Enhanced IoT Intrusion Detection Using an Improved Autoencoder and Adversarial Convolutional Encoders

Yukun Peng, Yu Chen*
Zhangjiakou Open University, Zhangjiakou 075000, China
E-mail: Pengyukun150@163.com; windyu518@163.com
*Corresponding author

Keywords: intrusion detection, IoT, network security, attention mechanism, encoder

Received: February 14, 2025

To develop an efficient and intelligent automated intrusion detection system for IoT, this study proposes a malicious network traffic recognition model based on an improved autoencoder and adversarial convolutional encoder (AECE). The model first uses mixed sampling and improved autoencoder for data augmentation. Then, convolutional neural networks and gated recurrent units are used to extract spatial and temporal features. AECE combines the idea of generative adversarial networks to enhance the model's adaptability to complex attack patterns. Finally, experimental validation was conducted on the NSL-KDD, UNSW-NB15, IoT-23, and CSE-CIC-IDS2018 datasets. The results showed that the designed data augmentation algorithm could effectively improve the clustering and classification performance of the dataset, with a minimum Xie Beni value of 0.259, a maximum decrease of 15.88% in Davidson Boudin index, and a maximum improvement of 0.214 in classification accuracy. In the IoT-23 dataset, the highest detection rate of the baseline model was 0.882, while the detection rate of the proposed intrusion detection model was 0.949, with an increase of about 7.6%. At the same time, the model had a minimum loss convergence value of 0.08, a response time of 368.16 ms, and the values of false alarm rate fluctuated between 0.10 and 0.20. The comprehensive values of data traffic per second and packet capture per second confirmed that the model had strong detection ability and efficiency for attack behavior. This study expands the application scope of deep learning in anomaly detection, providing new ideas and methods for improving the security and stability of Internet of Things systems.

Povzetek: Predlagan je model AECE za inteligentno zaznavanje vdorov v IoT omrežjih. Uporablja izboljšani avtoenkoder za povečanje podatkov (rešuje neuravnoteženost) ter konvolucijske in ponavljajoče se enote (GRU) za ekstrakcijo prostorskih in časovnih značilnosti. Na naboru podatkov IoT-23 je AECE dosegel odlične rezultate.

1 Introduction

The Internet of Things (IoT) can realize real-time collection, analysis and interaction of various data by connecting various devices, sensors, systems, etc. to the Internet. At present, IoT has been applied in smart homes, healthcare, transportation, logistics, etc. [1]. IoT contains numerous heterogeneous devices, protocols, and platforms, with complex and diverse interactions between components. IoT devices are typically distributed across a wide geographic area, and their highly interconnected and decentralized nature makes them a hotspot for network attacks, threatening the confidentiality and security privacy of IoT data [2-3]. Therefore, establishing effective IoT security monitoring and response mechanisms to promptly detect and respond to potential security threats is crucial. Traditional security defense techniques include deploying complex security mechanisms directly on devices, conducting regular security updates, and patch management. However, IoT devices are limited in computing power, storage space, and other aspects, and their diversity and dispersion make it difficult to identify and defend against potential threats from malicious attacks [4]. Intrusion Detection Systems (IDS) can detect and report potential security threats by monitoring and analyzing data sources like network traffic and system logs. IDS has the advantages of real-time and proactive defense, and can be used to achieve security defense for IoT devices. However, malicious cyber attacks continue to emerge and develop, with increasingly diverse attack methods and strong concealment and destructive capabilities. This makes the current IDS relatively fragile and unable to effectively respond to new security threats. Ensuring IoT security requires more advanced and efficient IDS solutions [5]. In this context, how to build an efficient and intelligent automatic detection scheme for malicious network traffic intrusion in the IoT and improve the accuracy and efficiency of malicious network traffic identification, has become a key issue that urgently needs to be addressed. Therefore, this study focuses on the Malicious Network Traffic Identification (MNTI) algorithm in IDS. It introduces feature fusion, Attention Mechanism (AM), and improved Generative Adversarial Network (GAN) to construct the MNTI model, which can fully explore and utilize the spatiotemporal correlation of network traffic data, thereby more accurately detecting known and unknown network attacks. Firstly, a Data

Augmentation Algorithm (DAA) based on Mixed Sampling (MS) and Improved Autoencoder (IA) is designed to provide a higher quality data foundation for subsequent MNTI model training. Then, a Convolutional Neural Network (CNN), Gated Recurrent Unit (GRU), and AM are combined to build an MNTI model. A GAN-based Adversarial Convolutional Encoder model (AECE) is introduced to further enhance the MNTI's adaptability to complex attack patterns. This study innovatively combines oversampling and undersampling techniques for MS, and introduces Variational Autoencoders (VAEs) for dimensionality reduction of discrete data. This can effectively solve the problem of imbalanced number of normal and abnormal samples in IoT datasets and enhance the authenticity and richness of samples.

The study is structured into four main sections. Section 1 is a review of the current research status of network IDS-related technologies in the industry. Section 2 elaborates on the construction process of DAA and MNTI models. Section 3 involves performance testing and application analysis of the designed MNTI model. Section 4 summarizes the experimental results.

2 Related works

Intrusion Detection (ItruD) technology is an important guarantee in network security, playing a crucial role in responding to network attacks and protecting systems from malicious activities. Numerous scholars have conducted research on it. Machine Learning (ML) and Deep Learning (DL) have been widely applied in information security. Qazi et al. constructed a hybrid network IDS based on DL technology. The system used CNN to collect local features and utilized deep Recurrent Neural Networks (RNN) to extract features. The public dataset has confirmed the effectiveness of this method, with a mean accuracy of 98.90% when detect malicious attacks [6]. Improving network security for cloud computing and IoT was crucial. Kasongo first utilized the eXtreme Gradient Boosting (XGBoost) Feature Selection (FS) algorithm to lower down the feature space of the data, and then built an IDS framework based on ML. The experiment confirmed the performance of the research results [7]. IDS could effectively protect the security of IoT. Hazman et al. designed an integrated learning IDS framework based on IoT intelligent environment. This framework integrated Adaptive Boosting (AdaBoost), FS technique Boruta, mutual information, and correlation. In dataset validation, this method performed well in accuracy, recall, and precision, with a Detection Rate (DR) of approximately 99.9%, a learning computation time of 33.68 seconds, and a detection time of 0.02156 seconds [8]. Ghanbarzadeh et al. designed an IDS method based on the Horse Swarm Optimization Algorithm (HSOA) and K-nearest Neighbors (KNN), which mimics the behavior of horses and selects effective features for ItruD. This method used a base function to update HSOA into a discrete algorithm and combined it with quantum computing to implement the transformation of a quantum inspired optimizer for improving population social behavior. This method has improved the average size and classification accuracy of FS by 6%, and the accuracy of ItruD has reached 99.8% [9].

Elnakib et al. designed an enhanced anomaly-based ItruD DL multi-class classification model based on ML. This method outperformed other DL models in accuracy in classifying network traffic behavior [10]. To enhance the security of IoT, Mohy Edine M et al. constructed an FS model using principal component analysis, univariate statistical testing, and genetic algorithm, and integrated KNN to build an IoT network ItruD model. This method had high accuracy and detection time of less than one minute [11]. In response to the increased security risks of data transmission caused by interconnected nodes in IoT, Alotaibi et al. constructed a binary classification model for IoT traffic using various supervised ML models and ensemble classifiers. The classifier's accuracy surpassed that of a single model, and the predictive classification was significantly reduced [12]. The current IDS still had a high level of false positives, so Al Ghuwairi et al. developed a method for early detection of cloud computing intrusions using time series data. This method included FS and FS-based prediction models, which could effectively solve the problem of misleading connections between time series anomalies and attacks. This method significantly reduced the use of predictive factors and improved the prediction error index, reducing training time, prediction time, and cross-validation time by about 85%, 15%, and 97% [13]. The security and privacy vulnerabilities of the Internet were very urgent. Ntizikira et al. used Federated Learning (FL), Differential Privacy (DP), and secure multi-party computation to enhance data confidentiality, and integrated Deep Neural Networks (DNN) to achieve real-time anomaly detection. This method had excellent accuracy, precision, and recall [14]. Omer N et al. used Firefly Algorithm (FA) to detect intrusions before evaluating IDS, and then used Probabilistic Neural Networks (PNN) for classification. This method performed well with an accuracy rate of up to 98.99% [15]. The summary table of the above related work is shown in Table 1.

In summary, although network IDS has received a lot of research, existing IDS models generally face problems such as imbalanced data samples, fragmented spatiotemporal features, and adaptability to unknown attack patterns. In response to the above issues, this study reconstructs the data distribution, uses CNN and GRU to jointly mine spatiotemporal features, and enhances the model's ability to recognize unknown attacks using AECE. It enhances the comprehensive defense effectiveness of the model in complex attack scenarios from three dimensions: data layer, feature expression, and detection mechanism.

Table 1: Summary table of related work.

Literature	Model	Data set	Result	Limitation
[6]	CNN + RNN	CSE-CIC-IDS2018	The average accuracy rate is 98.90%	High consumption of computing resources
[7]	RNN + XGBoost	NSL-KDD	The accuracy rate is 97.8%	Insufficient generalization ability for zero day attacks
[8]	AdaBoost + Boruta	UNSW-NB15	The accuracy rate is 99.9%	Weak robustness of adversarial samples
[9]	HSOA + KNN	CIC-IDS2017	The accuracy rate is 99.8%	Parameter tuning is complex
[10]	ML	IoT-23	The accuracy of multi class classification is 98.7%	Poor model interpretability
[11]	KNN + genetic algorithm	TON_IoT	The accuracy rate is 98.3%	Significant information loss
[12]	ML + ensemble classifier	BoT-IoT	The binary classification accuracy is 99.2%	Poor scalability in multiple attack scenarios
[13]	FS	AWS CloudTrail logs	85% reduction in training time	Restricted transferability
[14]	FL + DP + DNN	CIC-IDS2019	The accuracy rate is 96.5%	Slow convergence speed
[15]	FA + PNN	KDD Cup 99	The accuracy rate is 98.99%	Insufficient coverage of modern attack modes

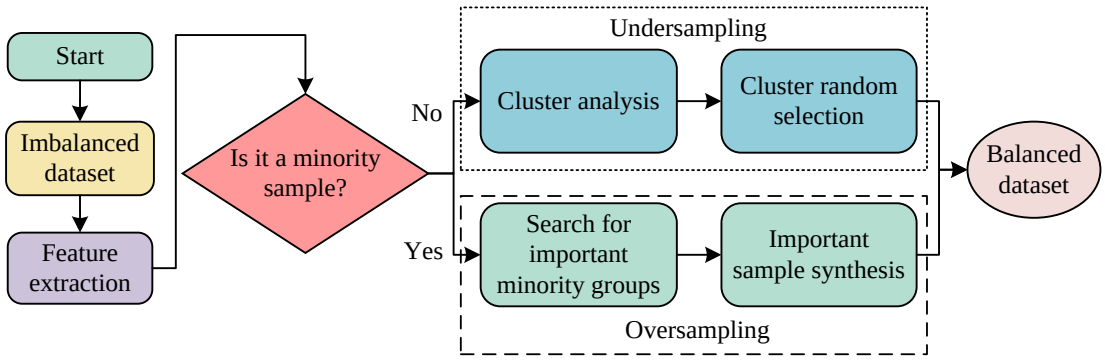


Figure 1: Schematic diagram of the workflow of DAA.

3 IoT security intrusion detection based on IA and AECE

ItruD technology can provide timely security alerts and response basis for network administrators. This study first designs DAA to improve the accuracy of traffic detection, and then integrates multiple DL technologies to construct the MNTI model.

3.1 Design of DAA based on MS and IA

With the popularity of IoT devices, the number of nodes has exploded, but normal network behavior accounts for the vast majority of traffic data, and there is a serious imbalance between the amount of abnormal samples and the normal samples. Unbalanced data samples can easily lead to a decrease in sample recognition accuracy [16]. Therefore, this study first designs DAA to address the sample imbalance, and Figure 1 shows the algorithm's framework.

In Figure 1, expansion operation is required for a few samples, while screening operation is required for most samples. Therefore, this study combines oversampling and undersampling techniques to construct a hybrid sampling system. Firstly, the category judgment threshold is determined, and the Synthetic Minority Oversampling Technique (SMOTE) is utilized to expand the imbalanced dataset and construct the training set for the classifier. SMOTE changes the distribution of minority classes by searching for their neighbors in the feature space and generating new synthetic samples between these samples [17]. In this study, SMOTE is used to expand minority class samples, balance the class distribution in the dataset, and ensure that the model can fully learn the features of various attack categories during training, thereby improving the model's generalization ability. Then, based on ensemble thinking, multiple classifiers are used to complete ensemble training. Finally, an ensemble classifier is used to search for important minority class samples and divide them into oversampling objects. XGBoost belongs to the category of ensemble learning algorithm Boosting. This algorithm improves prediction accuracy by integrating multiple weak learners into one strong learner. In IoT datasets, normal network behavior samples often outnumber abnormal samples. XGBoost can assign higher weights to minority class samples during the training process, thereby improving the recognition ability of minority classes and effectively solving the problem of data imbalance. Therefore, this study adopts XGBoost as the basic classifier. The basic learner of XGBoost is decision tree $h(x;\theta_m)$. x is the input data. θ_m is the parameter. The weighting of all decision trees is the final prediction result. The calculation process of θ_m is equation (1).

$$\theta_m = \left\{ \left(R_J, c_J \right) \right\}_{J=1}^{J} \quad (1)$$

In equation (1), R_J is the leaf node region, and $J \in R$. c_J is a constant. XGBoost generates decision trees in the direction of reducing residual g_t. The calculation process of g_t is equation (2).

$$g_t = \frac{\partial L\left(y_i, \hat{y}_i^{t-1} \right)}{\partial \hat{y}_i^{t-1}}, \mathrm{t} = \{1,2,...,\mathrm{N}\} \quad (2)$$

In equation (2), y_i represents the true value of the i-th sample, $\hat{y}_i^{t-1}$ represents the observed value of the sample at the $t-1$-th iteration, t represents the number of iterations, and N represents the maximum number of iterations. The update process of the estimation function $F(x)$ is equation (3).

$$F_t(x) = F_{t-1}(x) + kh\left(x; \theta_t \right) \quad (3)$$

In equation (3), k is a constant. The objective function of XGBoost is the superposition of the loss function and the penalty function, as calculated in equation (4).

$$L(\phi) = \sum_i^n l\left(y_i - y_i \right) + \sum_p^P \Omega\left(f_p \right) \quad (4)$$

In equation (4), y_i and y_i are predicted values and true values, and $\sum_i^n l\left(y_i - y_i \right)$ is the loss function. $\Omega\left(f_p \right)$ is the regularization term, and the calculation process is shown in equation (5).

$$\Omega(f) = \gamma J + \frac{1}{2}\lambda' \|w\|^J = \frac{1}{2}\lambda' \sum_{j=1}^{J} w_j^2 \quad (5)$$

In equation (5), w is the leaf weight. γ and λ' are regular penalty terms for leaves and their weights. In ML, models may overfit training data, leading to a decrease in predictive ability on new data. Regularization reduces the risk of overfitting by adding additional penalty terms to the loss function to limit the complexity of the model. Equation (5) limits the model's complexity by comprehensively considering the number of leaf nodes and leaf weight sizes in the tree, which helps to improve the predictive performance of the model on new data. To improve the prediction accuracy of XGBoost, the training set is introduced as a new function f for greedy optimization of the objective function, as expressed in equation (6).

$$L^{(t)} = \sum_i^n \left(l\left(y_i^{(t-1)} - y_i \right) + f\left(x_i \right) \right) + \Omega\left(f_p \right) \quad (6)$$

After expanding equation (6) according to the second-order Taylor formula, the final objective function $L^{(t)}$ is obtained through training simplification, as shown in equation (7).

$$L^{(t)} = \sum_{j=1}^{J} \left[\left(\sum_{i \in I} G_i \right) w_j + \frac{1}{2}\left(\sum_{i \in I} H_i + \lambda' \right) w_j^2 \right] + \gamma J \quad (7)$$

In equation (7), G and H are the first and second derivatives of the loss function. IoT datasets typically contain a large number of numerical features, and some density or hierarchical clustering algorithms may have issues with not being intuitive or having high computational costs when processing numerical data. The K-means algorithm has a simple principle and good clustering effect on numerical data. It can quickly divide the data into different clusters and select representative samples, thereby improving training efficiency. Therefore, this study adopts the K-means clustering algorithm for undersampling operation, and the objective function is shown in equation (8).

$$J(X, \pi) = \sum_{j=1}^{k} \sum_{i \in \pi_j} \left\| x_i - m_j \right\|^2 \quad (8)$$

In equation (8), π_j is class j. m_j is the center of a certain category. x_i is a data point. The MS method compensates for the shortcomings of traditional sampling techniques, but IoT datasets typically involve data with discrete characteristics. The SMOTE algorithm has low applicability to discrete data. VAE can map input data to latent space through an encoder, obtain representation vectors, output parameters of the representation vectors, and generate diverse new samples. This will increase the richness of the dataset and help improve the model's generalization ability. VAE has good processing ability for discrete data. Therefore, the study introduces VAE for dimensionality reduction of discrete data. VAE contains an encoder and a decoder. The encoder maps the input data x to the latent space to gain the representation vector z, and outputs the parameters of the representation vector. The decoder maps the representation vector back to the data space to generate new samples and ensures that the new samples are as similar as possible to the original input data [18-19]. The training objective of VAE is to optimize the variational lower bound $ELBO$, as shown in equation (9).

$$ELBO(q) = Eq(z|x)[logp(x|z)] \\ -DKL(q(z|x) \square p(z)) \quad (9)$$

In equation (9), $p(x|z)$ is the generative model defined by the decoder. $p(z)$ is a standard Gaussian distribution. DKL is the Kullback Leibler divergence. $q(z|x)$ is the posterior distribution. The working principle of DAA based on MS/IA is shown in Figure 2.

In Figure 2, data augmentation is divided into two stages: model training and data synthesis. Firstly, VAE is used to learn data features during the training phase and convert them into representation vectors with rich information. Then, the representation vector and data labels are input into the MS module to achieve balanced processing of the data. Finally, the decoder completes the conversion of the data format. In summary, the proposed

DAA based on MS and IA mainly consists of four steps. Step 1 inputs network traffic data and preprocesses the raw data. Step 2 determines the majority class and minority class samples, applies SMOTE to generate new composite samples for the minority class samples, and uses K-means clustering algorithm to undersample the majority class samples. Step 3 trains the VAE using the training set data and uses the trained VAE to perform dimensionality reduction and feature extraction on minority class samples. Step 4 fuses the synthesized samples generated by SMOTE and VAE to obtain an enhanced dataset, and performs weighted fusion with the majority class samples to obtain a balanced dataset. This study uses Xie Beni Index (XBI) and Davies Bouldin Index (DBI) as indicators to evaluate the clustering quality of DAAs. XBI evaluates clustering performance by measuring the distance between cluster centers and the closeness of data points within clusters. The smaller the value of XBI, the more tightly clustered the sample points within the cluster are, and the better the separation between different clusters, resulting in better clustering performance. DBI takes into account both intra-class sample similarity and inter-class sample difference, with smaller values indicating better clustering performance.

3.2 Design of MNTI model based on feature fusion and AECE

IDS is usually segmented into two types of signature detection and two main technologies. Anomaly detection is a detection technique that identifies abnormal activity by analyzing the normal behavior patterns of network traffic. When network traffic deviates from normal behavior patterns, the system considers it a potential malicious activity and triggers an alert [20]. Network traffic data typically contain a mixture of multiple types of information, which are correlated in both temporal and spatial dimensions. Network traffic data have obvious temporal characteristics. For example, network attack behavior usually shows sudden growth of traffic in a short period. At the same time, network traffic data also have spatial correlations. In the same IoT network, data transmission between servers and multiple clients may exhibit synchronous or correlated trends, and devices in the same network often share certain common network configurations and security policies. Therefore, to capture information at different levels, this study constructs the basic framework of the MNTI model based on the concept of feature fusion, as shown in Figure 3.

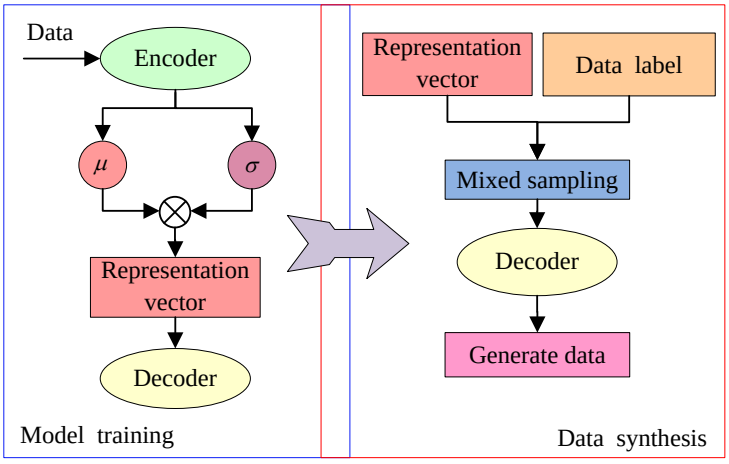


Figure 2: Schematic diagram of DAA based on MS/IA.

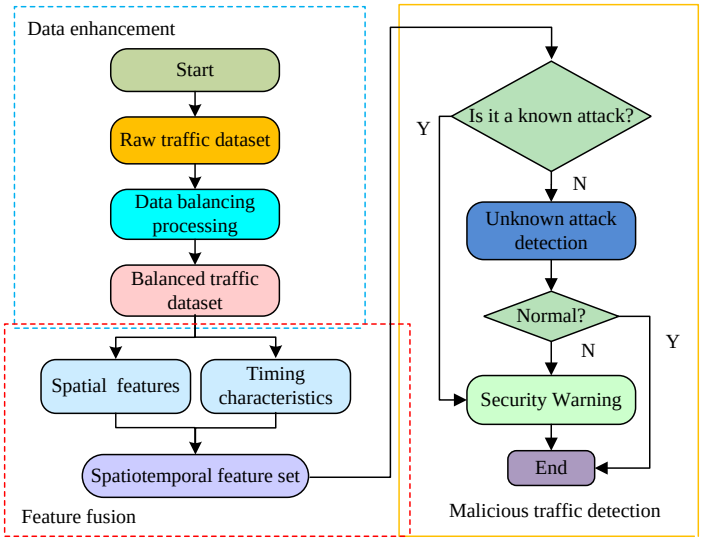


Figure 3: Basic framework structure of MNTI model based on feature fusion.

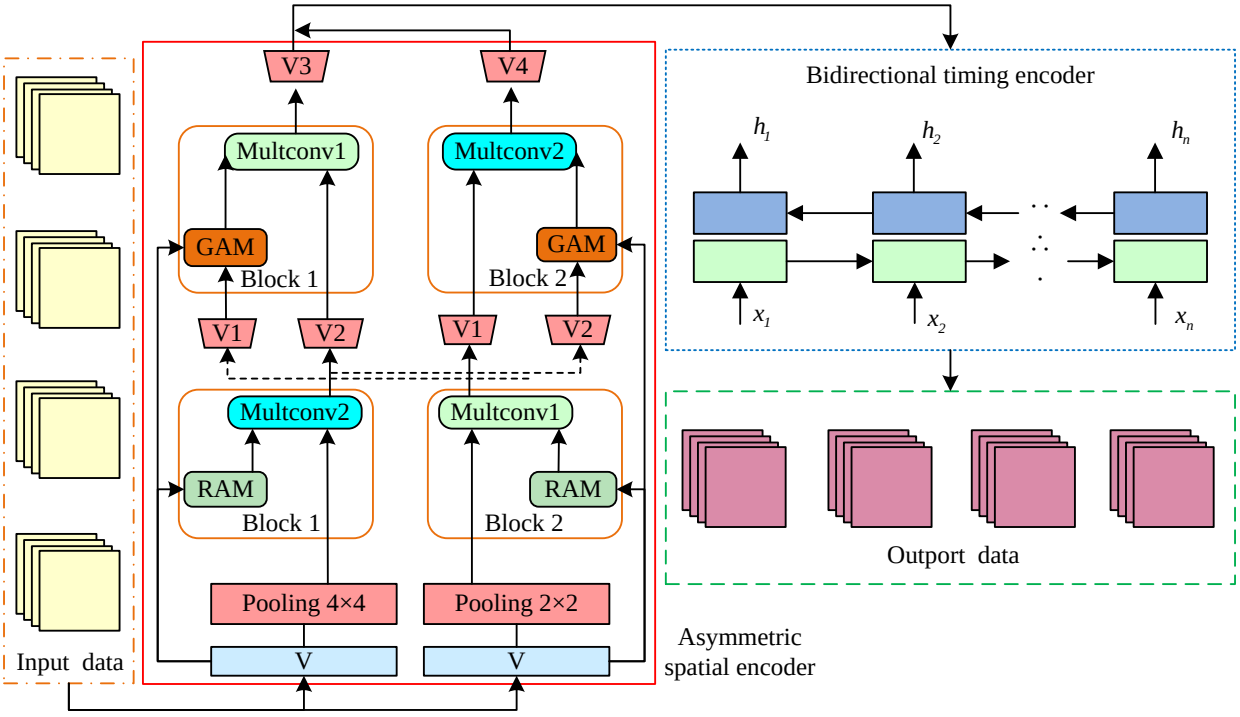


Figure 4: Schematic diagram of feature extraction encoder structure.

As shown in Figure 3, after data augmentation, the MNTI model mainly includes two modules: feature fusion and malicious traffic detection. Among them, the feature fusion module extracts spatial and temporal features of the balanced dataset, and disease fusion presents a spatiotemporal feature set. The malicious traffic detection module first determines whether it is a known attack. If it is a known attack, it directly initiates security warning measures. If not, further unknown attack detection will be conducted. If the traffic detection is abnormal, timely security warning measures should be taken and the network attack database should be updated. In the context of network traffic, spatial features reflect the combination relationship between different features in network traffic data, such as a combination pattern of features such as different IP addresses and different ports. Spatial features can also represent communication relationships between different devices or nodes. The temporal characteristics mainly describe the dynamic changes and patterns of network traffic data in the time dimension, reflecting the changing trends and periodic patterns of network traffic data in time, such as the peak and off peak periods of network traffic, periodic fluctuations in traffic, etc. This study uses CNN structure for spatial feature extraction and RNN suitable for sequence data processing for temporal feature extraction. The structure of the feature extraction encoder designed for the feature fusion module is shown in Figure 4.

In Figure 4, the feature extraction spatial encoder structure consists of an Asymmetric Spatial Encoder (ASE) and a Bidirectional Temporal Encoder (BTE). ASE is used to extract spatial features from raw data. BTE is used to extract temporal features from the extracted spatial features, achieving the effect of fusing features from different dimensions. Finally, spatial and temporal features are fused to form a comprehensive feature representation. ASE is based on traditional CNN architecture, consisting of four blocks that integrate two different types of AMs and convolutional kernels of different scales. Four blocks use two types of multi-scale convolutional layers. Both Multichannel 1 and Multichannel 2 contain three convolutional path calculations and use three various sizes of convolution kernels, namely 3×3, 5×5, and 7×7, to enhance the receptive field of the network. In addition, Block also introduces Global Attention Mechanism (GAM) and Residual Attention Mechanism (RAM). Firstly, RAM is used to fuse multi-scale inputs with the original image, and residual connections can be introduced to enhance the model's generalization ability. Then GAM is used to fuse the output of RAM with the original image. GAM can correlate and weight all positions in the input sequence, enhancing the model's overall understanding and processing ability of the input sequence. The selected basic RNN unit is GRU. GRU refers to a variant structure of RNN that can reduce gradient vanishing while preserving long-term sequence information. The BTE structure is shown in Figure 5.

In Figure 5, the BTE structure has undergone bidirectional improvement on the basis of traditional RNN and introduced multi head self AM. Bidirectional GRU (BiGRU) can extract forward and backward data and determine whether there is abnormal information in the current traffic data [21]. The merging strategy is used to fuse the forward and backward hidden states of BiGRU to generate the final sequence representation. To provide more comprehensive sequence feature information and improve the detection performance of the model, this

study adopts a concatenation strategy, directly concatenating the forward and backward hidden states into a vector. The update process of the forward update gate z_t and reset gate r_t in BiGRU is equation (10).

$$\begin{cases} r_t = \sigma\left(w_n x_t + u_n h_{t-1}\right) \\ z_t = \sigma\left(w_m x_t + u_m h_{t-1}\right) \end{cases} \quad (10)$$

In equation (10), σ represents the Sigmoid activation function, with an output value between 0 and 1. The larger the value, the more information from the previous time step is retained. w_n and w_m represent the weight parameters of the update gate and reset gate, respectively. u_n and u_m represent weight matrices. h_{t-1} represents the previous state. h is the hidden layer state. x_t is the input information at the current time. The calculation of output layer h_t is equation (11).

$$h_t = \left(1 - z_t\right) h_{t-1} + z_t h_t \quad (11)$$

In equation (11), h_t represents the updated value of the reset gate. The reverse calculation formula for BiGRU is equation (12).

$$\begin{cases} z_t^{\,a} = \sigma\left(w_m^{\,a} x_t + u_m^{\,a} h_{t+1}\right) \\ r_t^{a} = \sigma\left(w_n^{\,a} x_t + u_n^{\,a} h_{t+1}\right) \end{cases} \quad (12)$$

In equation (12), a is the reverse GRU. w_m^a and $w_n^{\,a}$ represent weight parameters. $u_m^{\,a}$ and $u_n^{\,a}$ represent weight matrices. h_{t+1} represents the state at the next moment. Finally, the hidden layer states of the forward and reverse GRUs are weighted and summed to obtain the final prediction result, as shown in equation (13).

$$y_t = \sigma\left(h_t \times w_y\right) \quad (13)$$

In equation (13), w_y is the weight between the hidden and output layers. Finally, the predicted temporal results are input into the multi head self-AM to achieve weighted summation of encoding. The detection objects of the traffic detection module include known and unknown network attacks. The known detector for network attacks is SoftMax. The SoftMax expression is equation (14).

$$SoftMax\left(x\right) = \exp\left(x\right) / sum\left(\exp\left(x\right)\right) \quad (14)$$

In equation (14), exp is an exponential function. In the feature fusion module, this study achieves the extraction of spatial and temporal features through CNN and GRU, while reducing the impact of redundant features. The convolutional layer automatically filters local features through convolutional kernels of different scales, while the gating mechanism of GRU filters out irrelevant temporal information. In addition, to further improve the model's performance, this study also ranks the importance of features. In the data augmentation stage, XGBoost is used to rank features and select the top-ranked features for subsequent model training. Meanwhile, in the feature extraction encoder, Genetic Algorithm (GA) and RA are introduced to automatically focus on the more important features for ItruD by learning the weights of features, thus achieving feature importance ranking. The detection model for unknown network attacks is shown in Figure 6.

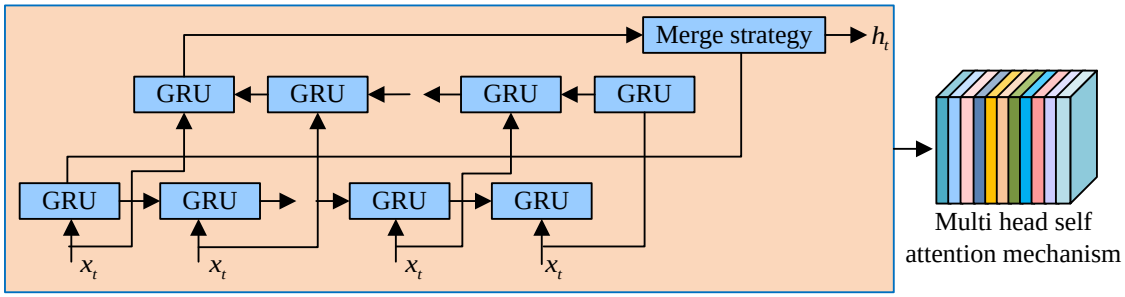


Figure 5: Schematic diagram of BBTE structure.

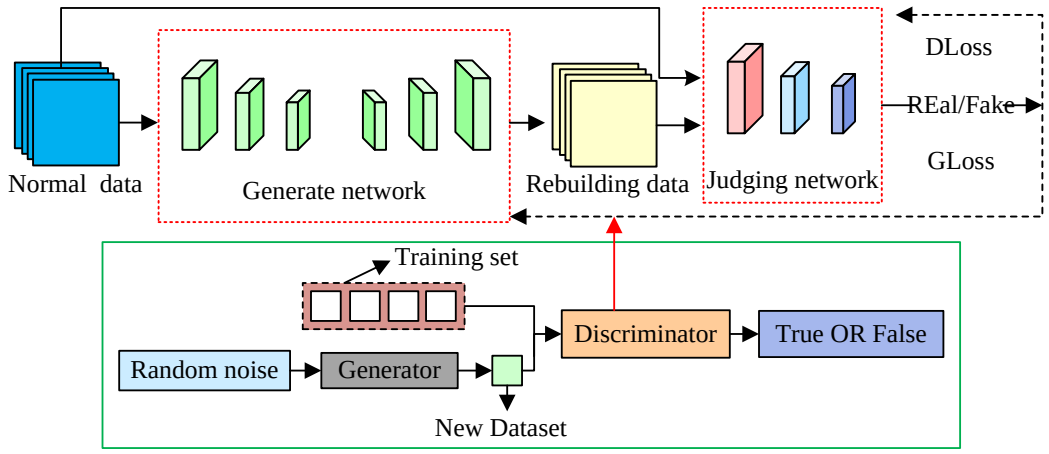


Figure 6: Unknown network attack detection model structure.

In Figure 6, the model is designed based on the concept of GAN and consists of two parts: the generative network and the judgment network. Generate models that produce fake data similar to real samples. The discriminative model is responsible for distinguishing and judging between real data and generated data [22]. In the early stages of training, the weights of the discriminator are randomly initialized. As training progresses, the GAN continuously learns how to generate more realistic data, while the discriminator also updates its parameters based on the feature differences between real and fake data. At the end of the training phase, the generator and discriminator reach Nash equilibrium, and the discriminator's loss tends to stabilize. The anomaly detection threshold is based on a dynamic adjustment strategy. In practical applications, if the False Alarm Rate (FAR) is too high, the threshold should be appropriately increased to reduce misjudgments of normal behavior. If the false alarm rate is too high, the threshold can be appropriately lowered to improve the detection ability of attack behavior. The training game process of GAN is equation (15).

$$\begin{aligned}&\min_{G}\max_{D}V(D,G)=\\&E_{x\sim P_{data}(x)}\left[\log\left(D(x)\right)\right]\\&+E_{z\sim P_{model}(z)}\left[\log\left(1-D\left(G(z)\right)\right)\right]\end{aligned}\tag{15}$$

In equation (15), z represents noise. x is the real sample data. $P_{data}(x)$ is the probability distribution function of x. $P_{model}\left(x^{(i)};\theta\right)$ is the probability distribution function for judging the network, and θ is the parameter. G and D are generative networks and discriminative networks. The detection model is defined as AECE. The generative network part includes encoders and decoders. The encoder and decoder both contain 3 convolutional layers and two pooling layers. The training process needs to make the reconstructed data of the decoder closest to the original data, and use the maximum reconstruction loss of normal traffic behavior as the threshold for detecting unknown attacks. The training process needs to maximize the probability of generating samples as real samples, consisting of convolutional, pooling, and fully connected layers. This study uses Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) as evaluation metrics. MAE, RMSE, and MAPE are commonly used indicators to evaluate the difference between predicted and true values in regression models. In ItruD, they can be used to measure the accuracy of predicting network traffic characteristics, indirectly reflecting the model's ability to distinguish between normal and abnormal traffic. MAE represents the average absolute error between predicted values and true values. RMSE emphasizes the impact of larger errors. MAPE displays model accuracy in the form of relative errors. The calculation of MAE, RMSE, and MAPE is shown in equation (16).

$$\begin{cases}MAE=\dfrac{1}{n}\sum_{i=1}^{n}\left|y_i-\hat{y}_i\right|\\RMSE=\sqrt{\dfrac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}\\MAPE=MAE=\dfrac{1}{n}\sum_{i=1}^{n}\left|\dfrac{y_i-\hat{y}_i}{y_i}\right|\times100\%\end{cases}\tag{16}$$

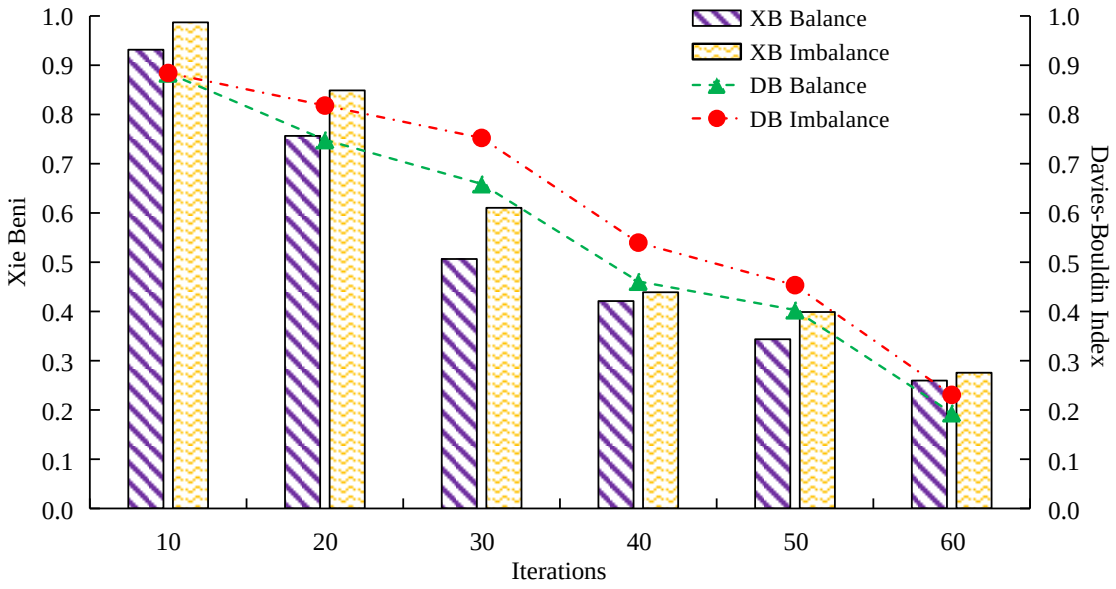
In equation (16), n represents the number of samples, y_i represents the true value, and $\hat{y}_i$ represents the predicted value.

4 Performance testing and application effect analysis of IoT security ItruD model

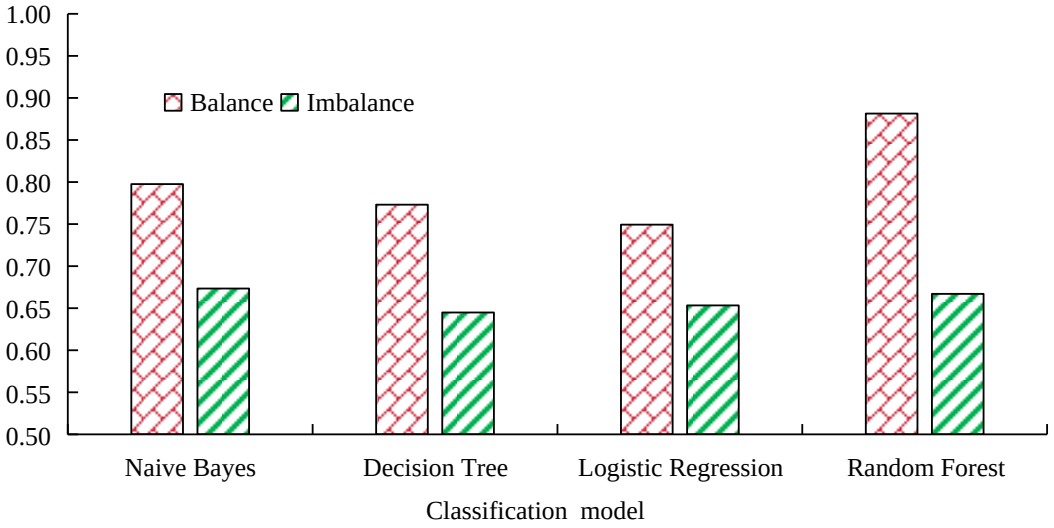
To verify the effectiveness of the designed DAA and MNTI models, this study conducts performance testing and application effect analysis, and discusses the results.

4.1 Performance testing of IoT intrusion detection model

The experiment is conducted using the CentOS 7 operating system and the DL framework is Pytorch 1.7. The central processing unit is Intel (R) Xeon (R) Silver 4214 2.20 GHz, with 128 GB of memory. The image processor is Ge Force RTX 2080Ti. The programming language is Python 3.8. The experiment selects Non-Intrinsic FS for KDD (NSL-KDD), UNSW-NB15, IoT-23, and CSE-CIC-IDS2018 datasets for performance testing. NSL-KDD includes normal connections and various types of attacks, covering multiple characteristics such as connection duration, source/destination ports, service type, protocol type, etc. UNSW-NB15 simulates network traffic in a real network environment, containing 175,341 network connection records, covering common network attacks and normal traffic. IoT-23 contains a large amount of device interaction data, sensor readings, and network communication records. CSE-CIC-IDS2018 contains network traffic data captured from multiple real network environments, covering various types of attacks and normal traffic. The eigenvalues are scaled to the range of 0-1 and divided into training, testing, and validation sets in an 8:1:1 ratio to standardize the data. The learning rate is set to 0.001, Epoch is 60, Batchsize is 32, hidden layer is 2, and Adam optimizer is used. Firstly, the performance of DAA is analyzed, and the clustering and comparison effects before and after data balancing are compared, as shown in Figure 7.



(a) Comparison of clustering effects



(b) Comparison of classification effects

Figure 7: Analysis of the effect of data enhancement algorithm.

Table 2: Results of ablation experiment.

Models	Detection rate	Precision	Recall	MAE
Without feature fusion module	0.856	0.865	0.846	0.214
Without DAA	0.824	0.834	0.813	0.245
Without AECE	0.879	0.887	0.871	0.198
Complete model	0.919	0.925	0.912	0.179

In Figure 7 (a), there is a significant difference in the clustering performance evaluation indicators of the dataset before and after data balancing. The XBI and the DBI both achieve better results on the balanced dataset, with a minimum XBI of 0.259 and a minimum DBI of 0.194, with a decrease of 5.78% and 15.88%, respectively. After DAA processing, the intra cluster compactness and inter class separation of the dataset are improved, and the clustering effect is improved. In Figure 7 (b), four different baseline classification models achieve better classification accuracy on the balanced dataset after data augmentation, with a maximum accuracy improvement of 0.214. To demonstrate the contribution of each component of the model to overall performance, ablation experiments are designed and studied. The ablation experiment uses the NSL-KDD dataset to compare the DR and error metrics of the complete model with models without feature fusion modules, DAA, and AECE. The results of the ablation experiment are shown in Table 2.

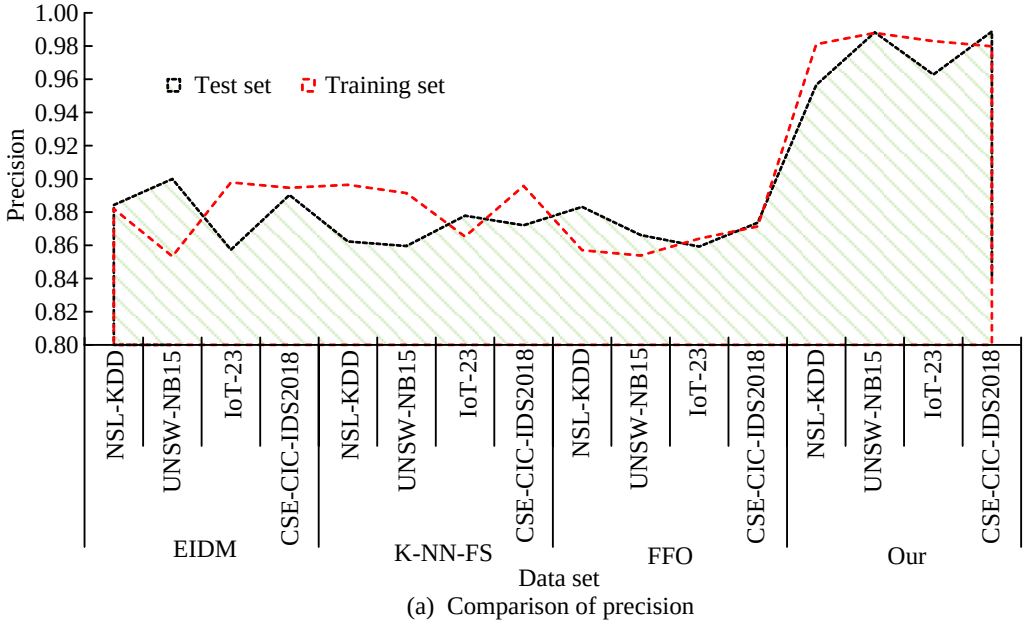
From Table 2, the DR, precision, and recall rate of the complete model are the highest, while the MAE is the lowest, indicating that the proposed improvement

strategies can effectively improve the ItruD performance. Among them, the model without DAA performs the worst in terms of metrics, indicating that DAA contributes the most to model performance and can significantly improve the model's ability to identify attack samples by solving the problem of data imbalance. The MNTI model is compared with the Enhanced Anomaly-based ItruD DL Multi-class Classification (EIDM) proposed in reference [10], the KNN classifier and FS-based ItruD model (K-NN-FS) in reference [11], and the Firefly Optimization (FFO) detection model in reference [15]. Wilcoxon signed rank test is used to evaluate the performance difference between the proposed model and the baseline model, with a $p<0.05$ indicating statistical significance of the difference. To ensure the reliability and stability of the results, each model is independently run 5 times on each dataset. The performance indicators reported are the average of these 5 runs, presented in the form of mean ± standard deviation. The classification performance of different ItruD models is shown in Table 3.

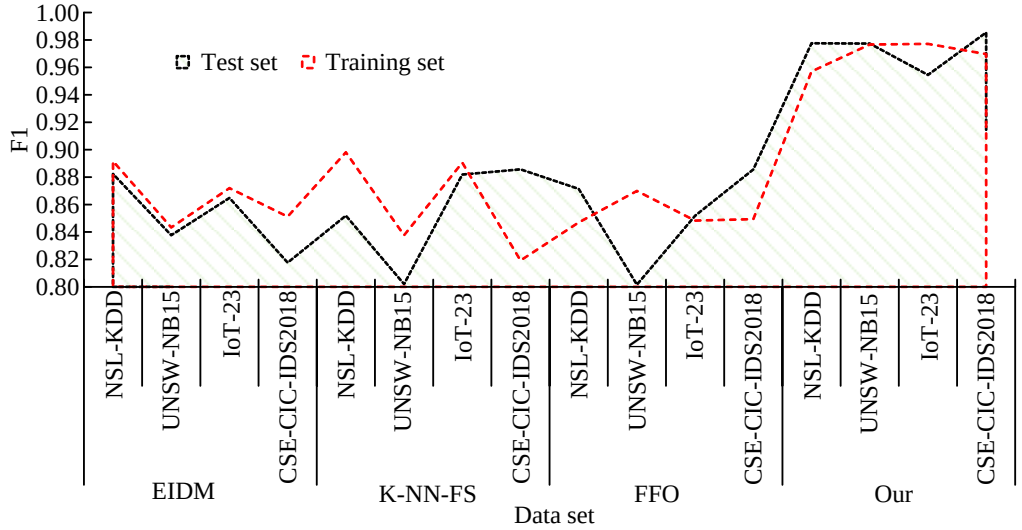
In Table 3, the performance of the proposed model on all four datasets is significantly better than the other three baseline models ($p<0.001$). The research model has the smallest value in the ItruD classification error index, with the minimum values of MAE, RMSE, and MAPE being 0.179, 0.236, and 0.197. The model detection errors of the other three literature are all greater than 0.3. This means that the designed model has the smallest classification error in ItruD and accurately distinguishes traffic between attack behavior and normal behavior. In addition, the DR of the proposed model is the highest, reaching 0.949. The maximum DR values for EIDM, K-NN-FS, and FFO models are 0.885, 0.882, and 0.853. High detection precision means that the model can effectively identify malicious traffic from a large amount of network traffic data, which is crucial for timely detection and response to network attacks. The F1 index is the harmonic mean of precision and recall, used to comprehensively evaluate the performance of a model. Figure 8 compares the scalability of different models.

Table 3: Classification performance of diverse ItruD models.

Model	Index	NSL-KDD	UNSW-NB15	IoT-23	CSE-CIC-IDS2018	*p*-value (vs research model)
Research model	MAE	0.179±0.015	0.198±0.018	0.269±0.022	0.199±0.017	-
	RMSE	0.236±0.020	0.273±0.024	0.286±0.023	0.284±0.026	-
	MAPE	0.199±0.019	0.197±0.019	0.200±0.021	0.261±0.023	-
	DR	0.919±0.013	0.921±0.011	0.949±0.008	0.900±0.015	-
Reference [10]	MAE	0.337±0.033	0.424±0.038	0.325±0.032	0.304±0.029	<0.001
	RMSE	0.430±0.039	0.394±0.037	0.400±0.037	0.392±0.036	<0.001
	MAPE	0.338±0.035	0.442±0.042	0.349±0.033	0.404±0.033	<0.001
	DR	0.729±0.026	0.885±0.018	0.796±0.023	0.823±0.017	<0.001
Reference [11]	MAE	0.419±0.042	0.419±0.040	0.426±0.042	0.325±0.035	<0.001
	RMSE	0.338±0.033	0.437±0.041	0.362±0.035	0.319±0.031	<0.001
	MAPE	0.465±0.045	0.497±0.048	0.359±0.034	0.450±0.043	<0.001
	DR	0.827±0.020	0.842±0.021	0.882±0.018	0.820±0.023	<0.001
Reference [15]	MAE	0.345±0.034	0.490±0.045	0.339±0.036	0.461±0.042	<0.001
	RMSE	0.478±0.047	0.411±0.038	0.416±0.041	0.451±0.043	<0.001
	MAPE	0.351±0.036	0.400±0.039	0.447±0.042	0.379±0.036	<0.001
	DR	0.841±0.023	0.853±0.022	0.828±0.020	0.833±0.022	<0.001



(a) Comparison of precision

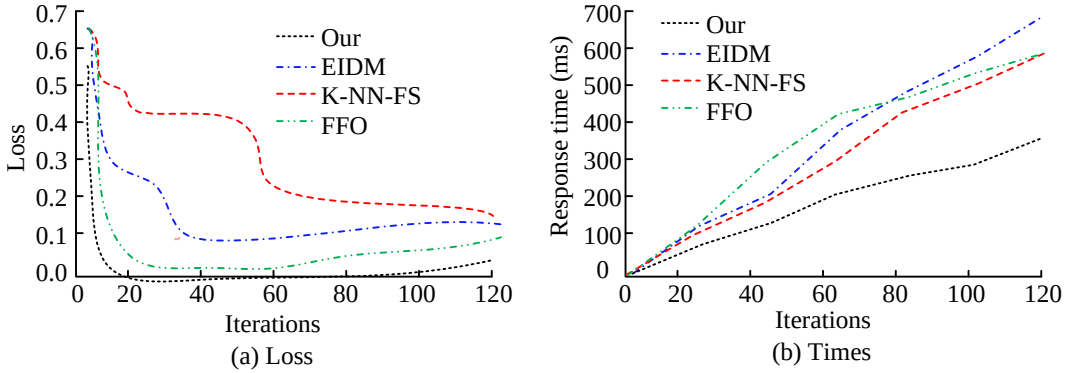


(b) Comparison of F1

Figure 8: Scalability comparison of different ItruD models.

Table 4: The training time of the model on different datasets (s).

Model	NSL-KDD	UNSW-NB15	IoT-23	CSE-CIC-IDS2018
EIDM	158.25	183.49	204.96	198.42
K-NN-FS	92.33	105.56	120.71	112.98
FFO	143.75	165.42	192.04	178.64
Research model	182.43	210.76	244.67	226.28



(a) Loss (b) Times

Figure 9: Comparison of loss function curves and response time for different ItruD models.

In Figure 8, the research model has significant advantages in detection accuracy and F1 index values, and performs well on four different datasets. The maximum precision values on each dataset are 0.981, 0.988, 0.983, and 0.989. The maximum values of F1 index are 0.978, 0.977, 0.977, and 0.985. The difference in values between the test and training sets of the research model is small, and the data fluctuation is not significant. The results indicate that the proposed model can maintain high detection precision and F1 index on different datasets, and has good generalization ability, balance, and stability. This is mainly due to the introduction of data augmentation, feature fusion and extraction, and adversarial training techniques in the model, which significantly improve the performance and scalability of IDS. The training time of the above model on different datasets is shown in Table 4.

From Table 4, compared to the comparison model, the proposed model has a longer training time on all four datasets, with the longest being 244.67 seconds. This is because the architecture of the proposed model is more complex, including feature fusion, IA, AECE, and other components, which increases the complexity of the model and leads to an increase in training time. The baseline model architecture is relatively simple, so the training time is relatively short.

4.2 Performance testing and application effect analysis of IoT ItruD model

It continues to compare the application effects of different ItruD models in practice. The loss function curve and response time of the model are shown in Figure 9.

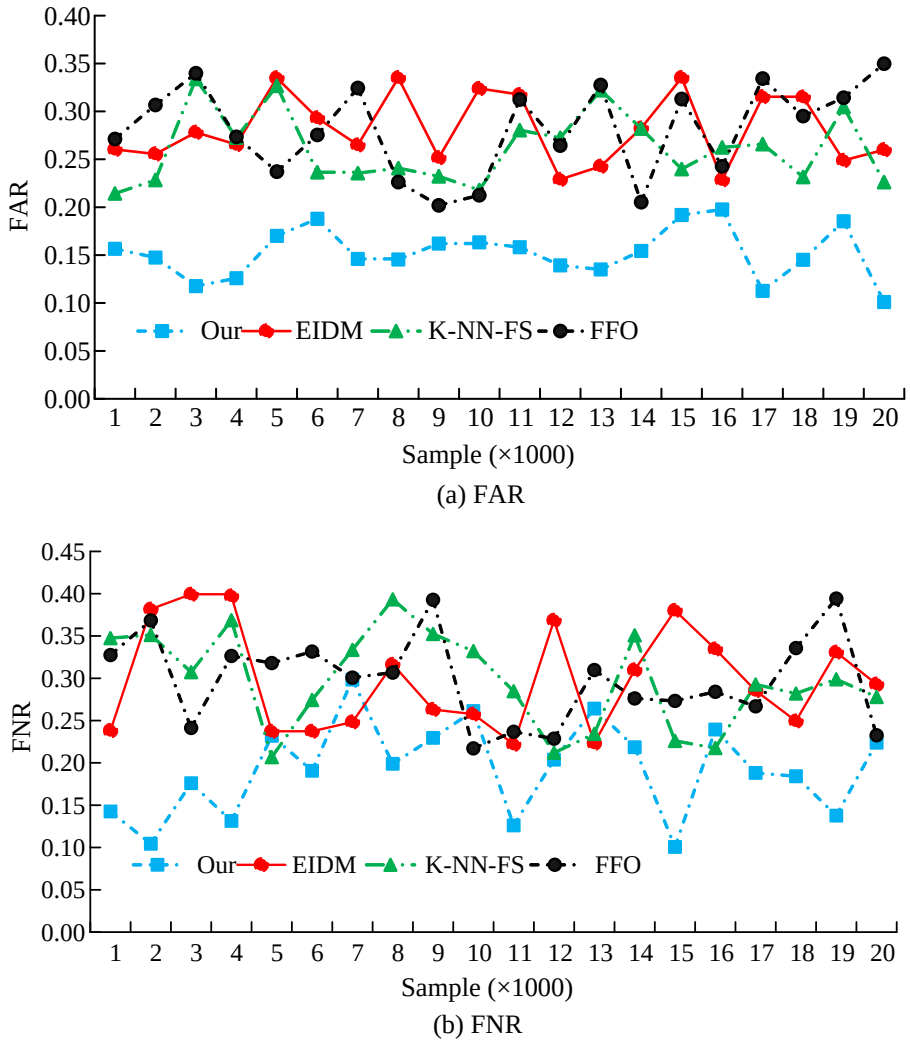
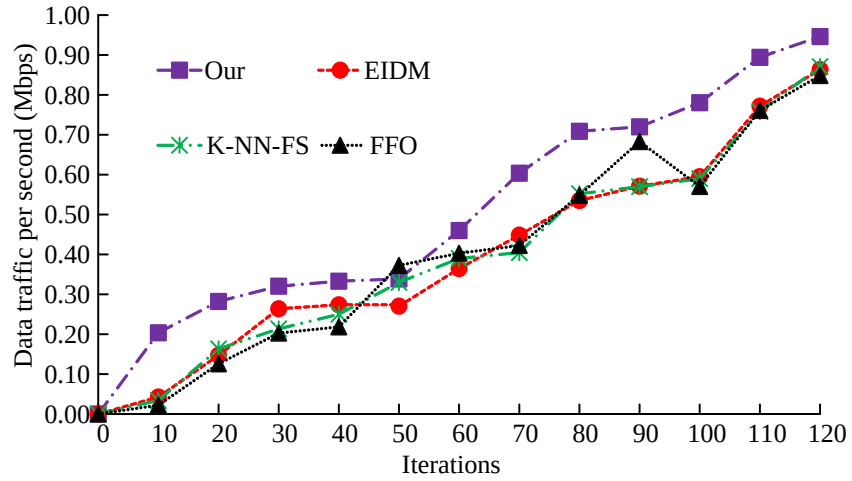


Figure 10: FNR and FAR of various models.

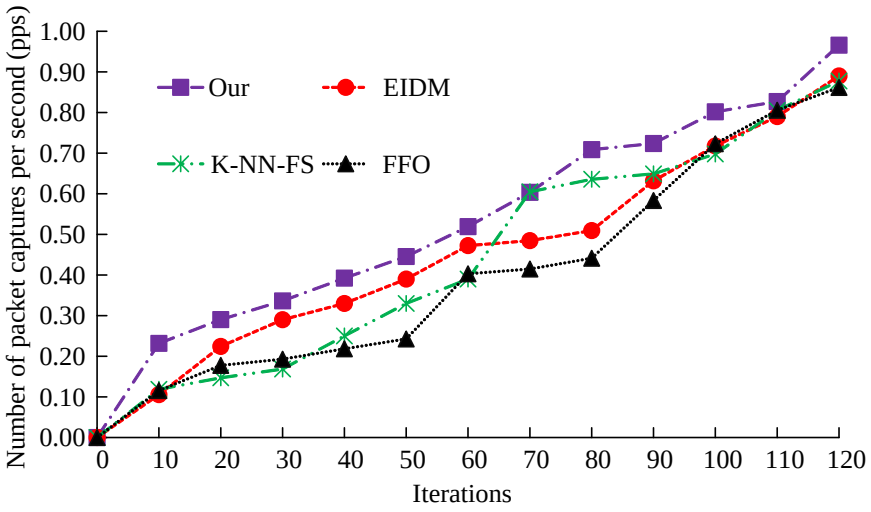
In Figure 9 (a), the research model has the fastest convergence speed on the loss function curve, can converge early in the iteration, and has a minimum convergence value of 0.08, which has a significant convergence advantage over other models. The fast convergence loss function curve indicates that the model can learn features and patterns in the data faster, and does not overfit the training data during the training process, but learns the general features of the data well. In addition, rapid convergence also indicates that the optimization process of the model is more efficient and can achieve the expected performance level in fewer iterations. In Figure 9 (b), the response time of the research model is 368.16ms. The response times of EIDM, K-NN-FS, and FFO models are 684.1 ms, 589.3 ms, and 598.4 ms. A shorter response time means that the model can detect and respond to network traffic faster in practical applications, which is crucial for IoT security IDSs with high real-time requirements. The False Negative Rate (FNR) and FAR of different models in application are displayed in Figure 10.

In Figure 10 (a), the FAR values of the proposed model fluctuate in the range of 0.10-0.20 under different sample sizes. The FAR of other models fluctuates within the range of 0.20-0.35. FAR reflects the tendency of the model to misjudge normal traffic as attack traffic. The proposed model has good recognition performance for normal behavior, with fewer false alarms. In Figure 10 (b), the research model achieves excellent FNR performance, with values fluctuating between 0.10-0.20. The proportion of actual attack samples that can be detected is relatively high compared to all actual attack samples. The results of data traffic per second and packet capture per second for different models are shown in Figure 11.

Figures 11 (a) and (b) show that the research model has the highest values in both data traffic per second and packet capture per second. Overall, the model is capable of processing a large number of data packets per second and has strong packet processing capabilities, reflecting the strong detection ability and efficiency of the research design for attack behavior. Based on Figure 10, this method has a low rate of missed attacks. In Figure 11, there is no packet loss phenomenon for all methods.



(a) Data traffic per second



(b) Number of packet captures per second

Figure 11: Comparison of data traffic per second and packet capture per second for different models.

5 Discussion

To cope with malicious attacks on IoT devices, this study conducted data augmentation based on hybrid sampling and auto-encoder, and constructed an MNTI model using feature fusion on this basis. The experiment showed that after balancing the DAA dataset, the minimum XBI value was 0.259, the minimum DBI value was 0.194, and the decrease was 5.78% and 15.88%, respectively. The classification accuracy of different classification base models has been improved. The minimum values of MAE, RMSE, and MAPE for the research model were 0.179, 0.236, and 0.197, and the maximum DR value was 0.949. The maximum accuracy of this model on four datasets was 0.981, 0.988, 0.983, 0.989, and the maximum F1 index was 0.978, 0.977, 0.977, 0.985. In the application process, the convergence speed on the loss function curve was the fastest, the convergence value was the smallest, and the response time was 368.16 ms. In addition, compared with the baseline models EIDM, K-NN-FS, and FFO, the proposed MNTI model also showed significant advantages in false positives and false negatives. The FAR and FNR values of the proposed MNTI model fluctuated within the range of 0.10-0.20, which could more accurately distinguish between normal and abnormal traffic, thereby reducing the false positive rate. In contrast, the FAR values of the baseline model fluctuated within the range of 0.20-0.35, indicating a relatively high false positive rate.

The DR of EIDM proposed in reference [10] on the NSL-KDD dataset was 0.729, while the MNTI model proposed in the study reached 0.919. On the UNSW-NB15 dataset, the DR of EIDM was 0.885, while the proposed MNTI model was 0.921. The DR of the proposed MNTI model was superior to that of the EIDM model on various datasets. Similarly, the DRs of the ItruD models proposed in references [11] and [15] were also lower than those of the proposed MNTI model. This was mainly attributed to the integration of various advanced DL techniques and ideas in this study, including feature fusion, AMs, and improved GANs. Combining CNN and GRU to extract spatiotemporal features and introducing VAE for dimensionality reduction and feature extraction of data can effectively capture spatial correlations and local

features, and better process sequence data. It can also enrich feature information, enabling the model to more accurately capture key features in network traffic data. The AM can automatically learn the importance of different features, making the model more focused on key features related to ItruD, thereby improving the model's discriminative ability. In addition, the AECE introduces the idea of GAN and utilizes adversarial training between the generative network and the judgment network to further enhance the model's ability to detect unknown attacks.

In practical applications, the proposed MNTI model demonstrates good scalability through its flexible design and modular structure. The feature extraction module can be adjusted according to the type of input data, such as replacing CNN with a network structure suitable for processing specific data types, or adding new feature extraction components to adapt to new data sources. The feature fusion mechanism can effectively integrate feature information from different modules, thereby enhancing the model's ability to process multi-source data. In addition, the depth and breadth of the model can be expanded according to actual needs to further improve its performance and application scope. For example, increasing the number of network layers to capture more complex feature patterns, or adopting multi task learning strategies to simultaneously process multiple related tasks.

6 Conclusion

This study aims to improve the accuracy of identifying malicious network traffic in the IoT environment to cope with malicious attacks on IoT devices. By using MS and VAE for data augmentation, the problem of data imbalance has been effectively solved, providing a high-quality data foundation for model training. On this basis, multiple technologies such as CNN, RNN, AM, and GAN are integrated to construct the MNTI model, which can comprehensively capture the characteristics of network traffic data. Experimental studies have shown that the proposed model has good detection performance and stability, can accurately distinguish between attack behavior and normal behavior of traffic, and has high security protection efficiency and real-time performance. However, the computational complexity of the proposed model is relatively high, and deployment on resource constrained IoT devices may pose certain difficulties. Therefore, in future research, the model structure should be further optimized by using techniques such as model compression and quantization to reduce the computational complexity of the model, making it more suitable for resource constrained IoT environments.

References

[1] Arash Heidari, and Mohammad Ali Jabraeil Jamali. Internet of Things intrusion detection systems: A comprehensive review and future directions. Cluster Computing, 26(6):3753-3780, 2023. https://doi.org/10.1007/s10586-022-03776-z

[2] Oluwadamilare Harazeem Abdulganiyu, Taha Ait Tchakoucht, and Yakub Kayode Saheed. A systematic literature review for network intrusion detection system (IDS). International Journal of Information Security, 22(5):1125-1162, 2023. https://doi.org/10.1007/s10207-023-00682-2

[3] Sampath Rajapaksha, Harsha Kalutarage, M. Omar Al-Kadri, Andrei Petrovski, Garikayi Madzudzo, and Madeline Cheah. Ai-based intrusion detection systems for in-vehicle networks: A survey. ACM Computing Surveys, 55(11):1-40, 2023. https://doi.org/10.1145/3570954

[4] Ankit Thakkar, and Ritika Lohiya. A review on challenges and future research directions for machine learning-based intrusion detection system. Archives of Computational Methods in Engineering, 30(7):4245-4269, 2023. https://doi.org/10.1007/s11831-023-09943-8

[5] Noor Aldeen Alawad, Bilal H. Abed-alguni, Mohammed Azmi Al-Betar, and Ameera Jaradat. Binary improved white shark algorithm for intrusion detection systems. Neural Computing and Applications, 35(26):19427-19451, 2023. https://doi.org/10.1007/s00521-023-08772-x

[6] Emad Ul Haq Qazi, Muhammad Hamza Faheem, and Tanveer Zia. HDLNIDS: Hybrid deep-learning-based network intrusion detection system. Applied Sciences, 13(8):4921-4936, 2023. https://doi.org/10.3390/app13084921

[7] Sydney Mambwe Kasongo. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. Computer Communications, 199(2):113-125, 2023. https://doi.org/10.1016/j.comcom.2022.12.010

[8] Chaimae Hazman, Azidine Guezzaz, Said Benkirane, and Mourade Azrour. IIDS-SIoEL: Intrusion detection framework for IoT-based smart environments security using ensemble learning. Cluster Computing, 26(6):4069-4083, 2023. https://doi.org/10.1007/s10586-022-03810-0

[9] Reza Ghanbarzadeh, Ali Hosseinalipour, and Ali Ghaffari. A novel network intrusion detection method based on metaheuristic optimisation algorithms. Journal of Ambient Intelligence and Humanized Computing, 14(6):7575-7592, 2023. https://doi.org/10.1007/s12652-023-04571-3

[10] Omar Elnakib, Eman Shaaban, Mohamed Mahmoud, and Karim Emara. EIDM: Deep learning model for IoT intrusion detection systems. The Journal of Supercomputing, 79(12):13241-13261, 2023. https://doi.org/10.1007/s11227-023-05197-0

[11] Mouaad Mohy-eddine, Azidine Guezzaz, Said Benkirane, and Mourade Azrour. An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection. Multimedia Tools and Applications, 82(15):23615-23633, 2023. https://doi.org/10.1007/s11042-023-14795-2

[12] Yazeed Alotaibi, and Mohammad Ilyas. Ensemble-learning framework for intrusion detection to enhance internet of things' devices security. Sensors,

23(12):5568-5587, 2023. https://doi.org/10.3390/s23125568

[13] Abdel-Rahman Al-Ghuwairi, Yousef Sharrab, Dimah Al-Fraihat, Majed AlElaimat, Ayoub Alsarhan, and Abdulmohsen Algarni. Intrusion detection in cloud computing based on time series anomalies utilizing machine learning. Journal of Cloud Computing, 12(1):127-143, 2023. https://doi.org/10.1186/s13677-023-00491-x

[14] Ernest Ntizikira, Wang Lei, Fahad Alblehai, Kiran Saleem, and Muhammad Ali Lodhi. Secure and privacy-preserving intrusion detection and prevention in the internet of unmanned aerial vehicles. Sensors, 23(19):8077-8104, 2023. https://doi.org/10.3390/s23198077

[15] Nadir Omer, Ahmed H. Samak, Ahmed I. Taloba, and Rasha M. Abd El-Aziz. A novel optimized probabilistic neural network approach for intrusion detection and categorization. Alexandria Engineering Journal, 72(6):351-361, 2023. https://doi.org/10.1016/j.aej.2023.03.093

[16] Yujun Wang. Deep learning models in computer data mining for intrusion detection. Informatica, 47(4):555-568, 2023. https://doi.org/10.31449/inf.v47i4.4942

[17] Zhenpeng Zhang. SD-WSN network security detection methods for online network education. Informatica, 48(21):51-66, 2024. https://doi.org/10.31449/inf.v48i21.6257

[18] Nour Moustafa, Nickolaos Koroniotis, Marwa Keshk, Albert Y. Zomaya, and Zahir Tari. Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions. IEEE Communications Surveys & Tutorials, 25(3):1775-1807, 2023. https://doi.org/10.1109/COMST.2023.3280465

[19] James Halvorsen, Clemente Izurieta, Haipeng Cai, and Assefaw Gebremedhin. Applying generative machine learning to intrusion detection: A systematic mapping study and review. ACM Computing Surveys, 56(10):1-33, 2024. https://doi.org/10.1145/3659575

[20] S. Sivamohan and S. S. Sridhar. An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework. Neural Computing and Applications, 35(15):11459-11475, 2023. https://doi.org/10.1007/s00521-023-08319-0

[21] Ngamba Thockchom, Moirangthem Marjit Singh, and Utpal Nandi. A novel ensemble learning-based model for network intrusion detection. Complex & Intelligent Systems, 9(5):5693-5714, 2023. https://doi.org/10.1007/s40747-023-01013-7

[22] Md. Alamin Talukder, Selina Sharmin, Md Ashraf Uddin, Md Manowarul Islam, and Sunil Aryal. MLSTL-WSN: machine learning-based intrusion detection using SMOTETomek in WSNs. International Journal of Information Security, 23(3):2139-2158, 2024. https://doi.org/10.1007/s10207-024-00833-z