Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# A Robustness Evaluation Method for SUDO-Based CBTC Intrusion Detection Systems

**Guohua Wang**

wangguohua@mail.lzjtu.cn

Lanzhou Jiaotong University

**Qifan Yan**

Inner Mongolia Power Group

**Lei Zhang**

Lanzhou Jiaotong University

**Chuang Lu**

Lanzhou Jiaotong University

**Research Article**

**Keywords:** CBTC, Intrusion detection, Adversarial attacks, Deep learning, Robustness

**Posted Date:** March 8th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-8878824/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

**Version of Record:** A version of this preprint was published at Multimedia Systems on July 11th, 2026.
See the published version at https://doi.org/10.1007/s00530-026-02488-3.

# A Robustness Evaluation Method for SUDO-Based CBTC Intrusion Detection Systems

Guohua Wang[1*], Qifan Yan[2], Lei Zhang[1], Chuang Lu[1]

[1*]School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070, China.
[2]Inner Mongolia UHV Power Supply Company, Inner Mongolia Power Group, Hohhot, 010010, China.

*Corresponding author(s). E-mail(s): wangguohua@mail.lzjtu.cn;
Contributing authors: yanqifan@stu.lzjtu.edu.cn; 1721992454@qq.com;
bjwlxylc@163.com;

**Abstract**

To evaluate the robustness problem of Communication Based Train Control System in intrusion detection system constructed using deep learning model, a robustness evaluation method based on symmetric uncertainty distance optimization is proposed. First, the data sample imbalance problem is solved by a hybrid sampling technique combining SMOTE and ENN, which expands a small amount of attack data and reduces the percentage of normal traffic data. Second, symmetric uncertainty is used to select the features with the highest relevance and lowest redundancy to form the optimal attack feature set. Finally, adversarial samples are generated for the feature set using distance optimization based adversarial attack. The experimental results demonstrate that the proposed SUDO adversarial attack on CNN improves the attack success rate(ASR)by 2.26%.The CNN and LSTM models used in the experiments have attack success rate of 35.34%and 23.69%.In ResNet18 model attack success rate is 31.24%and ResNet34 model attack success rate is 36.02%.

**Keywords:** CBTC,Intrusion detection, Adversarial attacks, Deep learning, Robustness

# 1 Introduction

The Communication-Based Train Control System (CBTC) primarily utilizes wireless communication technologies to enable real-time data exchange between trains and ground equipment, significantly enhancing operational flexibility and safety [1]. Compared to traditional track circuit systems, CBTC offers higher efficiency and greater adaptability. CBTC utilizes advanced wireless communication methods such as GSM-R, Wi-Fi, LTE-M, and 5G[2] to achieve precise data transmission and real-time control between trains and ground equipment. This not only enhances the flexibility of train scheduling and operational efficiency but also simplifies the CBTC system architecture, reducing system construction and maintenance costs[3]. Early CBTC systems featured relatively high security due to internal information exchange, limited data volume, and closed architecture. However, as systems become increasingly open, frequent interactions with external networks create more potential attack vectors. Particularly in modern CBTC systems, information openness and interconnectivity heighten vulnerability to external attacks [4], while the reliability of train-to-ground and train-to-train signaling systems remains an area for improvement [5]. Even CBTC systems utilizing 5G technology face diverse network threats to their intrusion detection systems [6]. Against this backdrop, ensuring the cybersecurity of CBTC systems has become an urgent issue requiring resolution.

Intrusion Detection Systems (IDS) stand as a critical technology in cybersecurity defense. Research has primarily focused on methods based on misuse detection and anomaly detection. Each approach possesses distinct advantages and disadvantages, and they are increasingly being integrated to enhance detection accuracy and coverage [7]. However, most current research primarily focuses on improving the detection performance of intrusion detection systems, while neglecting the robustness of IDS systems against adversarial attacks [8–13]. This lack of robustness may lead to significant vulnerability in practical applications when systems are subjected to attacks.

Adversarial attacks have garnered significant attention in recent years within the fields of machine learning and cybersecurity. These attacks introduce minute perturbations into input data, causing machine learning models to produce erroneous judgments when confronted with such data [14]. Although these perturbations are nearly imperceptible to humans, they can substantially reduce the classification accuracy of machine learning models, thereby misleading systems into making incorrect decisions [15]. This vulnerability stems from the inherently opaque and non-interpretable nature of deep learning models, which exhibit pronounced robustness weaknesses when confronted with carefully engineered inputs [16]. Adversarial attacks exploit this vulnerability precisely, posing a significant threat to intrusion detection systems built upon deep learning.

As shown in Fig. 1, the CBTC system block diagram depicts wireless base stations connected to CBTC ground equipment, onboard equipment, and the core routers of the wireless train control center. Data exchange with external networks also occurs via gateways. Throughout these communication processes, numerous network devices—such as switches and routers—serve as information exchange nodes. If these critical information exchange nodes are attacked, network communications will be compromised. The CBTC intrusion detection system is deployed at core routing nodes or

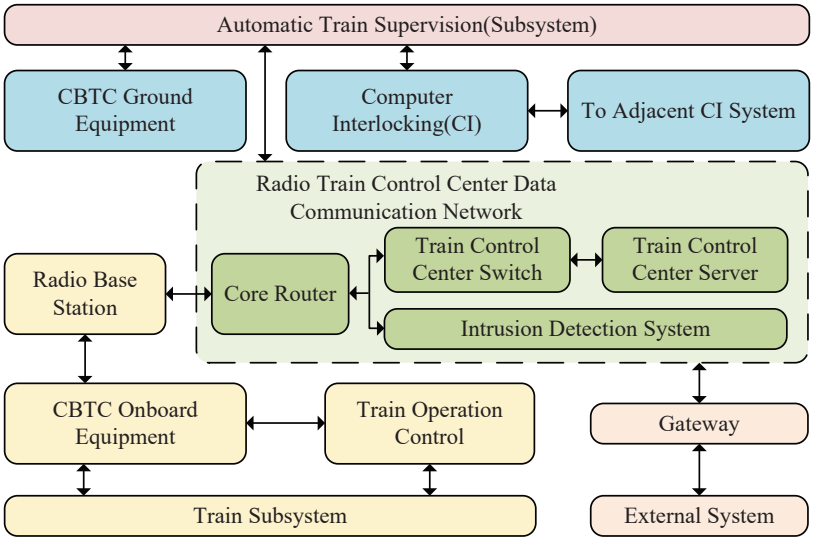



**Fig. 1**: System Diagram of CBTC

taps network traffic via bypass connections to detect anomalous traffic patterns. When train control data or communication network data transmission is compromised by attacks (such as tampering or spoofing attacks), the intrusion detection system may make erroneous decisions based on this abnormal data. This could directly jeopardize train operational safety, leading to severe consequences.

In 2017, Rigaki et al.[17] first applied adversarial example technology to the field of intrusion detection and demonstrated its potential threats to existing intrusion detection systems. Since then, an increasing number of studies have explored the application of adversarial attacks in intrusion detection systems. These include not only gradient-based adversarial attacks[18, 19] but also generative adversarial attacks based on generative adversarial networks[20, 21]. However, despite these studies proposing various effective attack methods in theory, relevant research remains extremely limited in actual CBTC systems. Assessing the robustness of CBTC intrusion detection systems requires addressing the following issues:

1. Large data transmission volume with significant imbalance. Attack samples constitute a very small proportion compared to benign samples. Generating adversarial attack samples alongside benign samples causes the feature information of minority-category attack samples to be overwhelmed by the information from majority-category benign samples. This prevents the model from learning the characteristics of the minority attack samples.
2. Data features exhibit high complexity and high dimensionality[22], particularly in the context of adversarial attacks. Effectively processing and analyzing these high-dimensional data has become a critical challenge.

To address the aforementioned issues, we propose a SUDO-based adversarial attack method that achieves significant attack effects with minimal perturbations. This approach is integrated into the CBTC intrusion detection system to simulate potential attack scenarios and evaluate the system's robustness under these conditions.

This paper primarily addresses the following aspects:

1. Firstly, this paper employs a hybrid sampling method combining SMOTE and ENN to augment the minority class samples, thereby enhancing the model's ability to recognize attack behaviors. Data imbalance is a significant factor affecting detection performance; by expanding the minority class samples, the model's generalization capability in detecting attack behaviors can be significantly improved.
2. Secondly, the Fast Correlation-Based Filter (FCBF) algorithm is proposed for feature selection[23], enhancing attack specificity while reducing feature redundancy. The FCBF algorithm calculates the correlation between features and target variables using Symmetrical Uncertainty (SU), rapidly selecting features most relevant to attack behaviors. This approach minimizes the impact of irrelevant or redundant features on model performance.
3. Finally, building upon the above work, adversarial samples were generated using the C&W (Carlini and Wagner) adversarial attack method proposed by Carlini and Wagner[24]. These samples were then used to replace the corresponding features of the original samples before being input into the intrusion detection system (IDS) of the CBTC system to assess the IDS's robustness. Existing research indicates that the C&W attack method maintains a high attack success rate even under various adversarial defense strategies[25, 26].

# 2 Research Methods

## 2.1 Overall Process Architecture

The overall process architecture of the proposed SUDO-based CBTC intrusion detection system robustness evaluation method is shown in Figure 2.

First, network traffic data is captured from traffic probes tapped off the core routing of the CBTC wireless train control center. CICFlowMeter extracts features and converts them into usable pcap files. After preprocessing, a hybrid sampling method combining undersampling and oversampling is applied to balance data proportions, preventing attack sample information from being drowned out by vast amounts of benign data during adversarial sample generation. Second, employ symmetric uncertainty-based feature selection to identify features with the highest relevance to labels and minimal redundancy, forming an optimal attack feature set. Then, iteratively train this optimal feature set using distance-optimized adversarial attacks, generating adversarial samples through minor perturbations. Finally, the features of the adversarial samples are replaced with those from the hybrid-sampled samples before inputting them into a deep learning-based CBTC intrusion detection system pre-trained on public datasets. The sample classification results of the IDS system

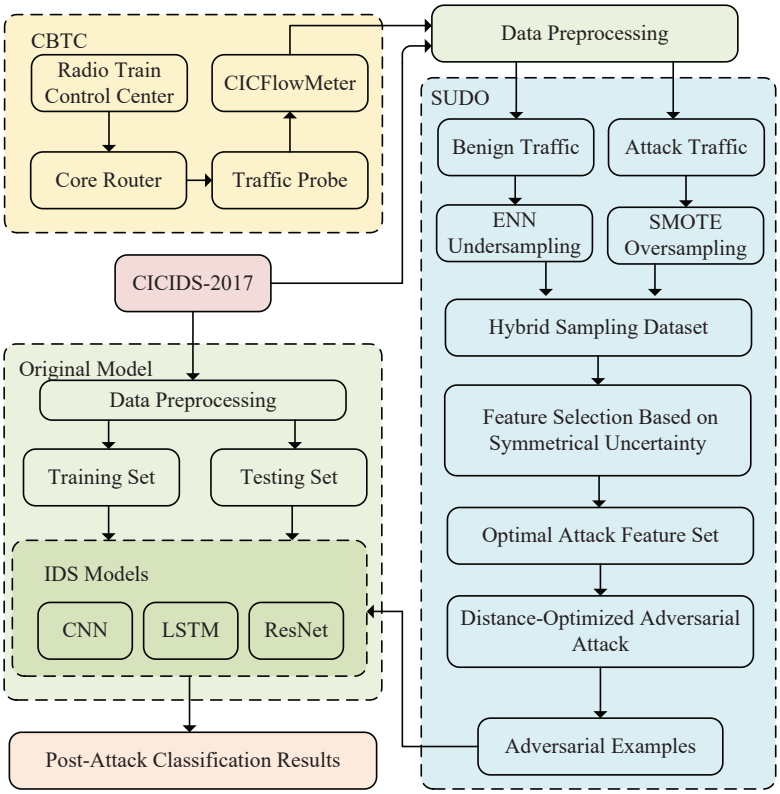



**Fig. 2**: Overall Process Architecture

after applying SUDO adversarial attacks are evaluated, with the robustness of the model assessed by the degree of decline in evaluation metrics.

# 3 Research Methods

## 3.1 Hybrid Sampling Method Combining SMOTE and ENN

### 3.1.1 SMOTE Oversampling

In classification tasks, data imbalance leads models to favor accurately identifying samples from the majority class. This pattern is particularly pronounced in CBTC system intrusion detection: since the system operates in a benign state for the vast majority of its runtime, attack or anomaly samples (such as DoS attacks and spoofing attacks) become significantly scarce and unevenly distributed within the dataset. Training directly on such imbalanced data makes it difficult for models to effectively

identify critical attack behaviors. To address this, this paper employs a hybrid sampling approach combining SMOTE and ENN to balance the dataset. By artificially augmenting the minority class (attack class) samples, a more representative training set is constructed. This fundamentally corrects the model's classification bias, enhancing its ability to detect security threats.

The data imbalance problem refers to situations in classification tasks where certain categories have significantly fewer samples than others, causing classifiers to favor categories with more samples. Due to the relatively scarce attack traffic samples and data imbalance in CBTC systems, hybrid sampling can effectively augment minority class samples, enhancing the model's ability to identify attack behaviors. SMOTE increases the number of attack samples by generating new synthetic samples among the existing minority attack samples, thereby balancing the dataset[27]. The formula is shown in Equation (1), which generates new samples through linear interpolation based on the differences between the selected minority samples and their neighboring samples.

$$x_{\text{new}} = x_i + \lambda \times (x_{\text{nieghbor}} - x_i) \tag{1}$$

Where $x_i$ is the starting point for generating a new sample, $x_{\text{new}}$is a sample point selected from the set of minority class samples. $x_{\text{nieghbor}}$ is a randomly selected neighboring sample from $k$ nearest neighbors of $x_i$, which serves as another reference point for generating $x_{\text{new}}$. $\lambda$ is a random number ranging between $[0, 1]$ that determines the proportion of the position of $x_{\text{new}}$ between $x_i$ and $x_{\text{nieghbor}}$.

### 3.1.2 ENN Undersampling

ENN is an undersampling method for data cleaning[28]. Its primary purpose is to enhance classifier performance by removing marginal or noisy samples, particularly on imbalanced datasets. ENN effectively refines data boundaries and reduces the impact of noisy samples.

Due to the random nature of the SMOTE algorithm when selecting samples, performing nearest-neighbor interpolation between sample points at the edges of the dataset may generate noisy data. This can blur sample boundaries, making them less distinct. ENN cleans up noise potentially introduced by edge interpolation, enabling the CBTC system's classifier to generalize better on imbalanced datasets and enhance its ability to detect attack behaviors.

First, for each sample $x_i$ in the CBTC intrusion detection system, Euclidean distance is used to identify and remove outlier or noisy samples that may compromise the accuracy of CBTC intrusion detection.

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{d} (x_{i,l} - x_{j,l})^2} \tag{2}$$

Where, in Equation (2), $x_i$ and $x_j$ represent two samples, $x_{i,j}$ and $x_{j,l}$ denote their $l$-th features, and $d$ is the dimensionality of the samples. Find the $k$ nearest neighbor samples to this sample $N(x_i) = \{x_{i1}, x_{i2}, \cdots, x_{ik}\}$.

Second, calculate the multi-class samples $\hat{y}$ in the nearest neighbor samples using the formula shown in Equation (3).

$$\hat{y} = \arg\max_{y} \sum_{j=1}^{k} \mathbb{I}\left(y_{ij} = y\right) \tag{3}$$

Where $\mathbb{I}\left(y_{ij} = y\right)$ is an indicator function that takes a value of 1 if $y_i j = y$ holds, and 0 otherwise. For each sample $x_i$ category $y$ among the nearest neighbors, the number of times it appears is calculated. If the majority of labels in its $k$ nearest neighbor samples differ from its own label $y_i$, it is deemed a noisy or boundary sample and removed.

## 3.2 Adversarial Attacks Based on Symmetric Uncertainty and Distance Optimization

### 3.2.1 Optimal Attack Feature Selection Based on Symmetric Uncertainty

The goal of adversarial attacks is to maximize the misclassification of a model's classification results by applying the smallest possible perturbations to the sample. Meanwhile, network traffic data collected in intrusion detection systems consists of high-dimensional features. To enhance the targeting of attacks and reduce feature redundancy, SU is applied for feature selection. This process identifies the features most relevant to the classification labels, constructing an optimal attack feature set for subsequent adversarial sample generation.

When conducting adversarial attacks against intrusion detection models for CBTC systems, attack construction must fully account for the system's inherent data characteristics and security constraints. Information exchange within CBTC systems involves high-dimensional traffic data, often containing redundant features and environmental noise. This invalid information not only interferes with the judgment of normal detection models but also reduces the effectiveness of adversarial perturbations while increasing unnecessary computational overhead. More critically, CBTC systems demand extreme safety-critical real-time performance. Increased algorithmic complexity due to redundant features directly conflicts with the system's real-time responsiveness. Therefore, performing data dimensionality reduction and feature selection before model training is an essential step. Therefore, this paper employs the FCBF algorithm to select data features, identifying and constructing a feature subset with high correlation to label information and low redundancy as the attack-specific feature subset. This approach ensures that other features and packet structures remain intact.

The FCBF algorithm is a feature selection method based on Symmetrical Uncertainty (SU). SU is a normalized mutual information metric used to measure the correlation between two variables. SU values range from 0 to 1, with higher values indicating stronger correlation. Specifically, the formula for calculating SU is given by Equation (4).

$$\mathrm{SU}(X, Y) = 2 \times \frac{\mathrm{IG}(X|Y)}{H(X) + H(Y)} \tag{4}$$

Where $\mathrm{IG}(X|Y)$ is the information gain between the feature $X$ and the target variable $Y$, while $H(X)$ and $H(Y)$ represent the entropies of $X$ and $Y$, respectively. Information gain measures the contribution of a feature to reducing the uncertainty of the target variable $Y$, defined as Equation (5):

$$\mathrm{IG}(X|Y) = H(Y) - H(Y|X) \tag{5}$$

During the feature selection process using the FCBF algorithm, the SU value between each feature and the target variable is first calculated. Subsequently, a predefined correlation threshold is applied to select features with higher correlation to the target variable. Only those features with SU values greater than or equal to the threshold are retained, forming the candidate feature set.

To further reduce redundant features, each pair of features in the candidate feature set undergoes a correlation assessment. If the SU value between any given pair of features is greater than or equal to the SU values between other features and the target variable, these two features are deemed redundant. In such cases, redundant features are removed, thereby further streamlining the feature set.

The specific process is shown in Algorithm 1.

**Algorithm 1** Optimal Attack Feature Selection Based on SU

**Require:** $S(F_1, F_2, \cdots, F_N, C)$ // dataset
**Require:** $\delta$ // Preset threshold
**Ensure:** $S_{\text{best}}$ // Optimal Attack Feature Set
**begin**
$S'_{\text{list}} \leftarrow []$ // Initialize an empty list of features
**for** $i = 1$ to $N$ **do**
  $\mathrm{SU}_{i,c} \leftarrow \mathrm{SU}(F_i, C)$
  **if** $(\mathrm{SU}_{i,c} > \delta)$ **then**
    append $F_i$ to $S'_{\text{list}}$
  **end if**
**end for**
order $S'_{\text{list}}$ in descending SU value // Remove redundant features
**for** each $F_i$ in $S'_{\text{list}}$ **do**
  **for** each $F_j$ in $S'_{\text{list}}$ **do**
    **if** $(F_i \neq F_j)$ **then**
      $\mathrm{SU}_{i,j} \leftarrow \mathrm{SU}(F_i, F_j)$
      **if** $(\mathrm{SU}_{i,j} \geq \mathrm{SU}_{i,c})$ **then**
        remove $F_j$ from $S'_{\text{list}}$
      **end if**
    **end if**
  **end for**
**end for**
$S_{\text{best}} \leftarrow S'_{\text{list}}$
**end**

Algorithm 1 is used to select a highly relevant feature subset from a given feature set. The algorithm's input includes a training dataset $S(F_1, F_2, \cdots, F_N, C)$ containing multiple features and their class labels, along with a predefined threshold $\delta$. The output is a selected feature subset $S_{\text{best}}$.

First, for each feature $F_i$, compute the $\text{SU}_{i,c}$ between that feature and the class label. If $\text{SU}_{i,c}$ exceeds a predefined threshold $\delta$, add that feature $F_i$ to the candidate feature subset $S'_{\text{list}}$. Second, sort the candidate feature subset $S'_{\text{list}}$ in descending order based on the symmetric uncertainty values $\text{SU}_{i,c}$ and obtain the features $F_j$ in $S'_{\text{list}}$.

Finally, traverse each feature $F_i$ in $S'_{\text{list}}$, and if $F_i$ differs from $F_j$, compute the $\text{SU}_{i,j}$ between $F_i$ and $F_j$. If $\text{SU}_{i,j}$ is greater than or equal to $\text{SU}_{i,c}$, remove $f_j$ from $S'_{\text{list}}$. Continue to obtain the next feature $F_i$ in $S'_{\text{list}}$ and repeat the above process. Continue until all features have been processed; the final selected feature subset is the filtered $S'_{\text{list}}$.

This algorithm evaluates feature importance by calculating symmetric uncertainty between features and class labels, and enhances feature selection efficiency by removing redundant features. This approach reduces the number of features while maintaining classification performance.

### 3.2.2 Distance-optimized adversarial attacks

Adversarial attacks aim to induce deep learning models to produce incorrect predictions by making subtle modifications to input data. Such attacks reveal a model's vulnerability to malicious inputs, thereby evaluating its security and robustness. C&W attacks represent a significant adversarial technique in deep learning, generating adversarial examples through distance optimization (DO) to cause misclassification by deep learning models.

The C&W attack transforms the task of generating adversarial examples into a mathematical optimization problem by constructing an optimization problem. This ensures that the attack samples can successfully deceive the model while maintaining similarity to the original samples. The objective is to find an adversarial example $x_{\text{adv}}$ such that the sample is misclassified by the model while minimizing the perturbation magnitude $\|x' - x\|_p$. Specifically, this can be described as the optimization problem in Equation (6).

$$J(x') = \min_{x'} \|x' - x\|_p + c \cdot f(x') \tag{6}$$

Where $\|x' - x\|_p$ denotes the $p$-norm of the perturbation quantity, commonly using the $L_0$ norm, $L_2$ norm, or $L_\infty$ norm. The $L_2$ norm is used to generate adversarial samples with minimal overall perturbation by optimizing the Euclidean distance. $c$ represents the confidence level, used to balance the perturbation magnitude and attack effectiveness, while $f(x')$ measures the attack effectiveness of the adversarial sample $x'$. $f(x')$ is defined by the following equation (7):

$$f(x') = \min(\max(Z(x')_i - Z(x')_t - \kappa) \tag{7}$$

Where $Z(x')$ is the model's logical output, $t$ is the target category, and $\kappa$ is a tunable parameter controlling the adversarial sample's confidence level. By minimizing

$f(x')$, we ensure that the adversarial sample $x'$ is classified by the intrusion detection system as the target category $t$ with higher confidence than non-target categories. This evaluates the CBTC system's robustness against high-confidence attacks.

The specific process is shown in Algorithm 2. First, the original sample $x$ is used as the initial value for the adversarial sample $x'$, and the optimization parameters $c$, $\kappa$, and the maximum iteration count are set. Next, through iterative optimization, the objective function $J(x')$ is minimized to obtain the optimal perturbation $\|x' - x\|_p$ . Finally, generate the final adversarial sample $x'$ based on the optimization results. If the maximum iteration count is reached or the optimization objective converges, terminate the optimization process; otherwise, continue optimizing.

**Algorithm 2** Distance-Optimized Adversarial Attacks

**Require:** $x$ // input sample
**Require:** $c$ // balance parameter
**Require:** $\kappa$ // confidence level
**Require:** $t$ // target category
**Require:** max_iter // maximum number of iterations
**Ensure:** $x'_{\text{best}}$ // optimal adversarial sample

**begin**
$w \leftarrow \arctanh(x)$
$x'_{\text{best}} \leftarrow x$
loss $\leftarrow \infty$
**for** each value for $c$ **do**
  **for** $i = 1$ to max_iter **do**
    $x' \leftarrow (\tanh(w) + 1)/2$
    $f(x') \leftarrow \max\left(\max_{j \neq t} Z(x')_j - Z(x')_t, -\kappa\right)$
    $J(x') \leftarrow \|x' - x\|_n + c \cdot f(x')$
    **if** $(J(x') <$ loss$)$ **then**
      loss $\leftarrow J(x')$
      $x'_{\text{best}} \leftarrow x'$
    **end if**
  **end for**
**end for**
**return** $x'_{\text{best}}$
**end**

# 4 Experimental Design and Results Analysis

## 4.1 Dataset

The CICIDS-2017 dataset provides a testing platform for intrusion detection system research that closely approximates real-world network environments[29]. The large-scale nature of CICIDS-2017 enables researchers to train complex machine learning

models and conduct in-depth statistical analysis. Furthermore, the dataset's 78 multidimensional features and 15 distinct attack categories, coupled with the inherent imbalance between attack and normal traffic, accurately reflect the scarcity of attack traffic observed in real-world scenarios.

## 4.2 Evaluation Indicators

Model classification results fall into four categories. True Positive (TP): the number of samples that are actually positive and correctly predicted as positive. False Positive (FP): the number of samples that are actually negative but incorrectly predicted as positive. True Negative (TN): The number of samples that are actually negative and correctly predicted as negative. False Negative (FN): The number of samples that are actually positive and incorrectly predicted as negative.

To evaluate model robustness, accuracy (ACC), recall, F1 score, and attack success rate (ASR) are used to measure the strength of robustness. Among these, ACC is employed to assess the model's classification capability before and after attacks, and can be expressed as in Equation (8).

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Recall measures a model's ability to identify samples in the positive class and can be expressed as Equation (9).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

F1 score provides a more balanced evaluation metric when simultaneously considering a model's accuracy and recall capability. It can be expressed as in Equation (10).

$$\text{F1} == \frac{2TP}{2TP + FP + FN} \tag{10}$$

where Precision $= \frac{TP}{TP+FP}$.

ASR is the proportion of attack-successful samples among all adversarial samples, serving as a measure of an attack's ability to mislead a model's classification capability. It can be expressed as follows, Equation (11).

$$ASR = \frac{N_{succeed}}{N_{total}} \tag{11}$$

A higher ASR indicates that attacks have a stronger effect on the model, and the model's robustness is weaker.

## 4.3 Experimental Setup

The experimental system runs Windows 11, with Python 3.9 as the programming language. The development environment is PyCharm, and the deep learning framework is PyTorch 1.12.1. Hardware specifications include: Processor: AMD Ryzen 7 6800H, Graphics Card: NVIDIA RTX 3070 Ti (8GB), Memory: 16GB DDR4.

The deep learning models employed in the experiment include: Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM)[30], and two residual neural networks (ResNet) of varying depths—ResNet18 and ResNet34[31].

Model training employs the Adam optimizer with gradient descent using cross-entropy loss. The batch size is set to 128, the learning rate to 0.0001, and training runs for 50 epochs with a dropout rate of 0.3.

## 4.4 Experimental Results and Analysis

### 4.4.1 Hybrid Sampling of SMOTE and ENN

The CICIDS-2017 dataset contains a total of 1,667,613 data points. Among these, benign samples constitute the largest category with 1,352,766 entries, while DDoS attacks represent the most prevalent attack type with 111,213 instances. Conversely, Heartbleed attack samples are the scarcest, numbering only 11. This dataset exhibits a significant imbalance issue.

First, data cleaning is performed to remove invalid feature data and data with missing features. Then, oversampling is applied to the minority class samples—attack traffic types with fewer than 5,000 instances. This method increases the number of minority class samples to over 5,000. Simultaneously, to address the benign samples constituting 81% of the dataset, undersampling was applied to reduce their count to approximately 670,000. This approach mitigates the risk of model overfitting to benign samples, ensuring the model does not overlearn non-critical features of benign traffic. The sample sizes before and after combined sampling are shown in Table 1.

**Table 1**: Number of sample classifications in the CICIDS-2017 dataset

| Serial number | Classification name | Original sample | Sample after hybrid sampling |
|---|---|---|---|
| 1 | BENIGN | 1352766 | 675868 |
| 2 | DDos | 111213 | 111212 |
| 3 | Dos Hulk | 90574 | 90259 |
| 4 | PortScan | 80337 | 80283 |
| 5 | DoS GoldenEye | 9169 | 9169 |
| 6 | FTP-Patator | 7938 | 7935 |
| 7 | DoS slowloris | 5796 | 5796 |
| 8 | DoS Slowhttptest | 5499 | 5499 |
| 9 | Bot | 1686 | 6704 |
| 10 | Web Attack-Brute Force | 1507 | 6028 |
| 11 | Web Attack-XSS | 652 | 5216 |
| 12 | SSH-Patator | 417 | 6672 |
| 13 | Infiltration | 27 | 6912 |
| 14 | Web Attack-Sql Injection | 21 | 5376 |
| 15 | Heartbleed | 11 | 5632 |

Through this hybrid sampling method combining oversampling and undersampling, the class proportions in the dataset are effectively balanced. This approach not only prevents the model from underfitting the minority class samples but also avoids

overfitting the majority class samples, thereby enhancing the model's generalization capability. Model training and evaluation are conducted on the hybrid-sampled dataset using a CNN model for both training and testing. The training loss and accuracy are shown in Figure 3.

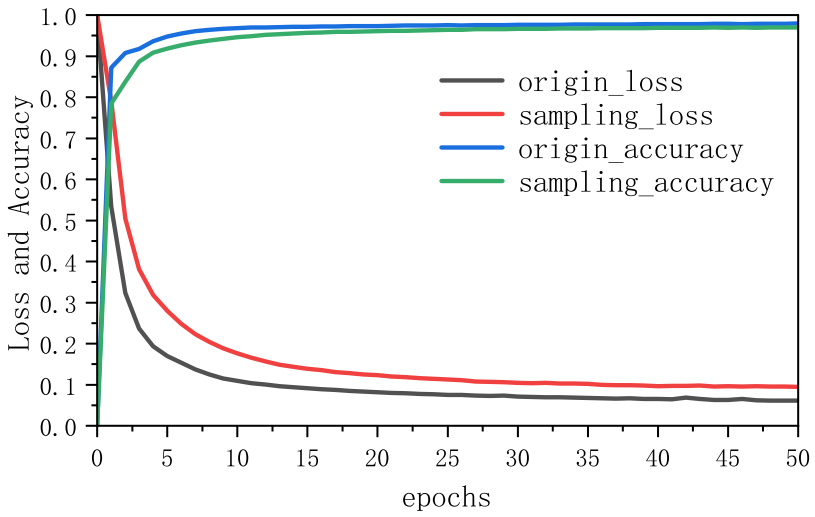



**Fig. 3**: CNN Training Curves for Original Samples and Hybrid Sample Samples

Table 2 experimental results indicate that among the five least frequent categories in the original dataset without hybrid sampling, only SSH-Patator achieved 53.49% accurate classification, while the other four attack categories were not learned by the model. After hybrid sampling, the classification accuracy for six out of the seven oversampled attack categories improved by at least 50%, with the exception of Web Attack-Brute Force.

The accuracy, recall, and F1 score of the model before and after hybrid sampling are shown in Fig. 4 below.

Figure 4 experimental results demonstrate that the hybrid sample achieved a 24.66% increase in recall and a 24.15% improvement in F1 score, while only experiencing a 0.24% decrease in accuracy. Not only does it maintain high accuracy for benign traffic, but recall and F1 score also show significant enhancement. This indicates a marked improvement in the model's generalization performance for detecting attack traffic.

### 4.4.2 Adversarial attacks based on symmetric uncertainty and distance optimization

To enhance attack effectiveness, a fast correlation-based filtering algorithm based on SU was employed to evaluate and select features from the CICIDS-2017 dataset. This

**Table 2**: Classification accuracy before and after hybrid sampling

| Serial number | Classification name | ACC (%) | |
|---|---|---|---|
| | | Original sample | Sample after hybrid sampling |
| 1 | BENIGN | 98.58 | 99.43 |
| 2 | DDos | 99.03 | 99.10 |
| 3 | Dos Hulk | 94.41 | 97.80 |
| 4 | PortScan | 98.94 | 99.17 |
| 5 | DoS GoldenEye | 97.31 | 96.06 |
| 6 | FTP-Patator | 58.79 | 65.24 |
| 7 | DoS slowloris | 66.09 | 100.00 |
| 8 | DoS Slowhttptest | 88.97 | 83.36 |
| 9 | Bot | 39.73 | 88.72 |
| 10 | Web Attack-Brute Force | 74.13 | 43.98 |
| 11 | Web Attack-XSS | 0.00 | 74.13 |
| 12 | SSH-Patator | 53.49 | 92.12 |
| 13 | Infiltration | 0.00 | 53.79 |
| 14 | Web Attack-Sql Injection | 0.00 | 89.80 |
| 15 | Heartbleed | 0.00 | 56.55 |

algorithm ranks features based on their correlation with attack labels, identifying those most relevant to the labels.

The correlation heatmap for the top 20 features most relevant to the label information is shown in Fig. 5 below.

Figure 5(a) shows the feature information entropy correlation heatmap after sorting by SU correlation. It reveals that some features are redundant for model training due to excessive similarity. After setting a correlation threshold, only the first feature exceeding the threshold is retained among several similar features. Figure 5(b) shows the correlation heatmap after removing redundant features. It is evident that eliminating similar features significantly reduces redundancy within the feature set.

After completing the sorting selection, to evaluate the specific impact of varying feature counts on model performance, features were progressively added to the model training in ascending order of SU values. The experiment began with the single most relevant feature and gradually increased the number of features until the top 30 highly correlated, low-redundancy features were included. The accuracy curve after training with 30 features is shown in Fig. 6.

Experiments demonstrate that under identical CNN models and hybrid sampling conditions, features ranked by decreasing SU relevance yield the greatest accuracy gains when the initial features exhibit the strongest label correlations. As feature counts increase, model performance improves but at diminishing rates. Once a certain threshold is reached, accuracy gains become negligible, indicating model saturation. When 19 features are employed in the model, the CNN accuracy approaches optimal and stabilizes, indicating that the key features capturing attack behaviors have been captured.

To evaluate the robustness of deep learning models, we first conducted control experiments using FGSM, PGD, C&W attacks, and SUDO adversarial attacks on

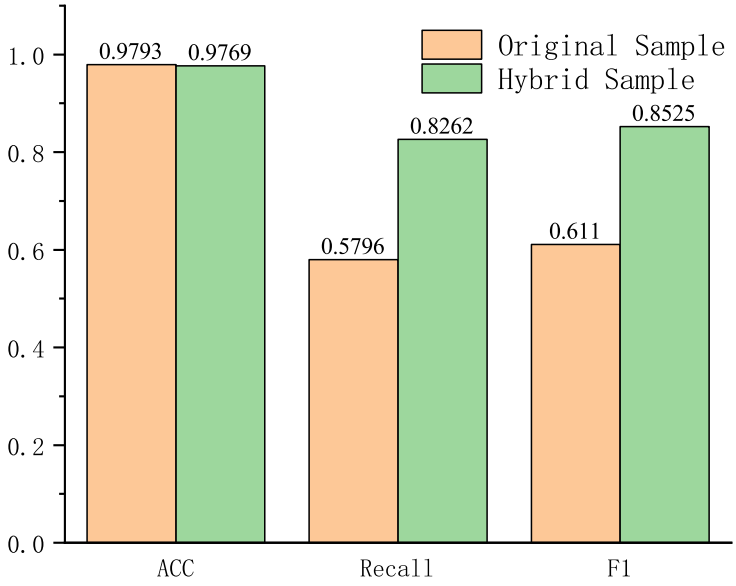


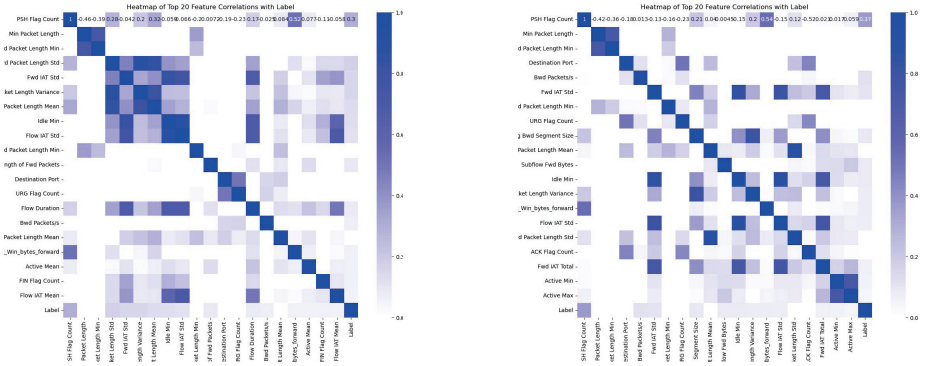

**Fig. 4**: Comparison of Evaluation Metrics Between Original and Hybrid Samples

CNN models to demonstrate the effectiveness of the proposed method. The results are shown in Table 3 below.

**Table 3**: Comparative Experiments between SUDO and Other Adversarial Attacks

| Adversarial Attack Methods | ACC | ASR | Recall | F1 |
|---|---|---|---|---|
| FGSM | 79.78 | 17.91 | 18.51 | 15.31 |
| PGD | 71.35 | 26.34 | 13.03 | 8.13 |
| C&W | 65.51 | 32.18 | 10.6 | 7.74 |
| SUDO | 63.25 | 35.34 | 6.67 | 5.28 |

Experiments demonstrate that in the intrusion detection scenario of CBTC systems, compared to PGD, FGSM, and C&W adversarial attacks, the optimal feature set generated by SUDO adversarial attacks—characterized by strong correlation and low redundancy—exhibits enhanced aggressiveness after hybrid sampling. Comparative experiments reveal that the ASR of SUDO adversarial attacks on CNN models increased by 17.43% relative to FGSM and 9.00% relative to PGD, and C&W by 2.26%. Compared to the highly aggressive C&W attack, recall decreased by 3.93%, and the F1 score dropped by 2.46%. This demonstrates that SUDO adversarial attacks




(a) Feature correlation heat map before feature selection

(b) Feature relevance heat map after feature selection

**Fig. 5**: Feature Correlation Heatmap Before and After Feature Selection

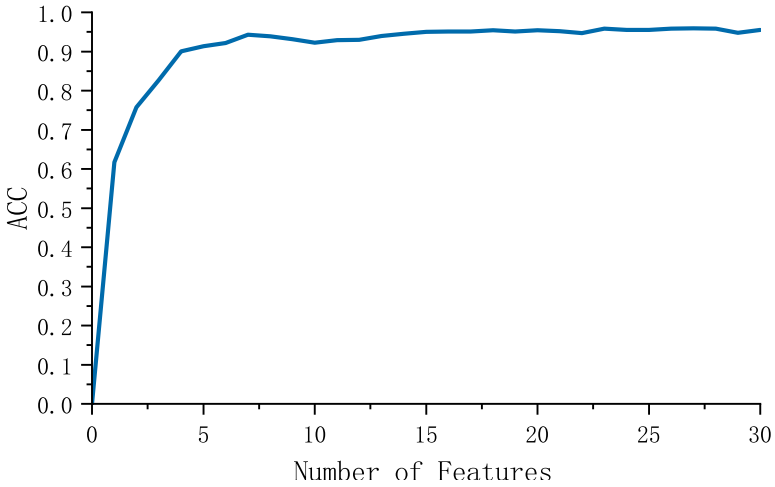



**Fig. 6**: Feature Quantity Accuracy Curve

exhibit greater aggressiveness within CBTC systems, enabling more effective exposure of the system's robustness vulnerabilities when subjected to adversarial attacks.

Then, SUDO adversarial attacks were applied to CNN, LSTM, ResNet18, and ResNet34 models. The robustness performance of each model was evaluated based on the degree of decline in its evaluation metrics, with the results shown in Table 4.

Experimental results demonstrate that CNN, LSTM, and ResNet models all exhibit significant performance degradation when subjected to SUDO adversarial attacks, confirming that these deep learning models share inherent robustness issues. CNN, due to its simpler structure, experienced a 35.34% decrease in ASR, a 75.95% drop in

**Table 4**: Evaluation indicator for different models before and after the SUDO adversarial attack

| Model | ASR | ACC | | Recall | | F1 |
|---|---|---|---|---|---|---|
| | | Before Attack | After Attack | Before Attack | After Attack | |
| CNN | 35.34 | 97.79 | 62.45 | 82.62 | 6.67 | 5.28 |
| LSTM | 23.69 | 92.32 | 68.63 | 70.22 | 18.92 | 16.43 |
| ResNet18 | 31.24 | 89.15 | 57.91 | 60.63 | 11.71 | 7.98 |
| ResNet34 | 36.02 | 90.14 | 54.12 | 63.91 | 7.67 | 5.13 |

Recall, and a 79.97% decline in F1 score. LSTM, suited for time-series network traffic data with a relatively complex structure, exhibited a smaller robustness decline under attack, yet still saw a 23.69% drop in ASR, a 51.3% decrease in Recall, and a 56.91% reduction in F1 score. ResNet18 demonstrated stronger robustness against SUDO adversarial attacks compared to CNN; ResNet34, with its greater depth and complexity relative to ResNet18, did not exhibit enhanced robustness under adversarial attacks, showing a 4.78% increase in ASR and larger declines in Recall and F1 scores of 7.23% and 5.76%, respectively. Ablation experiments demonstrate the feasibility and effectiveness of the proposed method for testing and evaluating the robustness of CBTC intrusion detection systems built using different deep learning models.

# 5 Conclusions

This paper proposes a SUDO adversarial attack method to detect the robustness of CBTC intrusion detection systems using deep learning models by combining hybrid sampling, feature selection methods based on symmetric uncertainty, and distance-optimized adversarial attacks.

First, the data imbalance problem is solved by the hybrid sampling method of SMOTE and ENN in the data preprocessing stage, and its effect is tested on the CNN model. Second, the data features are sorted using SU to obtain the optimal attack feature set. Finally, adversarial samples are generated through distance-optimized attacks. Through setting experiments, the accuracy, recall rate, and F1 value of CNN, LSTM, and two different depths of ResNet models are determined to detect the robustness of the CBTC intrusion detection system.

The experiments proved the effectiveness of the proposed SUDO adversarial attack. Among the various models, CNN had the most severe decline in recall rate and F1 value and the worst generalization. LSTM, a type of deep learning that processes time series, has stronger robustness when faced with adversarial attacks. The robustness of deep models is not simply positively or negatively correlated with their depth. Therefore, when designing and training deep learning models, without using adversarial defense methods, only the appropriate depth can achieve optimal performance when facing adversarial attacks. This requires continuous experimentation to obtain the most robust model.

In this study, although the proposed SUDO adversarial attack method effectively evaluated the robustness of the CBTC intrusion detection system, there are still some limitations. This study only tested a few types of deep learning models and cannot fully represent all possible architectures. Therefore, verifying the effectiveness of this method on a broader range of models remains one of the future tasks.

Future research should first conduct experimental tests using real-world traffic data to validate the effectiveness of the method. Then, further increase the types of attack methods and model architectures and test the performance of model attacks under different defense mechanisms.

## Declarations


- Funding
  This work was supported by the Key Research and Development Program of Gansu Province (Grant No. 25YFFA089), and the Natural Science Foundation of Gansu Province (Grant No. 25JRRA190).

- Conflict of interest
  The authors declare that they have no Conflict of interest.
- Ethics approval and consent to participate
  Not applicable.
- Consent for publication
  Not applicable.
- Data availability
  The datasets generated and/or analysed during the current study are available from the corresponding author on reasonable request.
- Materials availability
  Not applicable.
- Code availability
  The data used in this study are available upon reasonable request. The CICIDS2017 dataset utilized in this research is accessible to the public. It is located at the following URL: https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 28 February 2025).
- Author contribution
  Conceptualization, G.W. and Q.Y.; methodology, G.W. and Q.Y.; software, G.W. and Q.Y.; validation, G.W. and Q.Y.; formal analysis, G.W. and Q.Y.; investigation, G.W. and Q.Y.; data curation, G.W. and Q.Y.; writing—original draft preparation, Q.Y. and G.W.; writing review and editing, all authors.; visualization, L.Z. and C.L.; supervision, G.W. All authors have read and agreed to the published version of the manuscript.
- Corresponding author
  Correspondence to GuahuaWang

Editorial Policies for:

Springer journals and proceedings: https://www.springer.com/gp/editorial-policies

Nature Portfolio journals: https://www.nature.com/nature-research/editorial-policies

*Scientific Reports*: https://www.nature.com/srep/journal-policies/editorial-policies

BMC journals: https://www.biomedcentral.com/getpublished/editorial-policies

# References


[1] Farooq, J., Soler, J.: Radio communication for communications-based train control (cbtc): A tutorial and survey. IEEE Communications Surveys & Tutorials **19**(3), 1377–1402 (2017)

[2] Gheth, W., Rabie, K.M., Adebisi, B., Ijaz, M., Harris, G.: Communication systems of high-speed railway: a survey. Transactions on Emerging Telecommunications Technologies **32**(4), 4189 (2021)

[3] Song, H., Gao, S., Li, Y.: Train-centric communication based autonomous train control system. IEEE Transactions on Intelligent Vehicles **8**(1), 721–731 (2022)

[4] Gao, B., Bu, B., Zhang, W.: An intrusion detection method based on machine learning and state observer for train-ground communication systems. IEEE Transactions on Intelligent Transportation Systems **23**(7), 6608–6620 (2021)

[5] Kim, S., Won, Y., Park, I.H.: Cyber-physical vulnerability analysis of communication-based train control. IEEE Internet of Things Journal **6**(4), 6353–6362 (2019)

[6] Li, W., Yan, Z., He, R.: A novel machine learning based intrusion detection method for 5G empowered CBTC systems. In: Proc. 2022 International Wireless Communications and Mobile Computing (IWCMC), pp. 211–216 (2022)

[7] Liu, J., Qian, L., Zhang, Y.: Towards safety-risk prediction of CBTC systems with deep learning and formal methods. IEEE Access **8**, 16618–16626 (2020)

[8] Zhang, W., Bu, B., Wang, H.: An intrusion detection method of data tampering attack in communication-based train control system. In: Proc. 2019 IEEE Intelligent Transportation Systems Conference (ITSC), pp. 345–350 (2019)

[9] Sedjelmaci, H., Guenab, F., Boudguiga, A.: Cooperative security framework for CBTC network. In: Proc. 2018 IEEE International Conference on Communications (ICC), pp. 1–6 (2018)

[10] Zhang, Y., Liu, J., Sun, J.: Intelligent-prediction model of safety-risk for CBTC system by deep neural network. In: Proc. International Conference on Collaborative Computing: Networking, Applications and Worksharing, pp. 669–680 (2019)

[11] Li, Y., Zhu, L., Wang, H.: A cross-layer defense scheme for edge intelligence-enabled CBTC systems against MitM attacks. IEEE Transactions on Intelligent Transportation Systems **22**(4), 2286–2298 (2020)

[12] Lu, H., Zhao, Y., Song, Y.: A transfer learning-based intrusion detection system for zero-day attack in communication-based train control system. Cluster Computing **27**(6), 8477–8492 (2024)

[13] Zhang, R., Yang, G.: Research on DDoS attack detection based on GBDT-SVM model in SDN architecture. Journal of Computers **36**(1), 15–28 (2025)

[14] Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: Proc. 3rd International Conference on Learning Representations (ICLR) (2015)

[15] Khaleel, Y., Habeeb, M.A., Albahtial, A.S.: Network and cybersecurity applications of defense in adversarial attacks: A state-of-the-art using machine learning and deep learning methods. Journal of Intelligent Systems **33**(1), 20240153 (2024)

[16] Wu, W., Bu, B.: Security analysis for CBTC systems under attack–defense confrontation. Electronics **8**(9), 991 (2019)

[17] Rigaki, M., Elragal, A.: Adversarial deep learning against intrusion detection classifiers. In: Proc. NATO IST-152 (2017)

[18] Mohammadian, H., Ghorbani, A.A., Lashkari, A.H.: A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems. Applied Soft Computing **137**, 110173 (2023)

[19] Zhang, C., Costa-Perez, X., Patras, P.: Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. IEEE/ACM Transactions on Networking **30**(3), 1294–1311 (2022)

[20] Lin, Z., Shi, Y., Xue, Z.: IDSGAN: Generative adversarial networks for attack generation against intrusion detection. In: Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 79–91 (2022)

[21] Xu, N., Feng, W., Zhang, T.: FD-GAN: Generalizable and robust forgery detection via generative adversarial networks. International Journal of Computer Vision **132**(12), 5801–5819 (2024)

[22] Regragui, Y., Mazighi, A., Ballihi, L.: Impact evaluation of feature selection algorithms on machine learning-based intrusion detection. In: Proc. 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM), pp. 1–6 (2024)

[23] Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research **5**(Oct), 1205–1224 (2004)

[24] Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: Proc. 2017 IEEE Symposium on Security and Privacy (SP), pp. 39–57 (2017)

[25] Wang, G., Yan, Q.: A method for improving the robustness of intrusion detection systems based on auxiliary adversarial training Wasserstein generative adversarial networks. Electronics **14**(11), 2171 (2025)

[26] Wang, Z.: Deep learning-based intrusion detection with adversaries. IEEE Access **6**, 38367–38384 (2018)

[27] Chawla, N.V., Bowyer, K.W., Hall, L.O.: SMOTE: synthetic minority oversampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002)

[28] He, H.: ENN: extended nearest neighbor method for pattern recognition. IEEE Computational intelligence magazine **10**(3), 52–60 (2015)

[29] Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A.: Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSp, vol. 1, pp. 108–116 (2018)

[30] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)

[31] He, K., Zhang, X., Ren, S.: Deep residual learning for image recognition. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)