



DEIDS: a novel intrusion detection system for industrial control systems

Haoran Gu¹ · Yingxu Lai^{1,2} · Yipeng Wang^{1,2} · Jing Liu^{1,2} · Motong Sun¹ · Beifeng Mao¹

Received: 20 July 2021 / Accepted: 13 January 2022 / Published online: 7 February 2022
© The Author(s) 2022

Abstract

Owing to the development of industrial production, the hidden danger in industrial control systems (ICSs) has considerably increased, causing challenges in traditional safety defense methods. The combination of machine-learning or deep-learning algorithms and intrusion detection systems (IDSs) has become the mainstream method for solving this problem. However, these methods depend on a massive amount of high-quality attack traffic data, which cannot be obtained easily owing to the independence and unique characteristics of ICSs. In this study, we apply the reconstructed convolutional neural network and a data expansion algorithm named CenterBorderline_SMOTE (**CB_SMOTE**) to an IDS and propose data expansion intrusion detection system (**DEIDS**). The **DEIDS** is an end-to-end detection model that learns representative attack features from raw traffic and classifies them in a unified framework. Moreover, we adopt the classification activation map structure, which can deeply mine the potential characteristics of traffic and enhance the effectiveness of attack features. While enhancing the data quality, we introduce the designed **CB_SMOTE** algorithm into **DEIDS** to expand the data and solve the problem of insufficient attack data in the system. Our comprehensive experiments on different open datasets indicate that **DEIDS** achieves an excellent performance (97% detection accuracy) and outperforms the state-of-the-art methods. The experimental results also show that our method has high efficiency and high accuracy in processing ICSs datasets.

Keywords Industrial control system · Intrusion detection system · Data augmentation · SMOTE

1 Introduction

1.1 Motivation

Owing to the development of modernization, industrial production and control are gradually gaining attention. Intrusion detection is particularly important for the security of industrial control [1–5]. The intrusion detection system (IDS) is a commonly used defense method in cybersecurity and industrial control systems (ICS) research [6]. Recently, IDS uses machine learning [7–9] or deep learning as the main approach in ICS security. Using machine or deep

learning [10–12], the IDS can obtain accurate detection results and provide a more accurate alarm for attacks under different protocols [13–17]. This protects ICS safety.

However, the industrial control network is different from the traditional network, and each factory's network is relatively independent and closed. Owing to this, the data on the industrial control network are not universal. The conventional traffic generated in the actual industrial production process is larger than the actual attack traffic. This imbalanced distribution data set seriously affects the lack of training of machine and deep learning methods used in intrusion detection. This results in a decline in the accuracy of the intrusion detection. Although devices such as the testbed at Sandia National Laboratory can alleviate this difficulty, the number and types of attacks generated by the testbed are limited. Therefore, it is challenging to meet the IDS training and testing requirements. Unlike the traditional network traffic, the traffic protocol format in the industrial control network is relatively fixed. It is approximately composed of fixed-length fields such as Modbus.

✉ Yingxu Lai
laiyingxu@bjut.edu.cn

¹ Faculty of Information Technology, Beijing University of Technology, Beijing 100124, Beijing, China

² Engineering Research Center of Intelligent Perception and Autonomous Control, Ministry of Education, Beijing 100124, Beijing, China

Therefore, different attacks are highly similar in feature engineering based on machine learning IDS, which interferes with machine training. In addition, because more manual extraction processes are used to establish feature engineering, it is easy to consider the prior knowledge of researchers and interfere with the model's classification process of the model [7–9]. Therefore, increasing the quantity and quality of the ICS attack traffic data has become a focus issue and attracted both industrial and scholarly attention.

1.2 Limitation of the prior state-of-the-art methods

Combining machine or deep learning algorithms and IDS becomes the mainstream method for protecting ICS. The intrusion detection models in IDS include feature engineering and model training. Both of these two processes require a certain number of high-quality datasets. However, the current ICS data sets are unevenly distributed owing to the lack of attack traffic data, which cannot meet this requirement. This affects the training effect and detection accuracy of the IDS.

To solve the difficulty of the imbalanced data sets, several methods have been proposed in the past few decades. These are approximately divided into two categories: algorithm and data levels [18]. For the methods proposed at the algorithm level, the goal is to reinforce the existing classifier learning algorithms on a small number of categories. Gu et al. [19] and Khabsa et al. [20] proposed an improved support vector machine to allocate more penalty weights by assigning misclassified minority instances during the training. This is to improve the accuracy of minority class categories when dealing with unbalanced data sets. However, this method does not consider the different contributions of the minority class examples when learning decision boundaries, leading to over-fitting, which is affected by noise. Based on the aforementioned methods, it can be observed that the algorithm-level methods for solving the problem of imbalanced data greatly depend on the classification ability of the classifier. Moreover, they do not fundamentally solve the problem of the misclassified minority class. Furthermore, the decision boundary is also generated based on the existing data and cannot represent the actual data distribution.

Another solution to such problems is the use of the methods based on the data-level. The solution attempts to rebalance the category distribution by resampling the given imbalanced data. In Tao's investigation, compared with the algorithm-level methods, data-level methods seemed more common for imbalanced datasets [21]. It was reported that the balanced datasets obtained by resampling can be used for learning by any classification algorithms. However, in

many practical applications of imbalanced datasets, the minority class samples are often unavailable in the learning phase. The lack of minority class samples makes it difficult for the traditional resampling methods to deal with imbalanced datasets. Furthermore, it does not enable the IDS to achieve better training results. This makes the industrial equipment get attacked owing to minority class samples such as attack traffic data. Therefore, how to use the existing ICS imbalanced datasets to improve the detection accuracy of IDS to protect the industrial production process better has become a limitation that researchers need to solve.

1.3 Proposed approach

To solve the aforementioned limitations, in this study, we propose an end-to-end system known as data expansion intrusion detection system (**DEIDS**) for protecting ICS' safety. As shown in Fig. 1, the structure of **DEIDS** includes format conversion, data expansion, and intrusion detection modules. First, the format conversion module adjusts the raw attack traffic data's formation to suit the IDS' model training and feature extraction. Then, we use a discriminator to determine whether each category's traffic data are in a balanced state and whether the amount is sufficient. The data expansion module contains a redesigned convolutional neural network (CNN) and a data expansion algorithm. The redesigned CNN learns and extracts features to enhance the discrimination of the attack traffic. The algorithm uses enhanced features extracted by the redesigned CNN to expand to obtain the generated attack traffic tensors. The traditional machine or deep learning model is used in the intrusion detection module, and the generated attack traffic tensors are used for the training. The trained model has a strong ability and robustness to detect ICS attacks. The flexible architecture increases the generalization of **DEIDS**.

In this study, the data expansion module receives insufficient samples judged by the discriminator. This module has two submodules, the feature extractor and the sample expander. The insufficient samples in the converted format are first transmitted to the feature extractor. More accurate data features can be extracted from the traffic data using the feature extractor. Then, a new algorithm named CenterBorderline_SMOTE (**CB_SMOTE**) is designed and proposed in the sample expander. This expands the data features in the feature extractor and strengthens the boundaries of the data features. Finally, this submodule makes and obtains the feature engineering by using the expanded data features that benefit the training of the intrusion detection model. To design the algorithm, we refer to the interpolation idea of SMOTE method and improve this method. We abandon the K-nearest neighbor

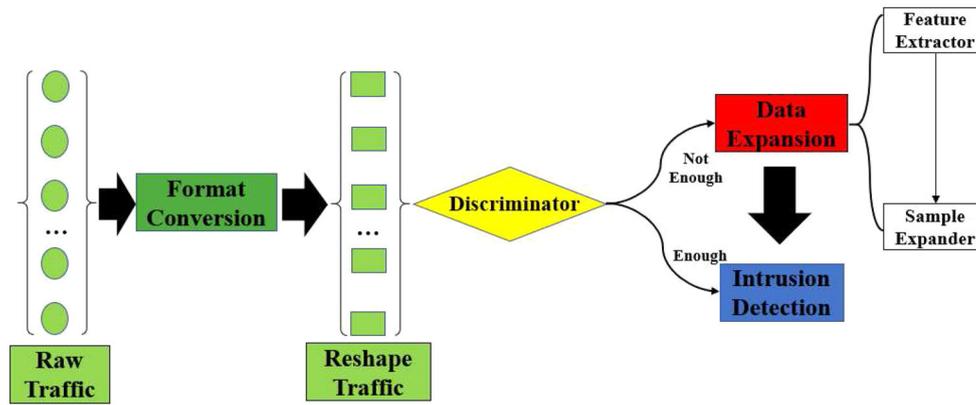


Fig. 1 Simple flow of DEIDS

idea, solving the limitation that it is easy to cause blindness in selecting data expansion location and data volume when the distribution differentiation of the positive and negative samples is not high. The boundary distribution is enhanced to solve the distribution marginalization, and it uses the existing minority class data to ensure that the newly generated data are within the decision boundary. The schematic diagram of the distribution of the generated data for different strategies is shown in Fig. 2. The method we use can ensure that the generated data are always kept on the positive side. Large data will not be generated on the boundary or in areas where it is difficult to demarcate, resulting in misclassification of the model.

1.4 Novelty and advantage of our approach

In this study, the main novelties and advantages of the proposed approach are as follows:

- Enhanced feature extraction** The traditional feature extraction depends entirely on the establishment of feature engineering by artificial operation or deep learning methods such as CNN. This study designs a feature extractor that modifies and redesigns the structure of CNN and uses reconstructed CNN to extract more accurate features. The advantage of this method is that it can improve the accuracy of feature extraction, avoiding unnecessary interference caused by human disoperation or prior knowledge.

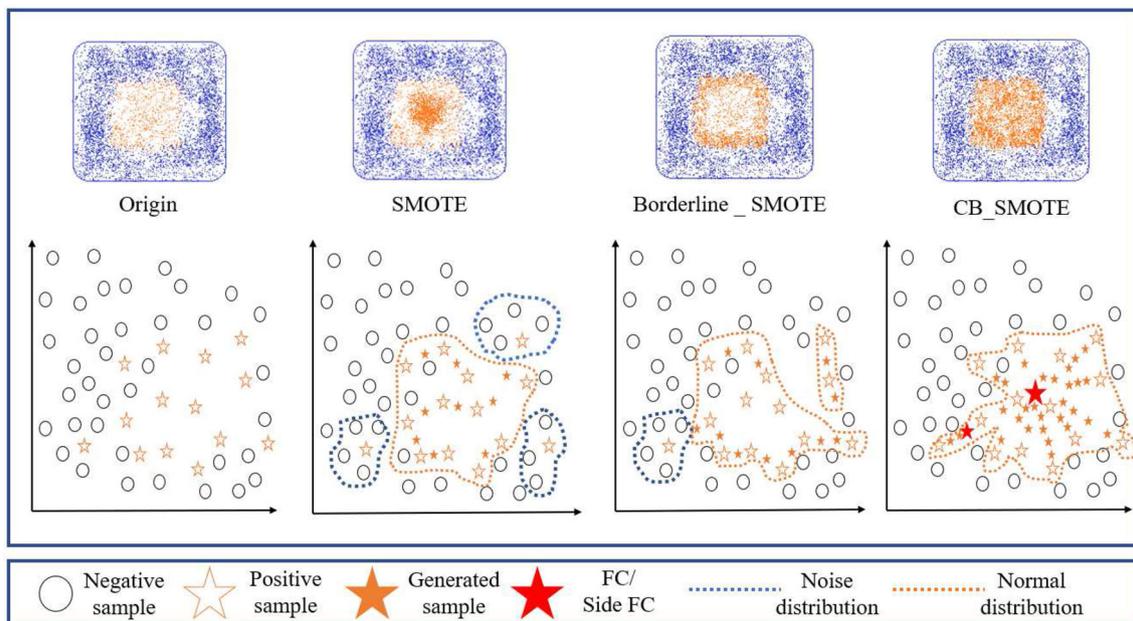


Fig. 2 Schematic diagram of the distribution of the data generated by our method and the traditional method

- *Efficient data expansion* The traditional method based on SMOTE uses the K-nearest neighbor. However, the idea K value is difficult to determine, the amount of calculation is significant, and it can be easily affected by noise problems. This study designs and proposes a new data enhancement method based on SMOTE–CB_SMOTE. Using the class boundary sample as the classification boundary, the seed sample is obtained by directly comparing the sample with the fitting center distance and aggregation degree. The new sample is synthesized on the connection between the seed sample and the fitting center to realize the oversampling strategy.

1.5 Key contributions

The main contributions of this study are as follows.

- **DEIDS** can accurately identify ICS attacks and obtain high-precision intrusion detection. The system is suitable for intrusion detection based on the ICS. Compared with the traditional intrusion systems, **DEIDS** has more extensive applicability and practicability. In the case of insufficient attack samples, it can use the internal algorithm to generate data to improve the detection accuracy. Therefore, it can protect industrial devices from attacks better.
- We present a novel algorithm based on SMOTE combined with a data's *fitting center* (*FC*) known as **CB_SMOTE**. The algorithm can solve imbalanced ICS datasets by producing attack traffic data. The algorithm can also deal with the boundary and strengthen the boundary's distinction to enhance the whole dataset to improve its quality. The datasets expanded by this algorithm are suitable for various types of machine or deep learning intrusion detection models. The intrusion detection model trained by this dataset can have a strong robustness and detection accuracy.
- The experimental results show that **DEIDS** has a very high detection accuracy. It can extract more valuable and accurate features, which provides excellent features. Compared with the traditional over-sampling methods, the expanded algorithm used by **DEIDS** can acquire higher quality attack data than the conventional generated data method. The detection accuracy of the intrusion detection model trained by the open ICS datasets is over 97%. The generated datasets can train the intrusion detection model and make the models obtain a more powerful detection ability to protect the ICS better.

2 Related work

Random over-sampling is a non-informative method, which rebalances the class distribution by randomly copying the minority class samples [22]. The disadvantage is that because of the repetition of the information in the training set, the accurate replication of substitution will lead to the over-fitting of the subsequent supervised classification algorithm. To overcome this defect, Chawla et al. [23] proposed an information-based over-sampling method known as the synthetic minority over-sampling technique (SMOTE). The algorithm generates new minority class instances by interpolating the k-nearest minority class neighbors. This method can provide more helpful information for classification than the random over-sampling, because it (the proposed method) creates new artificial minority class instances rather than simply copying the original minority class instances. To avoid the SMOTE algorithm's sample coverage problem, Han et al. [24] proposed a Borderline-SMOTE algorithm. This algorithm searches for "dangerous" samples in a small number of classes and generates new samples based on these samples. Therefore, He et al. [25] constructed a distribution function of newly generated samples based on the degree of "dangerous" samples to determine the number of unique samples generated based on each "dangerous" sample. Jo et al. [26] implemented a clustering-based sampling algorithm known as the cluster-based over-sampling (CBO), which is suitable for cases in which multiple disconnected aggregation points exist in the class distribution. Liu et al. [27] introduced the concept of the class average distance and proposed an unbalanced dataset learning algorithm known as DB_SMOTE using the center of gravity of sample data. This method is simple to use and adapts the use of datasets with clear boundaries.

However, the method based on SMOTE faces the following difficulties. First, as an interpolation method, SMOTE can quickly interpolate low-dimensional data to achieve the purpose of over-sampling. However, it is not suitable for high-dimensional data. Second, the SMOTE cannot guarantee the accuracy and effectiveness of the generated data owing to the K-nearest neighbor. This makes the selected interpolation space impure, making part of the generated data noisy.

Because of the advancements in machine and deep learning methods, samples are oversampled to enhance the data by combining algorithm-level methods. GAN [28–30], flow-based model, and other deep learning methods integrate the idea of algorithm-level methods and oversample the samples to enhance the data [31–34]. However, in this type of method, the oversampling speed is low because of a pre-training process. Moreover, the introduction of deep

learning brings uncertainty in the convergence process; therefore it is difficult to obtain samples quickly, and it is not suitable for the process of IDS intrusion detection.

In this study, we propose a novel IDS known as **DEIDS**. It involves an oversampling technique named **CB_SMOTE** based on SMOTE (a redesigned CNN detection model to make feature engineering), and a deep learning intrusion detection model based on the characteristics of the industrial control network. **CB_SMOTE** depends on the whole imbalanced dataset to achieve all the information from the data distribution in the dataset. Contrary to SMOTE, based on the traditional interpolation, our method abandons the idea of K-nearest neighbor, which makes our method to focus more data selection than blind over-sampling data. In addition, the algorithm and the redesigned CNN detection model can deal with higher-dimensional data such as network traffic. Compared with the deep learning methods (such as GAN) which require training, **CB_SMOTE** directly interacts with the dataset to generate data without training. This drastically speeds up the efficiency of the use of IDS.

3 Process of DEIDS

Owing to the characteristics and protocol limitations of the ICS, researchers have been faced with problems caused by the imbalance of attack data, such as low quality of attack data and severe shortage. There is no intrusion detection system and method suitable for ICS to deal with the insufficient samples and low quality. To solve these difficulties, we design a novel IDS known as **DEIDS**. It can utilize the imbalanced dataset of ICS for the targeted oversampling. Then, the attack dataset generated by expansion and the majority class samples are combined into a training set. This is transferred to the intrusion detection system model for full training.

The whole system has three modules: the format conversion, data expansion, and intrusion detection modules. The format conversion module is responsible for processing ICS attack traffic data, reshaping the data into the format of the model designed in this study. It uses a discriminator to determine whether the amount of attack traffic data meets the intrusion detection module's training requirements. The data expansion module comprises two submodules: feature extractor and sample expander. The two submodules cooperate to improve the ICS attack traffic dataset to meet the needs of the IDS from the view of feature extraction and data expansion.

The structure of the system designed here is shown in Fig. 3. First, the system transforms the data format. The attack traffic data are converted to tensors needed for model training, and the system will use these tensors

created by the attack data to train the CNN. They will then be passed through the discriminator for the first time to judge whether each attack category's amount is in a balanced state. Based on the assumption that it belongs to equilibrium distribution attack data, the number of each category's attack data is enough. In such a case, it will be transferred to the intrusion detection module for training and detection. Nonetheless, if there are few attack data in one type or some types, the tensors created by the attack data will be passed to the data expansion module. Using our designed classification importance discrimination module (CIDM), we can extract the relevant attack details of the attack traffic samples and construct detailed features of the attacks based on their classification importance. Then, using our proposed algorithm **CB_SMOTE**, entering these attack features can increase the number of minority class samples' attack features. After the generated feature is obtained, the feature is restored based on the standard tensor format to generate the feature engineering. Finally, the expanded attack samples train the intrusion detection model, effectively solving the low-quality and insufficient data and improving the detection accuracy. After the operation of the whole system, researchers can expand the imbalanced ICS attack datasets effectively and accurately. More accurate attack feature sets can be extracted for intrusion detection training by our system to make the intrusion detection results obtained more accurate.

3.1 Feature extractor based on the classification importance discrimination module

Because the artificial operation in feature engineering will interfere with the feature extraction, which affects intrusion detection models' detection accuracy, we design and use the feature extractor to extract the data features instead of the artificial feature engineering. It uses the reconstructed CNN's powerful feature extraction ability to extract features strongly associated with the attack type.

Figure 4 shows the procedure of the traffic flow tensors' generation process. In this study, the conversion rules are formulated based on the characteristics of industrial control network traffic. First of all, based on the conversion rules, this study intercepts the first fixed number of bits of each traffic packet, saves the original capture sequence, and integrates it into a flow. Second, the hexadecimal data in the flow payload are converted into decimal data and are sent into a flow tensor line. These tensors contain the payload or a flow of original ICS traffic for some time. Padding data are used to make up the insufficient number of bits to ensure that the length of each flow is the same. Finally, the flow tensor line is reshaped into the form of a flow tensor square.

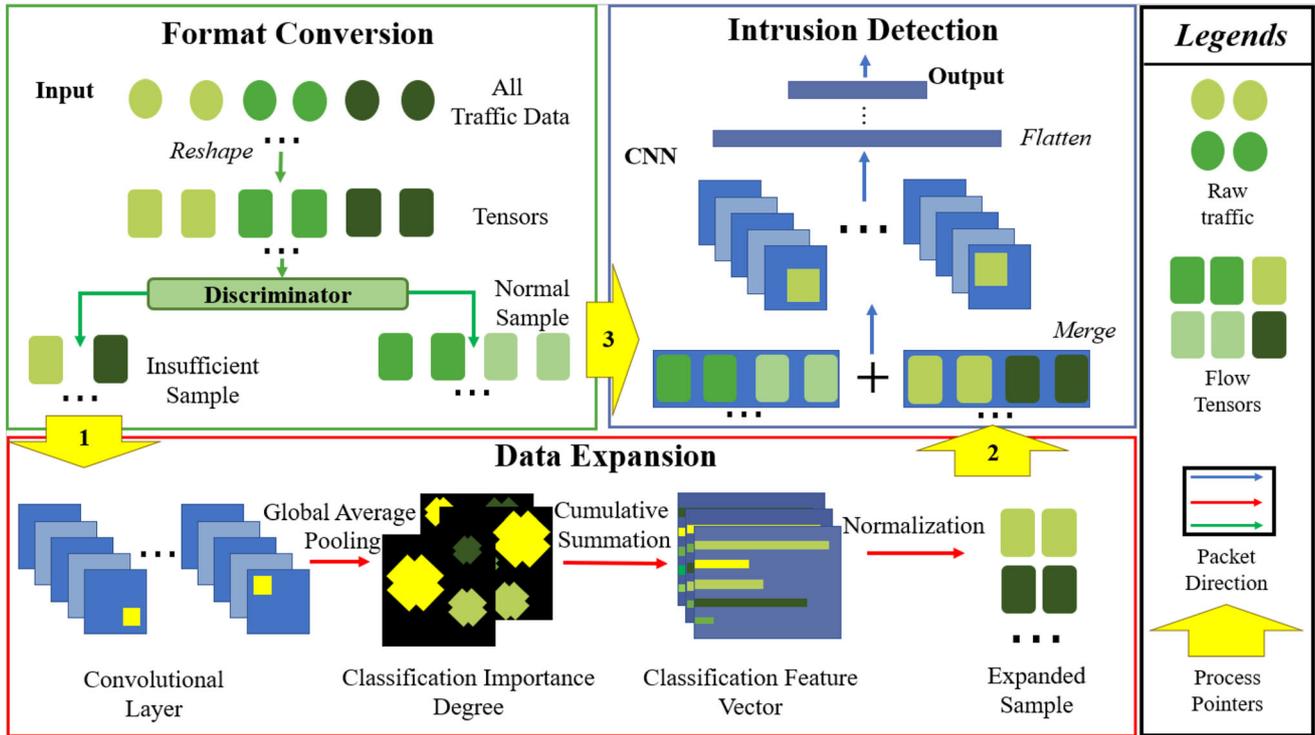


Fig. 3 Structure of the DEIDS

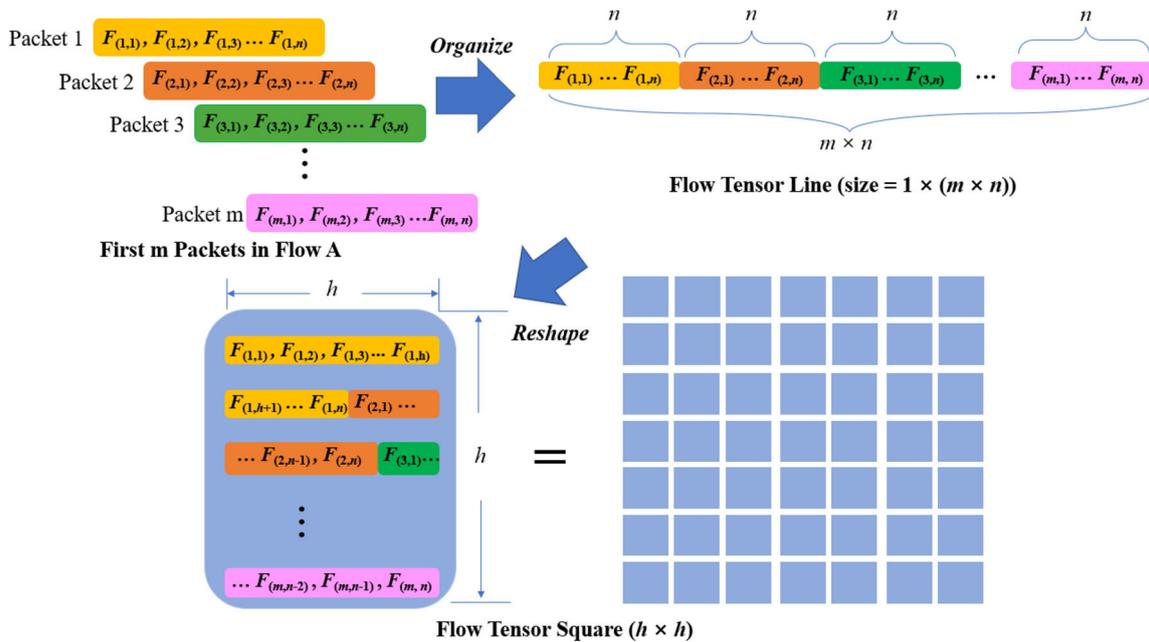


Fig. 4 Generation process of the traffic flow tensor

The traditional CNN model generally divides into convolutional and fully connected layers. The convolutional layer is responsible for extracting pixel region in the image,

and the fully connected layer performs reshaping analysis based on the pixel region to complete the classification. Zhou et al. [35] have shown that the pixel area retained

after several training iterations in the convolution process is the target part that can help the classification to extract the features. However, the fully connected layer negatively affects the retained information to affect the feature extraction adversely. They perceive that CNN has a significant positioning ability by using the global average pooling (GAP) layer, and the data information contained in the pixels position will not be negatively affected. This indicates that it will not cause a loss of the original image.

Thus, to extract the attack features more accurately, we draw lessons from Zhou’s method [35] to replace the fully connected layer in the traditional CNN model with the weight layer to eliminate the negative impact of the fully connected layer on the model feature extraction process. The reconstructed CNN uses a combination of the GAP and weight layers. The weight layer can reasonably complete the fully connected layer’s classification task, and the data will not be affected.

The reconstructed CNN structure is shown in Fig. 5. To make this method suitable for the network attack flow, the existing methods need to be adjusted. Although the prior knowledge possessed by researchers can help reduce the weight of the matrix and extract more valuable attack features, for machines, the effect of classification depends more on the feature attributes of the data. This attribute is likely to contradict or conflict with human knowledge. Therefore, for the classification model, prior human knowledge is intrusive and misleading. Moreover, Zhou’s

method selects several features with a high degree of activation as classification features. This is not desirable, because this is bound to cause the loss of information. In our paper, we designed a module known as the CIDM. This module replaces the taking of only the most essential feature details in Zhou’s method. However, it establishes a large matrix to preserve the feature classification importance degree in each flow. After passing through the CIDM, each flow can accurately capture every feature that plays a decisive role in the classification. Each type of attack can also use the CIDM to observe the precise details of the attack, and the characteristic information and location play the role of the attack.

Because of the CIDM, the CNN’s feature extraction steps have been adjusted to a certain extent. As shown in Eq. (1), the flow data’s tensors filtered by the CIDM can output a set of data features T_o^n . The feature set T_o^n represents the n th feature T in the O -dimension. The CNN provides each feature a corresponding weight w_o^c during the training process. This weight w describes the importance of the O -dimension’s features when the flow tensor is classified as C . Thus, when the image is classified into category C , the data feature set filtered by CIDM can be classified as D_c based on the corresponding importance provides to each feature during the CNN’s training process. We used this importance to select the features that play a more decisive and important role in the classification, as shown in Eq. (1).

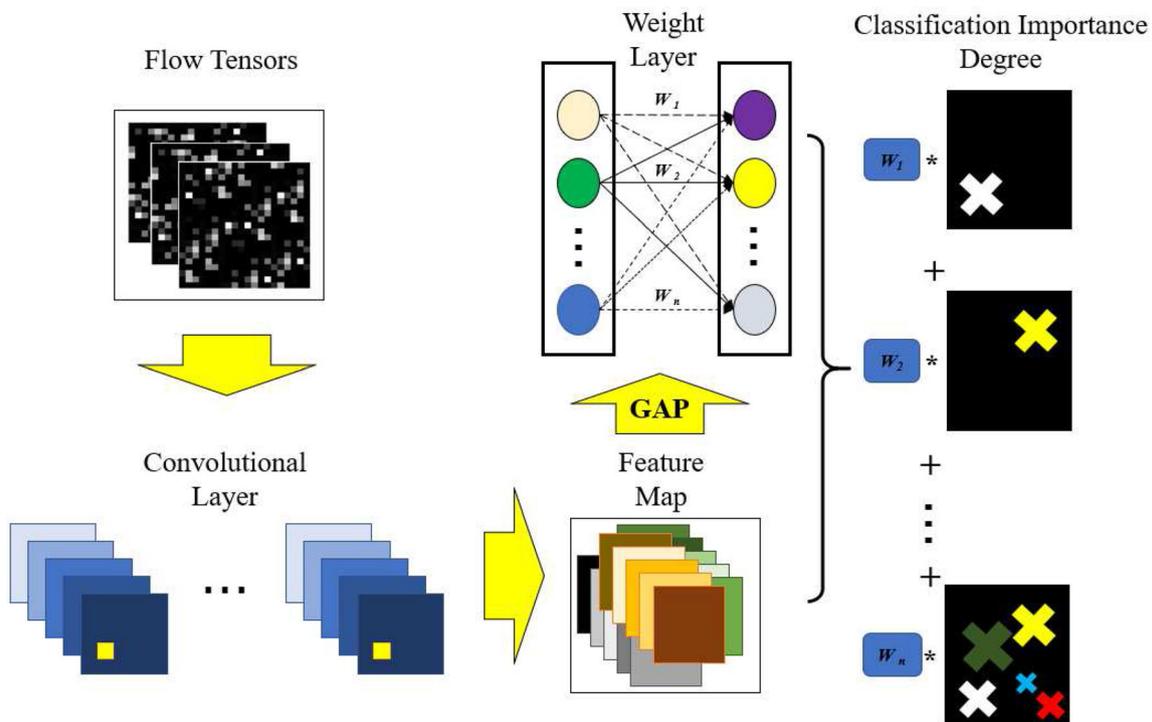


Fig. 5 Schematic diagram of the reconstructed CNN structure

$$T_o^n = \begin{bmatrix} T_o^0 & T_o^1 & \dots & T_o^n \\ T_o^0 & T_o^1 & \dots & T_o^n \\ \vdots & \vdots & \ddots & \vdots \\ T_o^0 & T_o^1 & \dots & T_o^n \end{bmatrix} \tag{1}$$

$$D_c = \sum_o w_o^c \sum_n F_o(T_o^n) = \sum_o \sum_n w_o^c F_o(T_o^n)$$

$F_o(T_o^n)$ denotes the importance of the degree of feature T_n in dimension O of the flow tensor. We performed identical deformation similarly. The highest convolution layer’s importance tensors of the detection model were not globally pooled. All the tensors were weighted and summed based on the weights extracted from the weight matrix. The classification importance degree D_c corresponding to the original tensors can be obtained.

Because the introduced GAP layer can be used to reduce the dimension of each dimension’s important tensors, when it is finally necessary to classify the flow tensors, the probability of being classified to C is expressed by Eq. (2).

$$P_c = \frac{\exp(D_c)}{\sum_n \exp(D_c)} \tag{2}$$

P_c represents the probability that the flow tensor is classified as C , and n represents the total number of categories owned. To judge whether the tensor classification is correct, we retained the distribution of each class probability to calculate various classification metrics subsequently.

3.2 Sample expander based on CB_SMOTE algorithm

After all the attack flow tensors pass through the redesigned CNN, the CNN can accurately extract each attack feature that plays a decisive role in classifying the attack types. Therefore, each attack type’s full feature importance degree can be obtained by accumulating the classification importance degree of all the flow tensors in each category. Then, we select the essential attack features extracted by CNN under each category as the feature engineering of this attack type. Because the quality and quantity of feature engineering cannot meet adequate training requirements and classification, it is necessary to expand and enhance feature engineering effectively. In this study, a module known as sample expander is designed for this purpose. For this module, we propose a method based on the data weights and distribution boundaries known as **CB_SMOTE**. This method can effectively expand and enhance insufficient attack samples and take the corresponding measures for boundary samples to avoid cases where boundary samples are easily misclassified as noise. This method can simplify the expansion steps and only

expand samples in the key positions that affect model detection.

In the sample expander with the **CB_SMOTE** algorithm, we use the feature engineering of each attack type obtained by the feature extractor to make a targeted expansion. $S = \{F_i, i = 1, 2, \dots, n\}$ is used to build a set of attack details for each category of data for the attack details extracted by the CNN. F_i represents the feature engineering matrix of attack i , which the importance obtained by the CIDM screens out. Then, in all the flow tensors of each attack type, the features of the corresponding position are regarded as the feature engineering of this flow based on the filtered importance and are saved to F_i . The following matrix can be observed, where n represents the number of features selected by each flow tensor and m denotes the number of flow tensors under this attack type. To expand the feature set of attack details of the minority classes, we designed the **CB_SMOTE** method. The method extracts the attack details from the same class’s attack detail set and locates the exact position’s eigenvalues in f_i . Then, it extracts the eigenvalues of the same position and calculates the average value. The average value in the data space is recorded as the *Fitting Center (FC)* and is calculated as shown in Eq. (3), where m here has the same meaning as the m in F_i .

$$F = \begin{bmatrix} f_1^1 & f_1^2 & \dots & f_1^n \\ f_2^1 & f_2^2 & \dots & f_2^n \\ \vdots & \vdots & \ddots & \vdots \\ f_m^1 & f_m^2 & \dots & f_m^n \end{bmatrix} \tag{3}$$

$$FC = \frac{1}{m} \sum_{j=1}^m f_m$$

The average distance from the features of the exact location in the attack detail of this type to the FC point is recorded as the average *Fitting Distance (FD)*. This distance reflects the degree of aggregation in the details of this type of attack. The smaller the value is, the more compact is the distribution’s degree. Otherwise, the distribution is sparse. The calculation is shown in Eq. (4).

$$FD = \frac{1}{m} \sum_{j=1}^m Dis(f_m, FC) \tag{4}$$

The over-sampling strategy’s key is to find the basic features and iteratively generate new features. During sample classification, samples at the boundary are the most prone to classification errors. Thus, we must pay special attention to more important weights to boundary samples. Here, we define to facilitate the description, and this definition is recorded as the *Basic Feature (BF)*. The BF refers to the feature in which the distance from the same position feature to the FC is greater than the FD . The method of

calculation of BF is shown in Eq. (5), where T_i represents the feature set of the i th attack.

$$BF = \{T_i \mid Dis(f_m, FC) > FD\} \tag{5}$$

Subsequently, all the BF s in the exact location are found to build a candidate set. The FC is designated as the reference point to avoid adding too much interference into the generated feature. A line segment is formed using the attack features and reference points in the candidate set. We generate new attack features on the line segment to ensure that the generated attack features are located inside the correct class. Based on the basic principle of the SMOTE algorithm, we combine the algorithm with the *Generated New Feature (GNF)* as shown in Eq. (6):

$$GNF = S_i + (S_i - FC) \times r \tag{6}$$

where S_i belongs to the candidate set sample and r is a random number taken between $[0,1]$.

Based on the previous description, the greater the distance between the attack features of the candidate set and the features of FC is, the more likely these features are to be misclassified. Thus, the number of features generated for such a sample needs to be increased accordingly. This is suitable for improving the classification model’s accuracy. According to the Euclidean distance calculation

Finally, we can obtain the distribution function P as shown in Eq. (7).

$$P_i = \frac{Dis(S_i, FC)}{S}, \sum_{i=1}^k P_i = 1 \tag{7}$$

We can obtain the number of new samples generated for each candidate attack feature by multiplying the probability of the feature distribution by the total number of features.

The ICS attack data characteristics and the correlation between information points are used to simplify the sample expansion process. This indicates that samples can be expanded for a specific location pixel of an individual attack detail. The corresponding position pixel information can be expanded based on the change in the first position. This method retains ICS attack data characteristics, and it is not easy to produce a large number of error data when expanding the sample.

The following shows the implementation of the **CB_SMOTE**. Assume few class sample sets $DS = \{(f_i, N_i), i = 1, 2, 3, \dots, n\}$, where i represents the number of attack detail categories for the samples, f_i represents the collection of specific attack details, and N_i represents the number of f_i category attack details.

Algorithm 1 CB_SMOTE

Require: Input parameters DS , balance factor σ , number of samples for majority classes maj , number of minority samples min

Ensure: Output the expanded DS

- 1: The total number of generated samples $num \leftarrow (|maj| - |min|) * \sigma$
 - 2: **while** $j = 1, j < i$ **do**
 - 3: The number of large class distribution of the generated samples $num_j \leftarrow \frac{N_j}{\sum_j N_j}$
 - 4: $FC \leftarrow \frac{1}{m} \sum_{j=1}^m f_m$
 - 5: $FD \leftarrow \frac{1}{m} \sum_{j=1}^m Dis(f_m, FC)$
 - 6: $BF \leftarrow \{T_i \mid Dis(f_m, FC) > FD\}$
 - 7: $P_i \leftarrow \frac{Dis(S_i, FC)}{S}$ and $(\sum_{i=1}^k P_i = 1)$
 - 8: **end while**
 - 9: **while** $k = 1, k < int(P_k * num_j)$ **do**
 - 10: **while** $l = 1, l < n$ **do**
 - 11: $GNF \leftarrow S_i + (S_i - FC) \times r$
 - 12: **end while**
 - 13: $f_j \leftarrow GNF \cup f_j$
 - 14: **end while**
 - 15: $DS \leftarrow f_j \cup DS$
-

method, the distance $Dis(S_i, FC)$ from each feature to the FC ’s feature can be obtained. Therefore, the sum of Euclidean distances from each attack detail to FC is obtained by accumulation. The synthesis is recorded as S .

In the algorithm, the *int()* function is rounding up. The balance factor is used to determine the total number of generated samples, which can be initially set to one here based on the requirement. This is to ensure a balanced

relationship between the over-sampled dataset and majority of sample sets.

3.3 Sample expander based on boundary enhancement CB_SMOTE

Because the proposed method generates new features by the random position on the line segment, if the CB_SMOTE cannot develop enough new features near the boundary, it does not enhance the quality of the boundary sample. To perform the aforementioned operation, we adopt the design idea of improving the boundary attack data. The boundary data and their nearest original boundary data are connected. The generated data between the boundary data and the nearest neighbor data are defined as the sample set to be generated. Therefore, the expanded dataset is calculated using the aforementioned CB_SMOTE algorithm. Thus, we can obtain sufficient generated data to enhance the boundary attack data quality and avoid misclassification.

boundary midpoints of the dataset to be expanded), and the result compared with the *FC*, the Difference Value (text-biDV), is expressed by Eq. (8):

$$DV = |RC - FC|, \quad (8)$$

$$\text{where } \left(RC = \frac{X_{\text{boundary}L} - X_{\text{boundary}R}}{2} \right)$$

where $X_{\text{boundary}L}$ and $X_{\text{boundary}R}$ are represented as the boundary point at the left and right boundaries, respectively. The distance of *DV* between *RC* and *FC* can be calculated using Eq. (8). The existence of *DV* indicates that there is an error between the actual and estimated values. Therefore, the generated sample cannot be well represented as the distribution of the original data. We use the sample expander specially designed for the boundary in this study to enhance the boundary of data distribution. The implementation of the algorithm used by the sample expander, which is specially designed for the boundary, is described later.

Algorithm 2 boundary enhancement CB_SMOTE

Require: Input parameters *GNF*

Ensure: Output the expanded *DS*

- 1: From *GNF* to find the boundary sample X_{S1} , X_{S2} and the nearest boundary sample X'_{S1} , X'_{S2}
 - 2: Calculate *RC1*, *RC2*, *FC1*, and *FC2* from Eq.(3) to (8)
 - 3: Calculate *DV1* , and *DV2* according to Eq.(8)
 - 4: **while** *DV1* > 0.01 **do**
 - 5: *FC1* is used as *FC*, and X_{S1} and X'_{S1} are used as the boundary, to combine the generated sample of the sample set to be generated with *FC* to calculate *FD* and *BF*
 - 6: Generate boundary generation new samples *GNF*
 - 7: Calculate *FC1* according to Eq.(3)
 - 8: **end while**
 - 9: **while** *DV2* > 0.01 **do**
 - 10: *FC2* is used as *FC*, and X_{S1} and X'_{S2} are used as the boundary, to combine the generated sample of the sample set to be generated with *FC* to calculate *FD* and *BF*
 - 11: Generate boundary generation of new samples of *GNF*
 - 12: Calculate *FC2* using Eq.(3)
 - 13: **end while**
 - 14: $f_j \leftarrow GNF \cup f_j$
 - 15: $DS \leftarrow f_j \cup DS$
-

In this study, the boundary dataset to be expanded is selected based on the previous paragraph's description. The *Real Center (RC)* of the boundary attack dataset (the two

If the *DV* needs to be further expanded, new data are calculated and generated by applying the aforementioned algorithm. Each time the data are generated, the *FC* is

calculated, and a new *FC* is obtained. Then, the distance between the *FC* and *RC* is compared until the difference is less than or equal to a particular threshold value. This can effectively solve the classification accuracy of the boundary features.

4 Experiment

In this study, relevant experiments were designed for **DEIDS** and **CB_SMOTE** to verify whether the proposed system and method can effectively improve the quantity and quality of ICS attack data to improve the detection accuracy of the intrusion detection system.

4.1 Dataset

There are two industrial control datasets selected in this study. One is the safe water treatment (SWaT) system [36], and the other is the test-bed designed by Rodofile [37] (referred to as the S7 dataset in this study), which simulates the actual mining and refining of the industrial process.

The SWaT system is an operating test-bed for water treatment and a small-scale version of a large-scale modern water treatment plant. The water purification process in SWaT is divided into six sub-processes. Experiments are performed on the SWaT system to study the network attack and the corresponding system response. In the process of the SWaT data collection, four major types of attacks occurred, including 36 sub-type attacks. Table 1 indicates the types of attacks, sensors, executors that may be attacked in different water treatment processes and distribution proportion of each type of attack. There are 449919 pieces of data in the experiment, of which 395298 and 54621 are standard and attack data, respectively. To facilitate the test, 10000 pieces of data are randomly selected based on the imbalance ratio of each sub-type attack. Among them, 8000 and 2000 are the training and test data, respectively.

In this study, another set of industrial control network datasets designed by Rodofile was used for experiments (S7 dataset). This simulates the actual mining and refinery of industrial processes. As shown in Table 2, we select approximately 10h of data in the dataset, including 30 cycles and 55 attacks (excluding flooding attacks), and the

original normal traffic is approximately 40000. We extract the original payload of the industrial control network traffic from the S7 protocol packet. The original data include several complete communication processes between the man-machine interface and primary control programmable logic controller(PLC). The attack process in this dataset is to send attack commands to the master PLC by connecting the attacker to the industrial control network, as shown in Table 3.

In the experimental setup of this study, the S7 dataset retains all the 55 pieces of the data and replays them based on their respective attack types. A total of 1200 attack data are obtained and are proportionally divided into 1000 and 200 as training and test sets, respectively. Then, the data extracted from the standard data and 1000 pieces of the attack data are integrated into 4000 and 800 pieces of flow as the training and the test sets, respectively.

4.2 Model processing and training

To solve the uneven distribution of industrial control samples, CNN must extract accurate attack details as features. Thus, we reconstruct and improve several classical CNN methods. First, the fully connected layer in the traditional model is wholly removed. The highest convolution layer in the model is directly connected to the weight matrix through the global average pooling operation. The format of the weight matrix is as follows:

$$weight = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix}$$

where *m* denotes the number of dimensions or filters of the highest convolution layer and *n* is the final classification number.

The weight matrix can replace the classification effect of the fully connected layer. It will not affect the location of the target in the convolution layer, similar to the fully connected layer’s operation. The CNN’s editability is used to adjust the convolution part’s size parameters to adapt to the traffic tensors developed here. The convolution process’s stride is controlled to one to reduce the information loss and further ensure the model’s effectiveness. The CNN

Table 1 SWaT attack type, attack distribution and location information table

Attack type	Process	Attack sensors	Attack executer	Attack times
SSSP	P1-P5	LIT101, MV101	MV101, P101, P102	26
SSMP	P1, P2, P5	AIT202, P201	MV101, P101, P102	4
MSSP	P6	AIT504, P501	–	2
MSMP	P6	AIT504, P501	–	4

Table 2 Types and distribution of abnormal attack behavior in the S7 dataset (section)

Time	Attack	Time	Attack	Time	Attack	Time	Attack	Time	Attack
2980	1	6198	5	8280	11	14517	5(F)	17157	8
3420	5	6237	6	8489	3(F)	15457	8(F)	17216	8
3999	8	6796	8	8959	7(F)	15861	6	20166	8
4231	3	6826	8	9648	10(F)	16575	8	21402	8
4787	7	7057	3	9917	2(F)	16605	8	21948	6
5413	10	7146	3	10672	6(F)	16655	8	21952	8
5659	1	7460	7	12053	8(F)	16753	8	22943	2
5718	2	7648	7	12680	1(F)	17087	8	22966	4

Table 3 Situation of the attack types in the S7 dataset

Label	Attack name	Control domain	Address	Flow type
1	Conveyor belt off	0 × 83	0 × 0330	CBOF
2	Conveyor belt on	0 × 83	0 × 0331	CBON
3	Conveyor belt-gate change direction	0 × 83	0 × 0332	CB-GCD
4	Conveyor belt-gate reset	0 × 83	0 × 0334	CBR
5	Water tank off	0 × 83	0 × 0328	WTO
6	Water tank on auto	0 × 83	0 × 0329	WTOA
7	Water tank on manu	0 × 83	0 × 032a	WTOM
8	Reactor off	0 × 83	0 × 0320	ROF
9	Reactor on	0 × 83	0 × 0322	RON
10	Change upper-threshold	0 × 83	0 × 0040	CU-T
11	Change lower-threshold	0 × 83	0 × 0060	CL-T
12	Global reset	0 × 83	0 × 0339	GR
13	Emergency stop	0 × 83	0 × 033b	ES

Table 4 Structure of the different models with CIDM

CNN-CIDM		ResCNN-CIDM	
Conv1	Conv2d-96, s = 2, kernel_size = 7, padding = 1 MaxPooling	Conv1	Conv2d-32, s = 1, kernel_size = 3, padding = 1, bias = False 3*3, MaxPooling, s = 2
Conv2	Conv2d-256, s = 2, kernel_size = 3 MaxPooling	Conv2	[3*3 32] [3*3 32]*2
Conv3	Conv2d-384, s = 2, kernel_size = 2	Conv3	[3*3 128] [3*3 128]*2
	Global average pooling	Conv4	[3*3 256] [3*3 256]*2
	Weight layer		Global average pooling
			Weight layer

structure designed in this study is shown in the following table. We retain the original convolution process and replace the fully connected layer of the CNN model with CIDM to make the reconstructed CNN-CIDM. The CNN-CIDM under different structures is tested, and the test results are shown in Table 4.

Based on the confusion matrix definition, we use the calculated values of *ACC*, *FNR*, and *FPR* as the measurement metrics.

$$ACC = \frac{FN}{TP + FN} \tag{9}$$

$$FNR = \frac{FN}{TP + FN} \tag{10}$$

$$FPR = \frac{FP}{FP + TN} \tag{11}$$

$$TPR = Recall = \frac{TP}{TP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$F1 - value = \frac{(1 + \beta) * Recall * Precision}{\beta * Recall + Precision} \tag{14}$$

$$RERR = 1 - \frac{1 - ACC_{new}}{1 - ACC_{pre}} \tag{15}$$

where *TP* is the number of normal traffic that is classified as usual, *FP* is the number of abnormal traffic that is classified as usual, *TN* is the number of normal traffic classified as abnormal, and *FN* is the number of abnormal traffic classified as abnormal. The *F1-value* combines the output results of precision and recall. The value ranges from zero to one, where one and zero represent the model’s best and worst outputs, respectively. The value of β is usually one.

To better measure this method’s advantages, we use the relative error rate reduction(*RERR*) to assess the improvement in the accuracy after using our approach to enhance the dataset [38]. *ACC_{pre}* denotes the accuracy without using our method, and *ACC_{new}* donates the metric value using our form. The *RERR* can effectively highlight the improvement or reduction in intrusion detection accuracy before and after using our method to show our approach’s advantages more quantitatively and intuitively.

Table 5 shows the training and testing results of two original models and two reconstructed models with CIDM. Each model was trained for 20 epochs, and each model was tested 15 times. The results of Table 5 show that the model’s detection accuracy after reconstruction is better than that of the model before reconstruction, and the accuracy of different models and the *F1-value* of the final model have reached a very objective value. This shows that the model has been thoroughly trained and can extract relatively reliable features. Therefore, we choose CNN-CIDM and ResCNN-CIDM models as our experimental model.

4.3 Experiment and results

We choose the *S7* dataset as the experimental sample selected by the parameters. Table 5 reveals that there is not

much difference between the CNN-CIDM and ResCNN-CIDM in the process of feature extraction. Therefore, we used CNN-CIDM as the experimental model with an excellent comprehensive training effect.

4.3.1 Verification of the influence of the boundary distance difference on data expansion

To find the best threshold setting of the boundary *FC* and *DV*, the following experiments are designed. The *DV* is set to 0.01, 0.02, 0.05, 0.1, 0.2, and 0.5, respectively. Based on these six different *DV* values, six different boundary-enhanced generated datasets can be obtained. The dataset is used to train the model. Then, cross-verification is performed using the actual attack test set for detection. The detection results shown in Table 6 are obtained.

The results in Table 6 show that the overall detection accuracy, *FPR* and *FNR* of the model without data enhancement are weaker than those of the model with boundary processing. This indicates that the sample has been further enhanced, and the quality of the attack data has been drastically improved compared to the one before the expansion. The boundary enhancement strategy can help the model to classify the attack data accurately. Then, by observing the different boundary detection results, when we set the *DV* to 0.01, the model’s detection accuracy, *FNR*, *FPR*, and the ability to identify the attack categories all achieve optimal results. Therefore, in all the experiments, *DV* is set to 0.01.

4.3.2 Verification of the universality of the CB_SMOTE algorithm

The improvement in accuracy usually does not exhaustively illustrate the effectiveness of the method. Usually, the model’s overall accuracy is likely to be high; however, the precision of single-class attacks is not high. To further verify the method’s effectiveness, we use **CB_SMOTE** to expand the *S7* data separately for each attack category, and the other 12 categories that have not been expanded are replayed and integrated. Then, we merge the expanded attack data and the other twelve kinds of attack data with

Table 5 Accuracy, detection rate, FNR, and FPR of different models by using the *S7* dataset

Model	Accuracy /%			Recall /%			F1-value
	Max	Avg	Min	Max	Avg	Min	
CNN	96.072	94.172	92.266	96.061	94.163	92.259	0.9417±0.019
CNN-CIDM	96.497	94.562	92.500	96.451	94.557	92.381	0.9456±0.021
ResCNN	97.608	95.437	93.322	97.607	95.356	93.212	0.9540±0.022
ResCNN-CIDM	97.702	95.544	93.966	97.679	95.512	93.856	0.9553±0.021

Bold text emphasize the superiority of this experimental data over the other comparison groups

Table 6 Influence of setting different DV values on the testing metrics

DV type	Detection rate /%	FNR /%	FPR /%	ACC /%
Without boundary enhancement	80.87	1.79	1.55	97.51
DV=0.01	87.28	1.17	0.86	99.09
DV=0.02	86.74	1.23	0.91	99.07
DV=0.05	86.33	1.25	0.92	98.91
DV=0.1	86.51	1.24	0.89	98.95
DV=0.2	85.49	1.26	1.14	98.88
DV=0.5	85.47	1.31	1.18	98.83

Table 7 Verification of the detection accuracy with different generated data

Flow type	ACC /%	FNR /%	TPR /%	Recall /%	Precision /%	Detection rate /%	F-value
CBOF	98.71	1.62	1.03	98.33	98.00	90.76	0.9816
CBON	99.21	1.27	0.87	97.78	94.46	89.29	0.9609
CB-GCD	99.05	1.32	0.92	97.78	98.00	89.71	0.9789
CBR	99.16	1.43	0.96	97.61	97.64	91.01	0.9763
WTO	99.25	1.29	0.91	98.61	96.73	91.36	0.9766
WTOA	99.37	1.23	0.89	99.00	95.07	91.09	0.9700
WTOM	99.46	1.17	0.87	99.85	94.13	93.32	0.9691
ROF	99.09	1.31	0.93	98.05	96.16	89.97	0.9710
RON	99.18	1.31	0.95	99.40	91.73	90.31	0.9541
CU-T	98.91	1.39	0.95	96.80	97.39	90.04	0.9709
CL-T	98.97	1.38	0.94	91.00	98.27	89.94	0.9450
GR	99.36	1.23	0.88	99.18	97.55	89.96	0.9836

the original training set to form a new training set for training. The actual data are used for detection. The test results are shown in Table 7.

As shown in Table 7, the method of **CB_SMOTE** is not universality. The model's detection accuracy is not negatively affected by the training with the generated data, and the *FNR* and *FPR* are reduced accordingly. Moreover, the model's recall rate and precision are also shown in this experiment, and the *F1-value* is calculated. The data show that the model trained by different generated samples can still achieve very high accuracy. In terms of the output data's feedback results, other models trained by the generated samples can stabilize the *F1-value* above 0.94. Thus, the model can achieve better classification results. There is no considerable difference between the model trained by the generated samples and the model trained using the actual data. When there is a lack of experimental data, the generation method proposed here can solve the data shortage and sample imbalance.

4.3.3 Comparison with the common expanded algorithms

To further verify the superiority of the proposed method, we compare the proposed method with SMOTE-related methods (SMOTE and borderline-SMOTE) and many extended sample methods based on the integrated

principles SMOTE-Boost and cluster-based under-sampling with boosting(CUS-Boost). The results of the comparison are presented in Table 8.

Table 8 shows that several different methods can improve the classification accuracy of other models, reduce the *FNR* and *FPR* of the model, and significantly improve the intrusion detection accuracy. Thus, several methods can make up for the accuracy of the model detection caused by the lack of information by generating a small number of samples. Comparing the two different datasets, we observe that the **CB_SMOTE** proposed in this study is feasible. Compared with the other two SMOTE-based methods, the *FNR* and *FPR* of model classification are significantly reduced and have obvious advantages in improving the accuracy of attack detection. Moreover, several methods of integrated ideas are compared. The results show that the approach we have proposed is better than the other techniques of all metrics.

As indicated by the experimental results of several detection metrics, our method has distinct advantages in both data sets. This is because our method locates the data boundaries and the areas where the data need to be generated more accurately.

Fig. 6 shows that the traditional dataset is divided into three areas: safe area, overlapping area and noise area. The most challenging place for the classifiers to distinguish is

Table 8 Accuracy, detection rate, FNR and FPR of the different methods by using the S7 and SWaT dataset

Model/Method			CNN- CIDM			ResCNN-CIDM		
			MAX	AVG	MIN	MAX	AVG	MIN
S7	SMOTE	Accuracy/%	97.98	97.29	96.57	97.36	96.67	95.47
		FNR/%	2.36	2.03	1.88	2.27	2.11	1.93
		FPR/%	2.95	2.74	2.56	1.97	1.78	1.52
		DR/%	94.68	93.44	91.58	96.73	96.50	95.13
	Borderline- SMOTE	Accuracy/%	97.93	97.33	96.81	97.50	96.72	96.68
		FNR/%	1.95	1.71	1.58	1.94	1.63	1.56
		FPR/%	1.73	1.56	1.39	1.60	1.41	1.35
		DR/%	95.87	95.41	95.28	96.55	95.84	95.53
	SMOTE- boost	Accuracy/%	97.35	96.92	96.69	97.48	96.62	95.71
		FNR/%	2.47	2.38	2.27	2.32	1.83	1.41
		FPR/%	2.99	2.74	2.48	2.63	2.27	1.82
		DR/%	94.61	93.58	93.18	97.28	96.84	95.67
	CUS- boost	Accuracy/%	97.55	97.13	96.98	96.72	95.49	95.16
		FNR/%	2.41	2.36	2.25	2.65	2.54	2.42
		FPR/%	2.98	2.85	2.49	3.08	2.91	2.87
		DR/%	94.81	93.92	93.58	96.55	95.19	94.98
	Our method	Accuracy/%	98.72	98.29	97.78	98.93	98.03	97.49
		FNR/%	1.44	1.32	1.29	1.75	1.34	1.26
		FPR/%	0.98	0.84	0.79	1.05	0.91	0.85
		DR/%	96.01	95.76	95.22	98.67	97.93	97.34
SWaT	SMOTE	Accuracy/%	99.71	95.02	88.67	97.50	94.97	89.54
		FNR/%	4.74	2.59	1.05	4.85	2.39	1.26
		FPR/%	4.75	3.66	0.95	4.83	2.28	1.20
		DR/%	98.94	94.62	88.07	97.04	94.83	89.29
	Borderline- SMOTE	Accuracy/%	96.96	94.14	92.04	97.46	95.67	93.05
		FNR/%	4.88	2.72	2.51	5.02	3.05	2.07
		FPR/%	5.07	4.36	3.55	5.75	3.63	2.16
		DR/%	95.37	93.58	91.05	97.08	94.93	92.71
	SMOTE- boost	Accuracy/%	97.32	96.54	96.08	97.92	96.52	95.81
		FNR/%	2.08	1.69	1.37	2.13	1.59	1.18
		FPR/%	3.36	3.07	2.64	3.45	3.15	2.23
		DR/%	95.55	94.62	94.35	97.51	96.45	95.51
	CUS- boost	Accuracy/%	97.01	96.42	96.00	97.22	96.35	96.06
		FNR/%	2.11	1.78	1.55	2.06	1.84	1.47
		FPR/%	3.35	3.33	3.03	3.29	3.09	2.95
		DR/%	94.88	94.57	94.32	96.92	96.27	95.54
	Our method	Accuracy/%	99.15	98.88	98.20	99.69	98.70	98.13
		FNR/%	1.74	1.37	0.37	1.66	1.45	0.75
		FPR/%	2.00	1.00	0.25	1.75	0.98	0.69
		DR/%	98.50	98.20	97.88	99.13	98.56	98.11

Bold text emphasize the superiority of this experimental data over the other comparison groups

the overlapping area rather than the noise and safe areas. Therefore, as a data-level oversampling method, data should be generated for the overlapping areas. However, the traditional method has such a problem: the SMOTE is performed on three areas. Therefore, the generated samples

are distributed in each area. Although the data generated by this method does not change the distribution of the original data, they cannot help improve the classification ability of the classifier. Consequently, borderline-SMOTE does better and plays an essential role in improving the

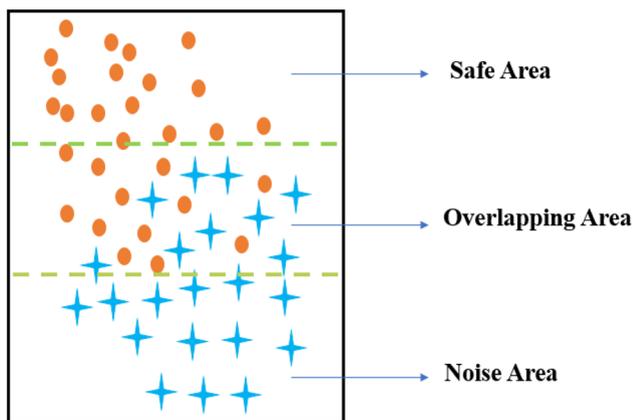


Fig. 6 Data distribution classification schematic

classification ability of the classifier. However, it only acts on the boundary of the overlapping area of the data set. This changes the distribution of the original data and negatively impacts the authenticity and integrity of the data set. Methods such as CUS-Boost and SMOTE-Boost are not oversampling method they are under-sampling methods. For datasets with the smallest amount of data, it is challenging to be effective.

Our method differs from other methods described above. First, our method is used only in the overlapping area. In contrast to the SMOTE method, our method follows the original distribution of the data set for data expansion. Second, our method generates data around the FC of the dataset, reducing the impact of the generated data on the original data distribution. Finally, our method borrows the idea of borderline-SMOTE, which is appropriately augmented for the boundaries of the dataset.

Furthermore, to further verify if our method has a strong advantage, based on the RERR metric introduced in this study, we observe the improvement in the intrusion detection model’s detection accuracy using different

methods to expand the data. As shown in Table 9, when the intrusion detection model is trained, the detection accuracy of the attack dataset generated by our method is 34% higher than the dataset expanded by other methods. The experimental results show that the method designed in this study improves detection accuracy and is suitable for different types of detection models. The proposed method can achieve more accurate and high-quality data than the commonly used sample expansion algorithm, enhancing the robustness and accuracy of the intrusion detection models.

Nowadays, more and more latest studies focus on big data industrial environment and the process of industrial production [1–5]. There are many kinds of attacks in industrial control systems and industrial control networks, but the number of attack samples is too few. In essence, this kind of problem cannot be classified as a big data problem. In other words, it is challenging to solve such problems by using the current big data method, such as intrusion detection of the industrial control system. Our proposed method solves the problem of few attack sample data such as industrial control systems and makes up for the gap of existing traditional methods.

After a comprehensive analysis of the above results, it is not difficult to find that our method has obvious advantages over the data-level methods. However, no method is perfect, and our method is no exception, which also has some limitations, which can be summed up as follows:

- *Unknown traffic attack* At present, our method can well detect known traffic attacks, but our method may not have the ability to detect unknown traffic attacks accurately. This is also one of the key points of our future research.
- *Total attack traffic* To some extent, our method can alleviate the negative impact caused by the lack of attack traffic data in IDS. But when the total amount of

Table 9 RERR of the different models by using the S7 dataset and SWaT

Model/Method		CNN-CIDM		ResCNN- CIDM	
		ACC-AVG/%	RERR/%	ACC-AVG/%	RERR/%
S7	Our method	98.29	–	98.03	–
	SMOTE	97.29	36.27	96.67	40.84
	Borderline-SMOTE	97.33	34.84	96.72	39.94
	SMOTE-boost	96.92	44.48	96.62	41.72
	CUS-boost	97.13	40.42	95.49	56.32
SWaT	Our method	98.88	–	98.70	–
	SMOTE	95.02	77.51	94.97	74.16
	Borderline-SMOTE	94.14	80.89	95.67	69.98
	SMOTE-boost	96.54	67.63	96.52	62.64
	CUS-boost	96.42	68.72	96.35	64.38

Bold text emphasize the superiority of this experimental data over the other comparison groups

attack traffic is very extreme, such as zero-shot, the expansion effect of our method is not so obvious.

5 Conclusion and limitations

In this study, we propose a new IDS named **DEIDS** to solve the problem of low precision in intrusion detection caused by the shortage and low-quality of attack traffic data in ICSs. The system mainly solves the difficulty in achieving a good intrusion detection accuracy, because the ICS attack data is imbalanced distribution. In this study, the CNN's structure is adjusted and the reconstructed CNN model can extract more accurate features of the attack data. A new data expansion method named **CB_SMOTE** is designed based on SMOTE, which is suitable for the ICS attack data. This method can effectively use the extracted accurate features of the attack data to enhance and expand the data, thereby enabling the CNN to retrain and update the proposed **DEIDS**. The generated attack data are used to train the intrusion detection model, and all the detection metrics are significant. This indicates its superiority over other traditional sample expansion algorithms. The analysis of the experimental data shows that the **CB_SMOTE** method proposed in this study can effectively solve the problems of insufficient ICS attack data and CNN learning difficulties when the attack data are insufficient. The algorithm designed in this study can be expanded and enhanced for different ICS attack datasets, and it has a certain universality as indicated by the related experiments. The experimental results also show that **DEIDS** can effectively enhance the characteristics of the attack data to improve the quality of the ICS attack data, and high-quality data can train the model more effectively. At present, we still need to conduct further research such as for defining the exact boundary of the data distribution and ensuring the integrity and authenticity of the generated data.

Acknowledgements Foundation item: National Key Research and Development Project, China (Key Technologies and Applications of Security and Trusted Industrial Control System, NO.2020YF2009500), Natural Science Foundation of Beijing Municipality, China (NO.19L2020).

Author Contributions Haoran Gu was involved in conceptualization, methodology, writing—original draft and also performed software. Haoran Gu, Yingxu Lai and Yipeng Wang performed investigation. Haoran Gu, Motong Sun and Beifeng Mao performed data processing, visualization and validation. Yingxu Lai, Yipeng Wang and Jing Liu performed supervision and writing—review and editing. Yingxu Lai and Yipeng Wang were involved in formal analysis. Yingxu Lai performed project administration.

Declaration

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Meng X, Zhang Y, Qiao J (2021) An adaptive task-oriented RBF network for key water quality parameters prediction in wastewater treatment process. *Neural Comput Appl* 33(17):11401–11414. <https://doi.org/10.1007/s00521-020-05659-z>
- Tang J, Xia H, Zhang J, Qiao J, Yu W (2021) Deep forest regression based on cross-layer full connection. *Neural Comput Appl* 33(15):9307–9328. <https://doi.org/10.1007/s00521-021-05691-7>
- Wang B, Mao Z (2020) Detecting outliers in industrial systems using a hybrid ensemble scheme. *Neural Comput Appl* 32(12):8047–8063. <https://doi.org/10.1007/s00521-019-04307-5>
- Chen Y, Ping Y, Zhang Z, Wang B, He S (2021) Privacy-preserving image multi-classification deep learning model in robot system of industrial iot. *Neural Comput Appl* 33(10):4677–4694. <https://doi.org/10.1007/s00521-020-05426-0>
- Wan J, Li J, Hua Q, Celesti A, Wang Z (2020) Intelligent equipment design assisted by cognitive internet of things and industrial big data. *Neural Comput Appl* 32(9):4463–4472. <https://doi.org/10.1007/s00521-018-3725-5>
- Wang Y, Fan K, Lai Y, Liu Z, Zhou R, Yao X, Li L (2017) Intrusion detection of industrial control system based on modbus TCP protocol. In: 13th IEEE International symposium on autonomous decentralized system, ISADS 2017, Bangkok, Thailand, March 22–24, pp 156–162. IEEE Computer Society. <https://doi.org/10.1109/ISADS.2017.29>
- Antón SD, Kanoor S, Fraunholz D, Schotten HD (2018) Evaluation of machine learning-based anomaly detection algorithms on an industrial modbus/tcp data set. In: Doerr, S., Fischer, M., Schrittwieser, S., Herrmann, D. (eds.) Proceedings of the 13th international conference on availability, reliability and security, ARES 2018, Hamburg, Germany, August 27–30, pp 41–1419. ACM. <https://doi.org/10.1145/3230833.3232818>
- Schuster F, Paul A, Rietz R, König H (2015) Potentials of using one-class SVM for detecting protocol-specific anomalies in industrial networks. In: IEEE symposium series on computational intelligence, SSCI 2015, Cape Town, South Africa, December 7–10, pp 83–90. IEEE. <https://doi.org/10.1109/SSCI.2015.22>
- Liu W, Qin J, Qu H (2018) Intrusion detection algorithm of industrial control network based on improved one-class support vector machine. *J Comput Appl* 38(5):1360–1365
- Fang Y, Ming LI, Wang P, Jiang X, Zhang X (2018) Intrusion detection model based on hybrid convolutional neural network and recurrent neural network. *J Comput Appl* 38(10):2903–7

11. Chu A, Lai Y, Liu J (2019) Industrial control intrusion detection approach based on multiclassification googlenet-lstm model. *Secur Commun Networks* 2019:6757685–1675768511. <https://doi.org/10.1155/2019/6757685>
12. Terai A, Abe S, Kojima S, Takano Y, Koshijima I (2017) Cyber-attack detection for industrial control system monitoring with support vector machine based on communication profile. In: 2017 IEEE European symposium on security and privacy workshops, EuroS&P Workshops 2017, Paris, France, April 26–28, pp 132–138. IEEE. <https://doi.org/10.1109/EuroSPW.2017.62>
13. Lai Y, Zhang J, Liu Z (2019) Industrial anomaly detection and attack classification method based on convolutional neural network. *Secur Commun Networks* 2019:8124254–1812425411. <https://doi.org/10.1155/2019/8124254>
14. Tang TA, Mhamdi L, McLernon DC, Zaidi SAR, Ghogho M, (2016) Deep learning approach for network intrusion detection in software defined networking. In: 2016 International conference on wireless networks and mobile communications, WINCOM 2016, Fez, Morocco, October 26–29, pp 258–263. IEEE. <https://doi.org/10.1109/WINCOM.2016.7777224>
15. Vinayakumar R, Soman KP, Poornachandran P, (2017) Applying convolutional neural network for network intrusion detection. In: 2017 International conference on advances in computing, communications and informatics, ICACCI 2017, Udipi (Near Mangalore), India, September 13–16, pp 1222–1228. IEEE. <https://doi.org/10.1109/ICACCI.2017.8126009>
16. Bo D, Xue W, (2016) Comparison deep learning method to traditional methods using for network intrusion detection. In: IEEE International conference on communication software networks
17. Naseer S, Saleem Y, Khalid S, Bashir MK, Han J, Iqbal MM, Han K (2018) Enhanced network anomaly detection based on deep neural networks. *IEEE Access* 6:48231–48246. <https://doi.org/10.1109/ACCESS.2018.2863036>
18. Liu S, Zhang J, Xiang Y, Zhou W (2017) Fuzzy-based information decomposition for incomplete and imbalanced data learning. *IEEE Trans Fuzzy Syst* 25(6):1476–1490. <https://doi.org/10.1109/TFUZZ.2017.2754998>
19. Gu X, Chung F, Ishibuchi H, Wang S (2017) Imbalanced TSK fuzzy classifier by cross-class bayesian fuzzy clustering and imbalance learning. *IEEE Trans Syst Man Cybern Syst* 47(8):2005–2020. <https://doi.org/10.1109/TSMC.2016.2598270>
20. Khabsa M, Elmagarmid AK, Ilyas IF, Hammady H, Ouzzani M (2016) Learning to identify relevant studies for systematic reviews using random forest and external information. *Mach Learn* 102(3):465–482. <https://doi.org/10.1007/s10994-015-5535-7>
21. Tao X, Li Q, Ren C, Guo W, Li C, He Q, Liu R, Zou J (2019) Real-value negative selection over-sampling for imbalanced data set learning. *Expert Syst Appl* 129:118–134. <https://doi.org/10.1016/j.eswa.2019.04.011>
22. Choi JS, Ha MK, Trinh TX, Yoon TH, Byun HG (2018) Towards a generalized toxicity prediction model for oxide nanomaterials using integrated data from different sources. *Sci Rep* 8(1):6110
23. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357. <https://doi.org/10.1613/jair.953>
24. Han H, Wang W, Mao B (2005): Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: Huang, D., Zhang, X.S., Huang, G. (eds.) *Advances in intelligent computing*, international conference on intelligent computing, ICIC 2005, Hefei, China, August 23–26, Proceedings, Part I. Lecture Notes in Computer Science, 3644, pp 878–887. Springer, (2005). https://doi.org/10.1007/11538059_91
25. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the international joint conference on neural networks, IJCNN 2008, Part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1–6, pp 1322–1328. IEEE. <https://doi.org/10.1109/IJCNN.2008.4633969>
26. Jo T, Japkowicz N (2004) Class imbalances versus small disjuncts. *SIGKDD Explor* 6(1):40–49. <https://doi.org/10.1145/1007730.1007737>
27. Liu Y, Liu S, Liu T, Wang Z (2014) New oversampling algorithm db\smote. *Comput Eng Appl*
28. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) *Advances in neural information processing systems 27: Annual conference on neural information processing systems 2014*, December 8–13, Montreal, Quebec, Canada, pp 2672–2680. <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html>
29. Mukherjee S, Asnani H, Lin E, Kannan S (2019) Clustergan: Latent space clustering in generative adversarial networks. In: The Thirty-Third AAAI conference on artificial intelligence, AAAI 2019, The Thirty-First innovative applications of artificial intelligence conference, IAAI 2019, The Ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27–February 1, pp 4610–4617. AAAI Press. <https://doi.org/10.1609/aaai.v33i01.33014610>
30. Wei X, Gong B, Liu Z, Lu W, Wang L (2018) Improving the improved training of wasserstein gans: A consistency term and its dual effect. In: 6th International conference on learning representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, conference track proceedings. OpenReview.net. <https://openreview.net/forum?id=SJx9GQb0->
31. Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: Bengio, Y., LeCun, Y. (eds.) 2nd international conference on learning representations, ICLR 2014, Banff, AB, Canada, April 14–16, conference track proceedings. [arXiv: org/abs/1312.6114](https://arxiv.org/abs/1312.6114)
32. Kingma DP, Dhariwal P (2018) Glow: generative flow with invertible 1x1 convolutions. In: Bengio, S., Wallach, H.M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018*, NeurIPS 2018, December 3–8, Montréal, Canada, pp 10236–10245. <https://proceedings.neurips.cc/paper/2018/hash/d139db6a236200b21cc7f752979132d0-Abstract.html>
33. Dinh L, Sohl-Dickstein J, Bengio S (2017) Density estimation using real NVP. In: 5th international conference on learning representations, ICLR 2017, Toulon, France, April 24–26, conference track proceedings. OpenReview.net, <https://openreview.net/forum?id=HkpbnH9lx>
34. Wang P, Li S, Ye F, Wang Z, Zhang M (2020) Packetcgan: exploratory study of class imbalance for encrypted traffic classification using CGAN. In: 2020 IEEE International conference on communications, ICC 2020, Dublin, Ireland, June 7–11, pp 1–7. IEEE. <https://doi.org/10.1109/ICC40277.2020.9148946>
35. Zhou B, Khosla A, Lapedriza À, Oliva A, Torralba A (2016) Learning deep features for discriminative localization. In: 2016 IEEE conference on computer vision and pattern recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016, pp 2921–2929. IEEE Computer Society. <https://doi.org/10.1109/CVPR.2016.319>
36. Mathur AP, Tippenhauer NO (2016) Swat: a water treatment testbed for research and training on ICS security. In: 2016 international workshop on cyber-physical systems for smart water networks, CySWater@CPSWeek 2016, Vienna, Austria, April 11, pp 31–36. IEEE Computer Society. <https://doi.org/10.1109/CySWater.2016.7469060>
37. Rodofile NR, Schmidt T, Sherry ST, Djameludin C, Radke K, Foo E (2017) Process control cyber-attacks and labelled datasets on

- s7comm critical infrastructure. In: Pieprzyk J, Suriadi S (eds) Information Security and Privacy—22nd Australasian Conference, ACISP 2017, Auckland, New Zealand, July 3–5, 2017, Proceedings, Part II. Lecture notes in computer science, 10343, pp 452–459. Springer. https://doi.org/10.1007/978-3-319-59870-3_30
38. Dahl GE, Yu D, Deng L, Acero A (2012) Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans Speech Audio Process* 20(1):30–42. <https://doi.org/10.1109/TASL.2011.2134090>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.