Research on intelligent alarms in nuclear power plants based on machine learning

Jianhua Chen[a], Sijuan Chen[a*], Jinye Guo[a],Ming Yang[a], Chenxi Zhang[a], Faqiang Qian[a]
[a] Shenzhen Key Leboratory of Nuclear and Radiation Safety,Institute for Advanced Study in Nuclear Energy & Safety, College of Physics and Optoelectronic Engineering, Shenzhen University, Shenzhen 518060, China
[*] **Corresponding author:** chensijuan@szu.edu.cn

Abstract: Alarm systems play a vital role in the process industry, ensuring safety and reliability. Most of the alarms triggered during the alarm flood are useless nuisance alarms, and poor alarm management is one of the main causes of unplanned shutdown of nuclear power plants, which in turn can become accidents and near misses, and identifying nuisance alarms is a key step to improve the performance of alarm systems. Currently, nuclear power plants generate a large amount of application data and simulator training data, which can be used to build more flexible, dynamic models and predict alarm behavior during alarm floods. By studying the dynamic evaluation and prediction methods of alarm chattering during the alarm flow, focusing on the processing of nuisance alarms, the advanced intelligent alarm technology based on machine learning is established, the alarm chattering is quantified, the machine learning model is provided, and the machine learning model is trained and the performance of the model is evaluated by using knowledge. It is of great significance to research that the intelligent alarm management scheme based on machine learning can dynamically identify and predict chattering alarms and better help operators make accurate decisions.

Keywords: Smart alarms,Nuisance alarm,Machine learning,Chattering alarm

1. INTRODUCTION

In the field of industrial security, alarm systems have always played a crucial role. With the integration of the alarm system with the Distributed Control Systems(DCS), the flexibility of the alarm system is greatly improved. However, this also poses some problems. The ease with which new alarm devices are deployed in modern industry has led to the installation of a large number of alarm devices, many of which have not been properly rationalized [1]. This results in operators dealing with a large number of alarms, many of which are useless nuisance alarms that do not provide new information or require action.

The presence of nuisance alarms makes it possible for operators to miss critical alarms during the alarm flood as they are overwhelmed by a flood of useless alarms, and identifying and managing nuisance alarms becomes crucial. Unreasonable alarm management [1] is one of the main causes of unplanned shutdowns of nuclear power plants, as well as possible accidents and near-misses. To address this issue, standard manuals for alarm management have been published and advanced alarm management techniques have been developed to reduce nuisance and quantifychattering, redundancy, and correlation between alarms [2][3].

In the digital age, technologies such as Industry 4.0 and the Internet of Things are profoundly impacting the entire industrial sector [4]. A lot of data can be stored in cloud services and server farms, but it's not easy to extract information and gain knowledge from that data. This results in a large amount of data being stored but lacking further analysis. In this context, machine learning techniques have attracted attention in the field of industrial security. These algorithms can learn from historical data, and the knowledge acquired during the learning phase can be used to predict future events [5]. Therefore, machine learning provides an effective means to develop dynamic and flexible models based on historical data and establish machine learning-based alarm management schemes [6].

This study focuses on the prediction and identification of chatting alarm types in nuisance alarms, using a dynamic chatting index to quantify the chattering. Based on chatting assessment data, three machine learning algorithm models are employed for identification and prediction. By comparing the comprehensive results of the performance metrics of these algorithms, the most suitable algorithm model for chatting alarm prediction and identification is selected.

2. ALARM SYSTEM

An alarm system is a collection of hardware and software that detects an alarm state, communicates an indication of that status to the operator, and records changes in the alarm state [7]. During abnormal events, the alarm system allows the operator to be informed of abnormal process conditions or equipment failures. A well-designed and reliable alarm system is a must to ensure the safety and stability of the power plant.

2.1. Nuisance Alarm and Alarm Flow

Alarm flow refer to periods of frequent alarm activity. Hundreds (or even thousands) of alarms can occur during alarm flow; In this case, it is very likely that important alarms will be missed. The duration of the alarm flow is variable; Starts when the alarm rate exceeds "10 alarms/operators per 10-minute interval" and ends when the alarm rate returns to normal ("less than 5 alarms/operators per 10-minute interval"). The alarm system should not be flooded more than 1% of the total time.

Nuisance alarms [8] are alarms that are excessive, unnecessary, or not restored after the operator has taken action. Nuisance alarms do not provide any new information to the operator, or there is no possible action to resolve the alarm. As a result, it distracts the manipulator. The number of nuisance alarms must be regularly assessed and reduced to ensure that the alarm system is stable and efficient.

2.2. Chattering Assessment

As mentioned earlier, a key step in improving the performance of your alarm system is to eliminate nuisance alarms. The main type of nuisance alarm is chattering alarm, and the main index parameter used to quantify chattering is the chattering index (Ψ). In this paper, the main work is to dynamically identify and predict chattering alarms.

A chattering alarm is an alarm that repeatedly switches between active and inactive states over a short period of time. In a matter of hours or even minutes, chattering alarms can be triggered hundreds of times, which is difficult for operators to manage. The rule of thumb for identifying chattering alarms is to have more than 3 alarm logs in a minute. However, this definition is relatively vague and lacks standards or guidelines for quantifying alarm chattering behavior. A method based on travel distribution to quantify alarm chattering [2] includes the creation of a binary alarm database, travel calculation, stroke distribution calculation, discrete probability function calculation, and chattering index calculation.

Through the above process, the perturbation index for each unique alarm that occurred during the study period will be obtained. The chattering index of an alarm can be interpreted as "the average frequency of reporting of an alarm, assuming that an anomalous event lasts for an indefinite period of time", and it has the following properties:The rule of thumb for determining chattering behavior is to have 3 or more alarm records (from the same alarm) in a minute

$$\Psi > 3 \text{ alarms/min} = 0.05 \text{ alarms/sec} \quad (1)$$

$\Psi \in [0,1]$ (the closer to 1, the more disturbed the alarm becomes); Ψ is measured in alarms/second.

3. MACHINE LEARNING

Machine learning involves machines acquiring knowledge from the past (i.e., past data) and performing multiple tasks (e.g., prediction, classification, pattern recognition, etc.) using all the techniques (i.e., algorithms) of the acquired knowledge [5][10]. There are three main categories of machine learning algorithms:1. Supervised learning; 2. Unsupervised learning; 3. Reinforcement learning.

In this study, only supervised learning was used, which can be divided into two broad categories: regression and classification. Since the purpose of this study is to classify alarms (i.e., alarms "will show chattering" or "will not propagate chattering"), only classifications are used.

3.1. Machine learning algorithm models

In the field of machine learning, the main goal of the algorithm is to discover a function (denoted as (f)) that effectively maps the relationship between the input features and the output labels. By "model", we refer to the method of constructing this function and its key features. By training the data, the optimal parameter settings of the model are found. A well-trained model is able to accurately translate inputs into desired outputs.

3.2. Machine-learning algorithm performance

In the evaluation phase of machine learning algorithms, it is important to quantify model performance through various indicators that are based on different types of prediction results: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). For example, for a dichotomous problem, "1" means "alarm will show chattering" and "0" means "alarm will not show chattering".

The confusion matrix helps to outline these results, showing the performance of the model in various kinds of predictions. For example, the model may have correctly predicted "Alarm will show chattering" once (TP = 1) and correctly predicted "Alarm will not show chattering" 91 times (TN = 91), with one false positive (FP=1) and seven false negatives (FN = 7).

Three metrics are widely used to evaluate the performance of algorithms [6]: Accuracy, Precision, and Recall .

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

The quantification of model performance depends not only on the accuracy (the proportion of total correct predictions to total predictions), but also on the precision (the proportion of correctly predicted positives) and recall (the proportion of correctly detected actual positives). In particular, accuracy alone may be misleading performance assessments. For example, even with an accuracy of 92%, the model may still be unsatisfactory if most of the 'alarm will show chattering' events are not detected.

Precision and recall often counterbalance each other, improving one may harm the other. For example, increased recall may substantially reduce precision and vice versa. In specific applications, such as alarm systems, recall may be more critical than precision, as missing true alarm events can have serious consequences. By adjusting the threshold of the model, an appropriate balance can be found between precision and recall to accommodate a specific scenario.

4. INTELLIGENT ALARM OF NUCLEAR POWER PLANTS BASED ON MACHINE LEARNING

The first step in developing a data-driven machine learning algorithm is to build a database of features and labels. The selection of the most relevant features is largely empirical and often requires trial and error. Next, the database is divided into two parts; The first part is used to train the model, and the second part is used to evaluate the model. Then, select the model and transform the features to match the model needs. Finally, the model is trained and evaluated. In this chapter, it is clarified how to obtain the binary alarm database from the original alarm database. Then, the basic idea of chattering index based on alarm big data is described, and the methods and steps of obtaining relevant indicators are clarified.

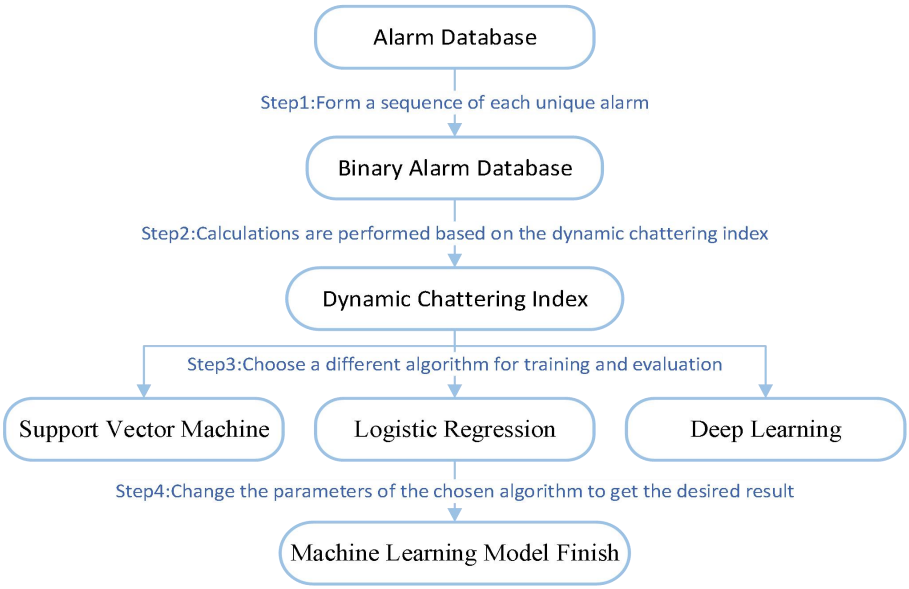

Figure 1. Diagram of the steps

4.1. Binary Database

A binary database is an efficient data structure for representing alarm events, where each event is uniquely determined by three attributes: a timestamp, a label name, and an alarm identifier. The combination of the label name and the alarm identifier is called a unique alarm, and each unique alarm can be represented by a binary sequence. In this sequence, each element corresponds to a timestamp, with "0" indicating that no alarm has occurred and "1" indicating that an alarm has occurred, and this sequence is usually sampled at a frequency of one element per second.

The process of building a binary database is divided into three main steps:
1. Identify all unique alarms during the observation period from the alarm database and store these alarms in a vector, each element of this vector represents a unique alarm, assuming that a total of N unique alarms are identified.
2. Establish a time vector at 1-second intervals from the beginning to the end of the observation period, assuming that this time vector contains M elements, i.e., the observation period has a total of M seconds.
3. For each unique alarm identified, create a binary vector of the same length as the time vector. If a unique alarm occurs at a certain point in the time vector, the corresponding binary vector element is marked as "1", otherwise it is marked as "0".

Table 1. Binary data table

Time stamp	3ABP1002KA-	...	3ABP1003KA-	...	3ABP1009KA-
2024-01-09 14:01:01	1	...	0	...	0
2024-01-09 14:01:09	1	...	0	...	0
2024-01-09 14:01:21	1	...	0	...	0
2024-01-09 14:01:33	...	...	1	...	...
2024-01-09 14:01:39	1	...	0	...	0
2024-01-09 14:01:41	0	...	0	...	1
2024-01-09 14:01:43	0	...	1	...	0
...	...	...	...	...	...

In this way, we can obtain an M×N matrix that represents the binary sequence of all the unique alarms that occurred during the observation period. The first column of the matrix corresponds to the time vector, the remaining columns correspond to the binary sequence of each unique alarm, and the first row represents all unique alarms. By removing these rows and columns containing only zeros, a more compact matrix is obtained, assuming that p rows containing only zeros and q columns containing only zeros are removed, then the size of the reduced matrix is (M-p)×(N-q). This reduced binary database provides a more manageable and analytical basis for further alarm technical analysis.

4.2. The dynamic chattering index, (Ψ_D)

The calculation of the chattering index takes five steps. The first step is to create a binary database. This section describes steps 2 to five:
Step 2: trip length (r) calculation, a trip is "the time difference between two continuous alarms on the same label (in seconds)". Therefore, whenever a "1" is found in the binary representation of the alarm, the trip is the time between the "1" and the next "1" in the binary sequence.
Step 3: Travel distribution (RLD) calculation, intuitively, if most alarms have short strokes, the alarm will show chattering behavior. To quantify this view, the number of repetitions per trip——n_r (e.g., 3 seconds) in the alarm binary sequence can be calculated. For each unique alarm in the binary database, calculate the alarm count associated with the trip (obtained in step 4), generate a list of the trip and associated alarm counts, and each unique alarm has its own list.
Step 4: Discrete probability function (DPF) is calculated, and the stroke distribution data can be normalized to obtain a discrete probability function. Normalization using the factor $\Sigma_r n_r$ yields that:

$$P_r = \frac{n_r}{\sum_r n_r} \qquad \forall r \in N \tag{5}$$

Where: r = a trip; n_r = alarm count associated with trip r; Pr indicates the probability of n_r. Therefore, for each unique alarm, a z-dispersion probability function is calculated if the alarm is associated with a z-specific stroke.
Step 5: chattering index (Ψ) calculation
The chattering index of a unique alarm can be calculated by adding the product between each probability function (P_r) and the inverse of the stroke (r):

$$\Psi = \sum_{r \in N} P_r \frac{1}{r} \tag{6}$$

The countdown of the stroke is used as a weighting function to emphasize attention on alarm counts with short strokes. Therefore, its chattering index will be calculated for each unique alarm. If the chattering index is greater than the threshold value (e.g. 0.05 alarms / s, see Equation (1)), the alarm is marked as a chattering alarm.

The dynamic chattering Index (Ψ_D) was developed to implement a machine learning method that can assess the chattering in real time. The method can predict whether the alarm will show the chattering behavior every time the alarm occurs. Machine learning algorithms rely on historical data to link each alarm ("1" in a binary sequence) to a measure that quantifies the likelihood of a chattering in the future. The dynamic chattering index provides input to the machine learning algorithm by generating after the specific manipulation on the raw data during the training phase.

The core idea of a dynamic chattering index is to calculate a chattering index every time a "1" in a binary sequence is found to generate a number of chattering equal to its number of times for each unique alarm. This is different from the initial chattering index (calculated only once for each unique alarm). Defining the time frame of the "future" is critical to predicting whether an alarm displays a chattering. It is suggested that the future time should not be too short to ensure the reliability of statistical tools, nor too long to avoid the loss of dynamic features.

The calculation process of dynamic chattering index is iterative, and n is the total number of alarms present in Table 3, then requiring n-1 iterations. The i-time (general) iteration contains the following steps:
(1) Select Index =i (for example, the first iteration is Index = 1);
(2) Select all events that occur within 20 minutes after the occurrence of Index =i in the binary sequence (e. g., the blue bracket indicates the first iteration, the green bracket indicates the second iteration, the red bracket indicates the ninth iteration, and so on);
(3) Calculate the "regular" chattering index by using the partial binary sequence obtained in step (2). The index is stored in the vector of the same dimension as the binary sequence, such that it is stored in position i.
Finally, for $\forall$ i $\in$ N (i <n), repeat steps (1) to (3).

In this way, each time a "1" is found in the binary sequence, a chattering index is calculated, which represents the tendency for the alarm to disturb within one hour after its occurrence. This chattering index is called the dynamic chattering index because it dynamically evaluates the chattering every time an alarm occurs.

The vectors containing the dynamic chattering index were integrated with binary sequences as shown in Table2 .which requires multiple iterations, in each iteration, the part of the binary sequence within a certain time (such as 20 minutes) is selected, and the conventional chattering index is calculated. This is done repeatedly until the dynamic chattering index for all alarms is calculated.

Table 2 Table of dynamic chattering index calculation

Index	Time stamp	Z	Ψ_D	Label
1	2024-01-09 14:01:01	1	0.075111919	1
2	2024-01-09 14:01:09	1	0.062639899	1
3	2024-01-09 14:01:21	1	0.055742088	1
4	2024-01-09 14:01:39	1	0.055835354	1
5	2024-01-09 14:01:48	1	0.000000000	0
6	2024-01-09 14:31:35	1	\	\
...	...	...	...	...
n	Time stamp n	1	\	\

4.3. Evaluate the machine-learning algorithms

The performance of the model will be evaluated based on the ability of the trained model to correctly predict the labels. Each row in the evaluation database (i.e., features associated with the occurrence of a specific alarm) is entered into the model. Ultimately, the model provides the predicted labels. However, for the original output of the model, it predicts not label =0 or label =1, but the probability of label =0 and label =1, such as a probability vector like [0.88,0.12], where 0.88 is the probability of label =1,0.12 is the probability of label =0, and then, comparing the probability with the threshold, converting the original output into labels, finally returned by the program.

By default, the probability threshold level is 0.5 (i.e., if the probability of a label is greater than 0.5, it is predicted). By comparing the predicted labels to the real labels, the software calculates and displays a range of performance indicators (e.g., accuracy, recall, precision, etc.). Thresholds can be defined according to the actual requirements to evaluate the performance of the model [6].

This study employs three different algorithms: Support Vector Machine (SVM), Deep Learning, and Logistic Regression. The reasons for selecting these algorithms are that SVM and Deep Learning have wide applications and are suitable for many specific machine learning scenarios. Logistic Regression was chosen because the focus of this research is mainly a binary classification problem, for which this algorithm may have a comparative advantage.The following is a demonstration of the results of three different algorithm models with a probability of 0.5.

Table 3 Performance metrics of different models

	accuracy	recall	precision
SVM	0.75	0.73	0.75
Deep learning	0.68	0.49	0.48
Logistic regression	0.96	1.00	0.95

According to the analysis of the results, logistic regression performed well in terms of accuracy, precision, and recall, indicating that it was able to effectively identify most positive samples while maintaining a low false positive rate. In contrast, the SVM and deep learning models performed significantly worse in the simulation, and all three indicators were significantly lower than the logistic regression algorithm.

The poor performance of support vector machines can be due to several reasons. SVMs perform well in the face of high-dimensional feature spaces, but SVMs may not be able to effectively distinguish samples if the amount of data is limited or the feature space is not complex enough. The poor performance of a deep

learning model may be related to the sample size of the data. Deep learning models often require large amounts of data to train to capture enough features and patterns. If the sample data volume is insufficient, the model training will be insufficient, resulting in degraded deep learning performance. So in the end, we chose to use a logical algorithm and test against the probability thresholds.

4.4. Logistic regression results are displayed

Conduct research on changes in probability thresholds. When the probability threshold changes between 0 and 1, you can see that the accuracy,precision, and recall also change.

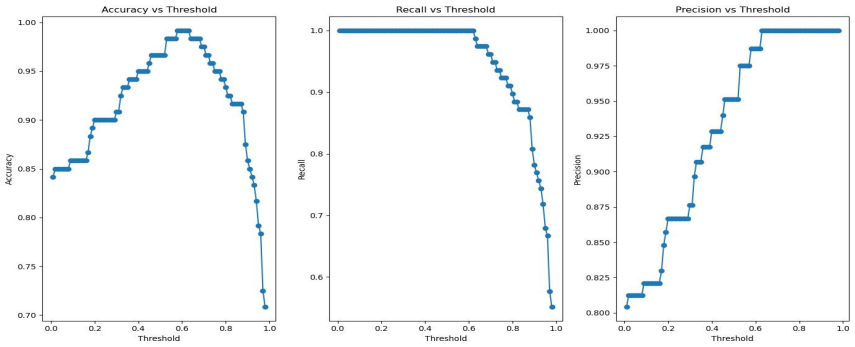

Figure 2. Graph of the change in thresholds

Enter the results into excel and perform simple processing on excel, you can see that as the threshold increases, the accuracy rises first and then decreases, while the recall rate keeps decreasing, and the accuracy generally increases, but there will be fluctuations.

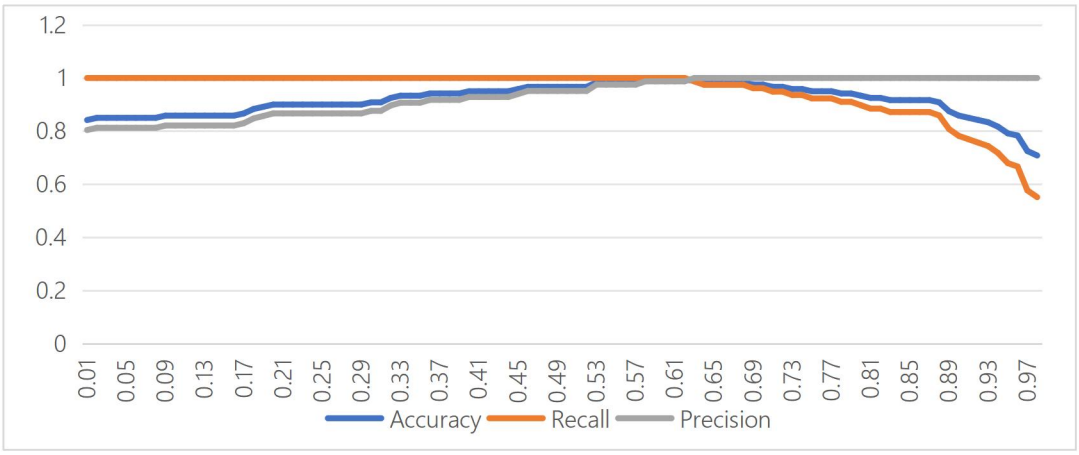

Figure 3. Performance metrics change by threshold

The trend of accuracy is easy to understand, because when the threshold is 0 or 1, there is only one item for TP and TN, reaching a maximum in the middle. As the threshold increases, TP decreases, but TP+FN is more stable, so the recall rate decreases. The main trouble with the change in accuracy is that the change of TP+FP with the threshold is inconsistent with TP, so it is more complicated.

It can also be seen from the figure that the model has the best effect when the threshold is 0.55~0.67.

5. CONCLUSION

This paper focuses on exploring a machine learning technique designed to predict nuisance alarms in nuclear power plants. This method establishes a dynamic chattering evaluation index to train and validates the model by analyzing the relevant database of nuclear power plant. The efficacy of the model is measured by its accuracy in predicting the chattering alarm.

Based on the results presented by the three indicators: accuracy, precision, and recall of the algorithm model, choosing the logistic regression algorithm for the prediction and identification of nuclear power disturbance alarms meets the requirements. By adjusting the threshold of the logistic regression algorithm according to the performance indicators, it can be found that the comprehensive effect of the model's prediction and identification is best when the probability threshold is between 0.55 and 0.67. Therefore, in subsequent work, the logistic regression algorithm can be selected, with the probability threshold adjusted to around 0.60 for further testing experiments.

Given that chattering alarms and alarm flooding are key challenges facing contemporary alarm systems, the model presented in this study provides a flexible and dynamic solution to these problems.In the current stage, the database of actual nuclear power plant data has not been established. In the future, the database of actual nuclear power plant data will be established according to the needs of the project, and the work will be carried out using the methods studied in the paper.

Acknowledgements

This study was supported by Shenzhen Natural Science Foundation- the Stable Support Plan Program (No. 20220811103029001) titled “Research on Risk-informed Dynamic Autonomous Operation Decision-making Technology Based on Real-time Situation Awareness for Nuclear Power Plants”, GuangDong Basic and Applied Basic Research Foundation (No. 2022A1515110545) titled “Research on Performance-based Real-time Risk Monitoring and Dynamic Prediction Technology for Nuclear”, LingChuang Research Project of China National Nuclear Corporation (No. CNNC-LCKY-202263) titled “Research on the reproduction technology of the dynamic interactive operation scenario for the nuclear power plant based on the data mining”, the Shenzhen Pengcheng Peacock Plan Talent Project (No. 827-000885) titled “Visualization technology for dynamic interactive operation scenarios of nuclear power plants based on big data and multi-agent”, Shenzhen Science and Technology Innovation Commission Key Technical Project (JSGG20210713091539014), GuangDong Natural Science Foundation-General Program (No. 2024A1515012727) titled “Research on Dynamic Risk-informed Multi-attribute Objective Collaborative Decision-making Mechanism for Unmanned Small Modular Reactors under Uncertainty Factors”, and Shenzhen Science and Technology Innovation Commission Key Technical Project (JSGG20210713091539014). Professor Sijuan Chen is the corresponding author for this paper.

References

[1] Kondaveeti, S. R. et al. (2010) Graphical representation of industrial alarm data, IFAC Proceedings Volumes (IFAC-PapersOnline). IFAC. doi: 10.3182/20100831-4-fr-2021.00033.
[2]]Kondaveeti, S. R. et al. (2013) ′Quantification of alarm chatter based on run length distributions′, Chemical Engineering Research and Design. Institution of Chemical Engineers, 91(12), pp. 2550–2558. doi: 10.1016/j.cherd.2013.02.028.
[3] Yang, F. et al. (2012) ′Improved correlation analysis and visualization of industrial alarm dat′, ISA Transactions. Elsevier Ltd, 51(4), pp. 499–506. doi: 10.1016/j.isatra.2012.03.005.
[4] Liu, J. et al. (2018) ′Artificial intelligence in the 21st century', IEEE Access, 6(April), pp. 34403–34421. doi: 10.1109/ACCESS.2018.2819688.
[5] Brink, H., Richards, J. and Fetherolf, M. (2016) Real-World Machine Learning. Manning Publications. Available at: https://books.google.no/books?id=DoQAswEACAAJ.
[6] Tamascelli N. A machine learning approach to predict chattering alarms[D]. NTNU, 2020.
[7] Management of Alarm Systems for the Process Industries, document ISA 18.02, ISA (International Society of Automation), Durham, NC, USA, 2009.
[8] ANSI ISA (2016) ′ANSI/ISA−18.2−2016 Management of Alarm Systems for the Process Industries′, ANSI ISA.
[9] B. Yang, J. Li, C. Qi, H. Li and Y. He, ″Novel correlation Aanalysis of alarms based on block matching similarities,″ Industrial & Engineering Chemistry Research, vol. 58, no. 22, pp. 9465-9472, 2019.
[10] Jordan M I, Mitchell T M. Machine learning: Trends, perspectives, and prospects[J]. Science, 2015, 349(6245): 255-260.