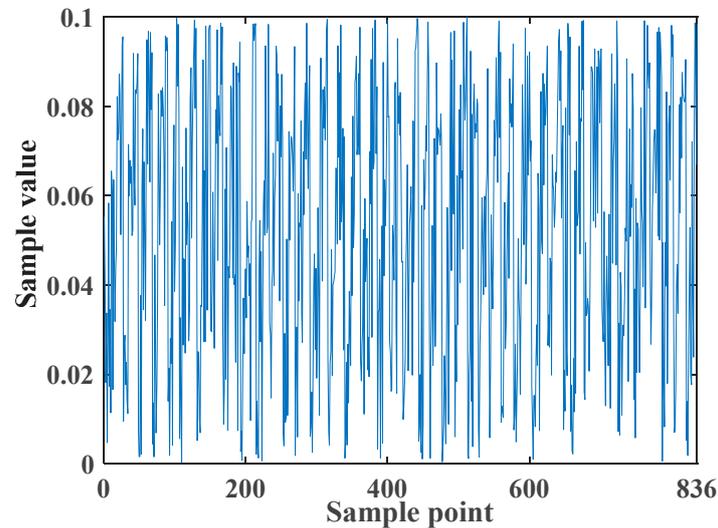


Figure 16. (a) Alarm signal data before adding adaptive delay factor; (b) alarm signal data after adding adaptive delay factor.

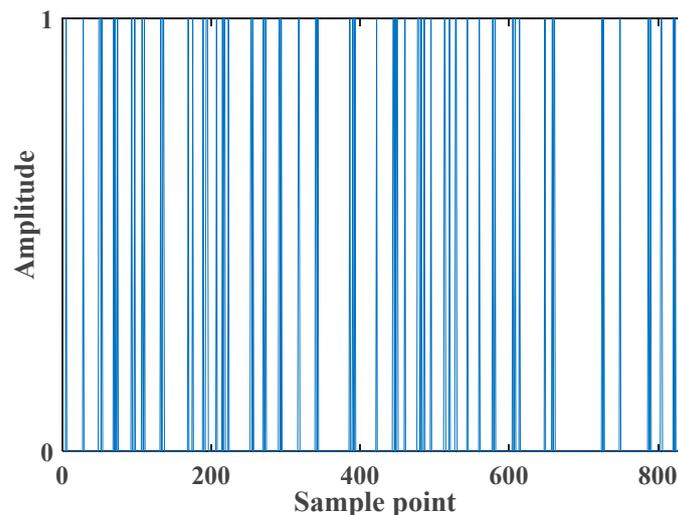
#### 4.3.3. Performance Verification under System Fault

To further investigate the effectiveness and reliability of the proposed adaptive delay alarm method when the system experiences faults, data from a 1000 MW USCTPU experiencing a fault are used as the research subject. The fault scenario involves the RSV2 opening too slowly, and the operational curve during the fault occurrence is depicted in Figure 17.



**Figure 17.** RSV2 opening speed curve when a fault occurs.

From Figure 17, it can be observed that during the fault occurrence, the opening speed of RSV2 is too slow, below 0.1, resulting in 836 alarms. Subsequently, when applying the adaptive delay alarm method during the fault occurrence, the generated alarm signals are shown in Figure 18. It can be seen from Figure 18 that the number of alarms after setting the adaptive delay factor is 118. Compared to the original system, the number of alarms has been significantly reduced. Therefore, it can be concluded that under fault conditions, the adaptive delay alarm method proposed in this paper still performs well, further validating the effectiveness and reliability of the proposed adaptive delay alarm method.



**Figure 18.** Alarm signal data after setting delay factor.

#### 4.3.4. Real-Time Analysis of the Proposed Adaptive Delay Alarm Method

To verify the effectiveness of the proposed adaptive delay alarm method validated on real-time operational data, the adaptive delay alarm method proposed in this paper

is deployed onto an online platform. The adaptive delay alarm algorithm and the alarm signal generation algorithm's code are integrated into different modules, replacing the entire functionality of the algorithms. Real-time operational data of a 1000 MW USCTPU DCS, stored in the online platform's database, are utilized, with the measurement point being the high-pressure main steam flow rate. The original alarm threshold is set to 250. The sampling period is 1 s, and 4 h of real-time operational data are captured as the test data. The real-time operational curve is depicted in Figure 19.

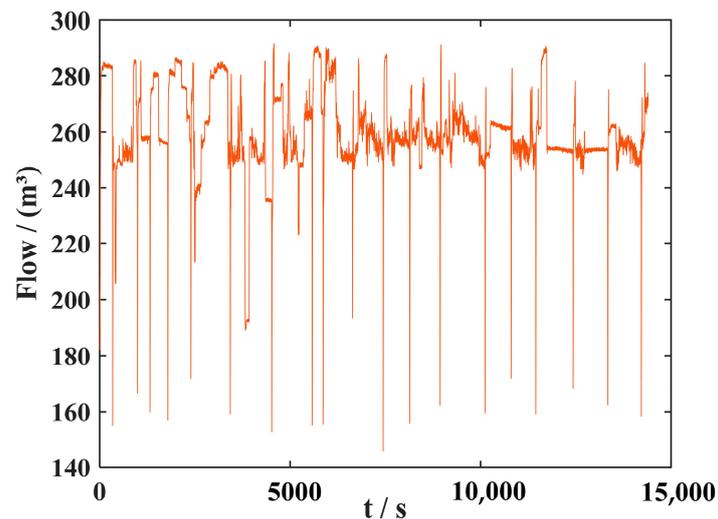


Figure 19. Real-time operation data curve of high-pressure main steam flow rate.

The real-time operational data are inputted into the alarm signal generation module, and based on the alarm threshold, the number of alarms is determined. After statistical analysis, it is found that a total of 184 alarms occurred within the 4 h period. Subsequently, the real-time operational data are processed using the adaptive delay alarm module. The processed alarm signals are depicted in Figure 20. From Figure 20, it can be observed that after applying the delay factor, the number of alarms reduces to 27. This indicates that the adaptive delay alarm method proposed in this paper exhibits a superior performance even during real-time system operation, effectively reducing the number of alarms.

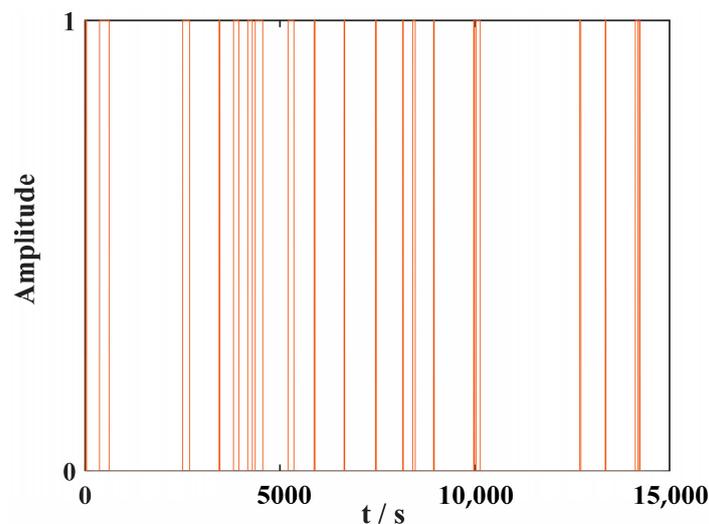


Figure 20. Real-time alarm data after setting delayed alarms.

## 5. Conclusions

This paper has proposed an intelligent alarm method, which includes the optimization method of alarm threshold based on KDE and COA. Unlike traditional methods, this method has taken the *MAR* as the optimization objective, aiming to significantly reduce the *MAR* while controlling the *FAR* within a reasonable range. It also considers the *EDD* of the alarm system and combines the *FAR*, *MAR*, and *EDD* to construct the objective function for threshold optimization. Finally, the COA has been used to optimize and solve the objective function, obtaining the optimized alarm threshold. The results have shown that the proposed threshold optimization method can significantly reduce the *MAR* and *EDD*, almost without increasing the *FAR*, thus eliminating the safety hazards that a high *MAR* may bring and improving the safety of USCTPUs.

Furthermore, to address the issue of a high number of nuisance alarms, this paper has introduced an alarm adaptive delay method. Initially, based on the pre-set alarm threshold and original alarm data, the time intervals between each alarm are obtained. Subsequently, the delay factor is adaptively updated based on the mean and variance of these alarm time intervals. The results have indicated that the inclusion of the delay factor can significantly reduce the number of nuisance alarms and enhance the quality of alarms.

The intelligent alarm method proposed in this paper is designed for single-variable alarm systems, but in practical production operations, multi-variable alarm situations are more common. Once an abnormality occurs in one variable, it may cause simultaneous abnormalities in other related variables, leading to alarm surges, which are highly detrimental to safe production operations. Therefore, future work will focus on studying the correlation of variables in multi-variable alarm systems. We will determine the topology of variables by analyzing causal relationships among them, thereby more accurately identifying the source of alarms.

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

**Funding:** This research was funded by the Key R&D Program Project of Zhejiang Province under Grants No. 2022C01206.

**Data Availability Statement:** Some or all data, models, or codes generated or used during the study are available from the corresponding author by request.

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

## References

1. Wei, C.; Wang, Y.; Shen, Z.; Xiao, D.; Bai, X.; Chen, H. AUQ-ADMM algorithm-based peer-to-peer trading strategy in large-scale interconnected microgrid systems considering carbon trading. *IEEE Syst. J.* **2023**, *17*, 6248–6259. [[CrossRef](#)]
2. He, K.X.; Wang, T.; Zhang, F.K.; Jin, X. Anomaly detection and early warning via a novel multiblock-based method with applications to thermal power plants. *Measurement* **2022**, *193*, 110979. [[CrossRef](#)]
3. Lai, S.; Chen, T. A method for pattern mining in multiple alarm flood sequences. *Chem. Eng. Res. Des.* **2017**, *117*, 831–839. [[CrossRef](#)]
4. Hu, W.; Chen, T.; Shah, S.L. Discovering Association Rules of Mode-Dependent Alarms From Alarm and Event Logs. *IEEE Trans. Control Syst. Technol.* **2018**, *26*, 971–983. [[CrossRef](#)]
5. Jablonski, A.; Barszcz, T.; Bielecka, M. Modeling of Probability Distribution Functions for Automatic Threshold Calculation in Condition Monitoring Systems. *Measurement* **2013**, *46*, 727–738. [[CrossRef](#)]
6. Fricker, R.D.; Banschbach, D. Optimizing Biosurveillance Systems that Use Threshold-based Event Detection Methods. *IEEE Trans. Control Syst. Technol.* **2012**, *13*, 117–128. [[CrossRef](#)]
7. Fricker, R.D.; Banschbach, D. A correlation consistency based multivariate alarm thresholds optimization approach. *ISA Trans.* **2016**, *65*, 37–43.
8. Aslansefat, K.; Gogani, M.B.; Kabir, S.; Shoorehdeli, M.A.; Yari, M. Performance evaluation and design for variable threshold alarm systems through semi-Markov process. *ISA Trans.* **2020**, *97*, 282–295. [[CrossRef](#)]

9. Xiong, W.; Wang, J.; Chen, K. Multivariate Alarm Systems for Time-Varying Processes Using Bayesian Filters With Applications to Electrical Pumps. *IEEE Trans. Ind. Inform.* **2018**, *14*, 504–513. [[CrossRef](#)]
10. Zhi, Z.; Tao, L.; Guan, H.; Si, L.; Gai, L.; Wei, H. A Fault Detection Method Based on Data Reliability and Interval Evidence Reasoning. *Acta Autom. Sin.* **2020**, *46*, 2628–2637.
11. Han, L.; Gao, H.; Xu, Y.; Zhu, Q. Combining FAP, MAP and correlation analysis for multivariate alarm thresholds optimization in industrial process. *J. Loss Prev. Proc.* **2016**, *40*, 471–478. [[CrossRef](#)]
12. Zhang, G.; Wang, Z. Correlation Degree and Clustering Analysis-Based Alarm Threshold Optimization. *Processes* **2018**, *10*, 224. [[CrossRef](#)]
13. Tian, W.; Zhang, G.; Liang, H. Alarm clustering analysis and ACO based multi-variable alarms thresholds optimization in chemical processes. *Process Saf. Environ. Prot.* **2018**, *113*, 132–140. [[CrossRef](#)]
14. Zhang, G.; Wang, Z.; Mei, H. Sensitivity clustering and ROC curve based alarm threshold optimization. *Process Saf. Environ. Prot.* **2018**, *141*, 471–478. [[CrossRef](#)]
15. Zhao, Y.; Zhao, C.H. Dynamic multivariate threshold optimization and alarming for nonstationary processes subject to varying conditions. *Control Eng. Pract.* **2018**, *124*, 105180. [[CrossRef](#)]
16. Cai, Y.; Hong, L. Satisfaction Optimization Based Self-adaptive Adjustment of Process Alarm Thresholds. *Control Eng. China* **2019**, *26*, 417–422.
17. Hashim, H.; Clifford, E.; Ryan, P. False Alarm Moderation for Performance Monitoring in Industrial Water Distribution Systems. *Adv. Eng. Inform.* **2022**, *52*, 101592. [[CrossRef](#)]
18. Luo, Y.; Gopaluni, B.; Cao, L.; Wang, Y.; Cheng, J. Adaptive online optimization of alarm thresholds using multilayer Bayesian networks and active transfer entropy. *Control Eng. Pract.* **2023**, *137*, 105534. [[CrossRef](#)]
19. Wang, Y.; Ding, Y.; Shahrapour, S. TAKDE: Temporal Adaptive Kernel Density Estimator for Real-Time Dynamic Density Estimation. *IEEE Trans. Pattern Anal.* **2023**, *45*, 13831–13843. [[CrossRef](#)]
20. Shi, X.; Gu, L.; Li, X.; Jiang, T.; Gao, T. Automated spectral transfer learning strategy for semi-supervised regression on Chlorophyll-a retrievals with Sentinel-2 imagery. *Int. J. Digit. Earth* **2024**, *17*, 2313856. [[CrossRef](#)]
21. Hu, B.; Li, Y.; Yang, H.; Wang, H. Wind speed model based on kernel density estimation and its application in reliability assessment of generating systems. *J. Mod. Power Syst. Clean Energy* **2017**, *5*, 220–227. [[CrossRef](#)]
22. Wei, S.; Liu, X.; Shi, H.; Gan, J. Remaining Useful Life Prediction of High-Dimensional Kernel Density Estimation With Adaptive Relative Density Window Width Considering Multisource Information Fusion. *IEEE Sens. J.* **2024**, *24*, 6543–6568. [[CrossRef](#)]
23. Yun, S.; Zanetti, R.; Jones, B. Kernel-based ensemble gaussian mixture filtering for orbit determination with sparse data. *Adv. Space Res.* **2022**, *69*, 4179–4197. [[CrossRef](#)]
24. Chen, K.; Wei, S.; Zhang, J. Intelligent Optimization Algorithm Based on Two-dimensional Convolution Operation. *J. Armored Forces* **2023**, *2*, 102–108.
25. Zeng, Z.; Tan, W.; Zhou, R. Performance assessment for generalized delay-times in alarm configuration. *J. Process Control* **2017**, *57*, 80–101. [[CrossRef](#)]
26. Yu, H.; Gao, M.; Zhang, H.; Chen, J.; Liu, J.; Yue, G. Dynamic Modeling of Key Operating Parameters for Supercritical Circulating Fluidized Bed Units based on Data-Knowledge-Driven Method. *J. Therm. Sci.* **2024**. [[CrossRef](#)]
27. Trojovský, P.; Dehghani, M. Subtraction-Average-Based Optimizer: A New Swarm-Inspired Metaheuristic Algorithm for Solving Optimization Problems. *Biomimetics* **2023**, *8*, 149. [[CrossRef](#)] [[PubMed](#)]
28. Cheng, M.; Sholeh, M. Optical microscope algorithm: A new metaheuristic inspired by microscope magnification for solving engineering optimization problems. *Knowl.-Based Syst.* **2023**, *279*, 110939. [[CrossRef](#)]
29. Su, H.; Zhao, D.; Heidari, A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H.L. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. [[CrossRef](#)]
30. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.