Article

Not peer-reviewed version

Deep Learning-Based Intrusion Detection for Rare Class Network Attacks

Yu Yang [*] , Yuheng Gu , Yu Yan

Posted Date: 24 August 2023

doi: 10.20944/preprints202308.1665.v1

Keywords: intrusion detection; internet of things; Deep Learning; AutoEncoder; network security

Article

Deep Learning-Based Intrusion Detection for Rare Class Network Attacks

Yu Yang [1]*, Yuheng Gu [1] and Yan Yu [1]

[1] College of Information Engineering, Chinese People's Armed Police Force Engineering University, Xi'an 710086, China

* Correspondence:yuxsky91039@163.com

Abstract: With the continuous development of network technology, complex network systems generate massive unbalanced attack traffic. Due to the severe imbalance in the quantities of normal samples and attack samples, as well as among different types of attack samples, intrusion detection systems suffer from low detection rates for rare class attack data. In this paper, we propose a geometric synthetic minority oversampling technique based on optimized kernel density estimation algorithm. This method can generate diverse rare class attack data by learning the distribution of rare class attack data while maintaining similarity with the original sample features. Meanwhile, the balanced data is input to a feature extraction module built upon multiple denoising autoencoders, reducing information redundancy in high-dimensional data and improving the detection performance for unknown attacks. Subsequently, a soft voting ensemble learning technique is utilized for multi-class anomaly detection on the balanced and dimensionally reduced data. Finally, an intrusion detection system is constructed based on data preprocessing, imbalance handling, feature extraction, and anomaly detection modules, and validated on the NSL-KDD and N-BaIoT datasets. Comparative experiments with baseline models and other state-of-the-art methods demonstrate that the proposed system improves the detection rate of rare class attack data. Furthermore, it achieves a good overall detection rate on the Internet of Things dataset (N-BaIoT), indicating its strong applicability.

Keywords: intrusion detection; internet of things; deep learning; AutoEncoder; network security

1. Introduction

With the rapid development of the Internet of Things (IoT) [1], IoT technology is becoming increasingly widespread in areas such as smart factories [2], autonomous driving [3], and smart cities [4]. One of the main tasks of IoT technology is to sense the surrounding environment through IoT devices and collect target data for relevant devices to take action based on the acquired information. While IoT greatly improves people's production and lifestyle, it also poses serious security risks. Due to the large number of devices typically present in IoT and the uneven residual resources between interconnected devices, general IoT and IoT systems are vulnerable to various network attacks [5], particularly in fields closely related to people's production and lifestyle such as Industry 5.0 [6], autonomous driving, and smart cities. Ensuring the stability and security of IoT systems has become an urgent problem to solve, with common attacks in each field shown in Table 1. Although extensive efforts have been made to equip IoT devices with security tools and defense mechanisms, these security mechanisms are sometimes not suitable due to the heterogeneous nature and computational resource limitations of the huge number of IoT devices, such as complex encryption or identity authentication mechanisms, which may even affect the devices' original tasks [7]. An intrusion detection system (IDS) [8–11] is a lightweight security method that is more aimed at resource-constrained IoT devices and helps to identify unauthorized attack behaviors. It is one of the effective methods to ensure IoT security.

Table 1. Common network attacks in various fields.

Industry 5.0	Autonomous Driving	Smart City	Smart Factory
Ransomware	Jamming	DDoS	APT
Malware	Spoofing	Cyber Espionage	Phishing
APT	Disrupting	APT	Malware
Phishing	Injecting	IoT device hacking	Social Engineering

Compared to traditional rule-based or signature-based intrusion detection methods [12,13], machine learning-based intrusion detection systems (IDS) [14,15] are capable of identifying anomalous attacks by learning substantial amounts of Internet of Things (IoT) security data, while exhibiting notable classification performance. However, when encountering IoT security data with high-dimensional and imbalanced characteristics [16], traditional machine learning-based intrusion detection technology is unable to identify rare and unknown attack data, resulting in slower detection rates, which makes it challenging to meet the security needs of IoT devices. For instance, in IoT security data, anomalous attack data often only accounts for a small portion compared to normal data. When employing traditional machine learning algorithms for classification, the classifier tends to lean towards majority class data [17], leading to the misclassification of rare attack data as normal data. Even though the final classification accuracy may be high, misclassifying rare attack data as normal data poses a severe threat to the security of IoT systems.

The proposed intrusion detection model, KGMS-IDS, is aimed at addressing the challenge of detecting rare-class anomalous and unidentified attacks in high-dimensional and imbalanced IoT data. The primary contributions of this paper are outlined below:

1. Geometric SMOTE (G-SMOTE) enhances the linear interpolation mechanism by introducing geometric transformations in the feature space, allowing for a better approximation of the distribution of minority class samples. The G-SMOTE algorithm is applied to the intrusion detection field, and the Kernel Density Estimation (KDE) algorithm is adopted to improve the G-SMOTE algorithm to handle imbalanced processing in high-dimensional and imbalanced IoT traffic.
2. A feature extraction module -Multi-Noise and Attention Mechanism-based Denoising Autoencoder (MDSAE) is proposed to extract deep feature representations of high-dimensional IoT data, thereby enhancing the robustness of the data after dimensionality reduction.
3. The integration of three modules, KGSMOTE, MDSAE, and Soft Voting Ensemble Model (SVEDM), for multi-category anomaly detection of IoT traffic effectively improves the overall detection rate of the IDS. The ablation experiments show that these modules are interrelated and mutually reinforcing, and the detection performance of the multi-module IDS is better than that of the single-module intrusion detection model. The comparison experiments show that KGMS-IDS has higher overall detection rate and lower false alarm rate compared with other intrusion detection methods.

The rest of this paper is organized as follows: Section 2 provides an overview of current intrusion detection methods. Section 3 describes KGMS-IDS and its modules. Section 4 evaluates the proposed approach through experiments. Section 5 summarizes the research results.

2. Related Work

In recent years, intrusion detection methods based on machine learning (ML) and deep learning (DL) have become a research hotspot in the field of IoT security due to the explosive growth of IoT data. The main objective of ML is to enable computers to learn automatically without human intervention or assistance and subsequently control actions accordingly [18]. Mehmood M et al. [19] employed the Random Forest Recursive Feature Elimination (RFRFE) method for feature selection, which screened out features that had a positive impact on classifier performance, and utilized Fine

Gaussian SVM (FGSVM) for binary anomaly detection on the NSL-KDD dataset achieving an accuracy of 99.3%. Hammad M et al. [20] proposed a polynomial mixture modeling method based on Median Absolute Deviation and Random Forest algorithm (MMM-RF) for classifying network attacks. They employed techniques such as feature selection and dimension reduction by T-distributed Stochastic Neighbor Embedding, and applied Synthetic Minority Over-sampling Technique (SMOTE) and Random Under-sampling to handle imbalance, achieving an abnormal detection accuracy of 99.98% on the CSE-CIC-IDS2018 dataset. To address the uncertainty in the network intrusion detection process, Xie J et al. [21] proposed a network intrusion detection algorithm using Dynamic Intuitionistic Fuzzy Sets (IFSs). Firstly, they used the chi-square test to select the best features, followed by time-series processing to construct dynamic intuitionistic fuzzy patterns from the reduced dataset. Finally, they generated a classifier using the proposed distance measure of dynamic IFS and tested it on the KDD 99, NSL-KDD, and UNSW-NB15 datasets. The experimental results demonstrated that the proposed classification method outperformed traditional single machine learning algorithms.Prajisha C et al. [22] put forward an intrusion detection mechanism for IoT that utilizes the Enhanced Chaotic Crow Search Algorithm (ECSSA) for feature selection and the LightGBM algorithm for classification. The proposed mechanism achieved detection accuracies of 99.38%, 98.91%, and 98.35% on the MC-IoT, MQTT-IoT-IDS2020, and MQTTset datasets, respectively. Kumar R et al. [23] detected DDoS attacks against mining pools in blockchain IoT networks by training Random Forest (RF) and eXtreme Gradient Boosting (XGBoost) systems on distributed fog nodes. The experiment conducted on the BoT-IoT dataset demonstrated that XGBoost performs better in binary attack detection, whereas RF performs better in multi-class attack detection. However, a single traditional classifier is unable to handle high-speed and ever-evolving attacks. To address this challenge, Khan MA et al. [24] proposed an AutoML-based OE-IDS model that uses a soft-voting method to develop the best ensemble strategy for intrusion detection in network environments. The AutoML framework selects the best classifier, and the soft-voting method is used to develop the optimal ensemble strategy, resulting in excellent detection accuracy on the UNSW-NB15 and CIC-IDS2017 datasets. Aburomman A et al. [25] proposed a heterogeneous ensemble classifier that integrates three classifiers: k-nearest neighbors (k-NN), artificial neural network (ANN), and naive Bayes (NB). They employed a weighted majority voting strategy for multi-class anomaly detection, achieving a five-class detection accuracy of 83.43% on the complete NSL-KDD dataset and addressing the problem of unstable single-classifier training.

Despite the advancements in the field of intrusion detection systems (IDS) based on traditional machine learning, there are still limitations regarding their generalization ability, robustness, and adaptability. In the presence of complex classification problems characterized by high-dimensional and imbalanced IoT security data, the accuracy of these systems often fails to meet user demands. To overcome this issue, Kunang YN et al. [26] proposed a deep learning-based IDS that utilizes a deep autoencoder (PTDAE) for feature extraction and detects reduced data in a deep neural network (DNN). The proposed approach attained effective multi-class detection results on the NSL-KDD and CSE-CIC-ID2018 datasets. Lv Z et al. [27] proposed a hierarchical intrusion security detection model composed of a self-encoding three-layer neural network and a stacked denoising autoencoder-support vector machine (SDAE-SVM). This model reduces the load of IoT-based intrusion detection and enhances detection performance. Zhang Y et al. [28] presented an IoT intrusion detection method based on the Improved Conditional Variational Autoencoder (ICVAE) and Boundary Synthetic Minority Oversampling Technique (BSM). Through ICVAE, the method learns the posterior distribution of different class samples, and uses BSM to synthesize rare attack samples. The balanced data is then input into the softmax classifier, effectively improving IoT attack detection accuracy under sample imbalance conditions.Andresini G et al. [29] represented network flows as 2D images and utilized GANs to learn their distribution, which enabled them to generate rare class attack data. Finally, they input the balanced 2D image data into a convolutional neural network (CNN) for classification detection, and achieved significant detection results on four benchmark datasets. On the other hand, Kumar V et al. [30] generated rare class attack samples using Wasserstein Conditional Generative

Adversarial Network (WCGAN), trained an XGBoost classifier on the balanced dataset, and tested it on three benchmark datasets - NSL-KDD, UNSW-NB15, and BoT-IoT - where they obtained notable detection rates.

While the aforementioned methods have increased the detection rate of IoT attack data, their performance in training and detecting unknown attacks in the IoT is not always optimal. This article proposes an IoT intrusion detection method based on MDSAE and KGSmote that incorporates previous research to improve model stability. Multi-class experiments were conducted using SVEDM. Table 2 compares the detection methods of various models mentioned above with the proposed method presented in this study.

Table 2. Comparison with relevant survey results.

Methods	Datasets	ML/DL	Ensemble leaning	Unknown attack	Year
RFRFE-FGSVM [19]	NSL-KDD	ML	×	×	2022
MMM-RF [20]	CSE-CIC-IDS2018	ML	×	×	2022
IFSs [21]	KDD 99/NSL-KDD UNSW-NB15	ML	×	×	2021
ECSSA-LightGBM [22]	MC-IoT/MQTTset MQTT-IoT-IDS2020	ML	×	×	2022
RF/XGBoost [23]	BoT-IoT	ML	×	×	2022
OE-IDS [24]	UNSW-NB15 CIC-IDS2017	ML	✓	×	2023
ACOR-WMV [25]	NSL-KDD	Both	✓	×	2022
PTDAE-DNN [26]	NSL-KDD CSE-CIC-ID2018	DL	×	×	2021
SDAE-SVM [27]	NSL-KDD	Both	×	×	2020
ICVAE-BSM [28]	NSL-KDD/CIC-IDS2017 CSE-CIC-IDS2018	Both	×	×	2022
GAN-CNN [29]	KDDCUP99/UNSW-NB15 CIC-IDS2017/AAGM17	DL	×	×	2021
WCGAN-XGBoost [30]	NSL-KDD/BoT-IoT UNSW-NB15	Both	×	×	2023
Proposed method	NSL-KDD/N-BaIoT	Both	✓	✓	2023

3. Method

3.1. Model Structure

The structure of KGMS-IDS proposed here isshown in Figure 1, and the parameters of each module are shown in Table 3. it mainly consists of three modules: imbalance processing module, feature dimensionality reduction module and classification module, and the detailed parameters of each module are shown in Table 3. The main contributions and roles of each module of KGMS-IDS are as follows. The KGSMOTE module is used to generate rare class attack data, which makes a major contribution to improving the detection rate of KGMS-IDS for the MDSAE module can reduce the information redundancy of the original high-dimensional data, and at the same time improve the robustness of the reduced-dimensional data through multiple noise, which makes a major contribution to improving the detection capability of KGMS-IDS for unknown attacks. The SVEDM module is the last module of KGMS-IDS, which implements the classification of the improved data to detect anomalous attacks among them. The experiments show that the KGMS-IDS modules are interrelated and contribute to each other, and each module contributes to improving the detection capability of the system. Overall, KGMS-IDS is able to effectively improve the detection rate of rare class attacks and unknown attacks in IoT through the proposed KGSMOTE and MDSAE, combined with SVEDM. The specific workflow for the implementation and integration of the modules in KGMS-IDS is divided into the following four steps.

1. Data pre-processing module: The training and test sets are input into the data pre-processing module, and the data are cleaned and transformed to form clean data for model training. Firstly, the data are processed by missing values and outliers, and the irregular data in the original data such as the rows containing None, NaN, inf and nan in the numerical feature columns are removed. Secondly, the MinMaxScaler method is used to normalize the cleaned data and limit the pre-processed data to [0,1]. Finally, the one-hot method is used to transform the discrete features in the data into a vector group of 0,1 combinations. The data after the data pre-processing module is input to the next imbalance processing module for imbalance processing.
2. Imbalance processing module: The imbalance processing module is mainly based on random downsampling algorithm and KGSMOTE algorithm. The training set in the data after pre-processing is taken out, and the training set isinput into the imbalance processing module based on KGSMOTE. The majority class traffic in the dataset is first randomly downsampled, and then the rare class attack data is generated by the KGSMOE algorithm. It should be noted that the KGSMOTE model only does imbalance processing on the training set to meet the requirements of an IDS deployed in a real IoT environment. The data after the imbalance processing module is input to the feature downsampling module for feature downsampling.
3. Feature reduction module: Input the training data processed by the imbalance processing module into the MDSAE-based feature reduction module to train the MDSAE model. The encoder part of the trained MDSAE model is taken out and the trained parameters are kept. The trained encoder is then used to perform feature downscaling on the training and test sets of the IoT dataset respectively. The dimensionality reduction removes the redundant information from the original high-dimensional data and improves the robustness of the data. The processed data from the feature dimensionality reduction module is input to the classification module to detect multi-class anomalous attacks.
4. Classification module: First, the SVEDM-based classification module is trained using the training dataset processed by the dimensionality reduction module. Then the test dataset are input to the trained classification module for multi-classification anomaly detection, and the final detection results are obtained.

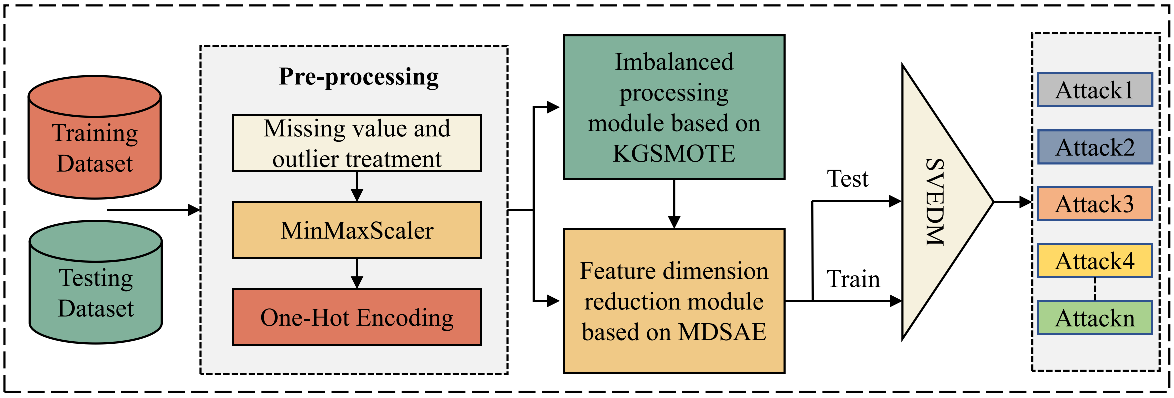

Training Dataset
Testing Dataset
Pre-processing
Missing value and outlier treatment
MinMaxScaler
One-Hot Encoding
Imbalanced processing module based on KGSMOTE
Feature dimension reduction module based on MDSAE
Test
Train
SVEDM
Attack1
Attack2
Attack3
Attack4
Attackn

Figure 1. Framework of the proposed model.

Table 3. Parameters of KGMS-IDS.

Module	Parameter settings	

KGSmote	K(x)=Gaussian kernel
	bandwidth=0.2
	truncation_factor=1,sampling_rate=0.8/0.3
	k_neighbors=5	

MDSAE	Batch size=1024
	Optimizer=Adam,learning rate=0.001
	Epoch=50
	Activation=Relu
	Loss function=Huber Loss
	Hidden layer1=80,Hidden layer2=30	

SVEDM	XGBoost
	(weights=0.286)
		max_depth=10
		learning_rate=0.4
		subsample=0.8
		n_estimators=400

	RF
	(weights=0.571)
		n_estimators=100
		max_depth=10

	C4.5
	(weights=0.143)
		n_estimators=100
		max_depth=10

3.2. Imbalanced Data Processing Module Based on KGSmote

In IoT data, the imbalance between normal and attack traffic makes classifiers prone to misclassify rare-class attack data as normal data. To address this issue, we adopt the Random Under-Sampling (RUS) algorithm to undersample majority-class data and filter out redundant samples. Meanwhile, we use our proposed KGSMOTE model to oversample rare-class attack samples, generating new rare-class attack samples and improving the detection rate of rare-class attacks.

SMOTE [31,32], Borderline-SMOTE, ADASYN, and other algorithms [33,34] are classic oversampling methods used to deal with the issue of imbalanced data classification. Their core idea is to randomly select several nearest neighbor samples for each rare-class attack sample, and perform interpolation operations among these samples based on their distances to generate new instances. These new instances are added to the training set as rare-class attack samples, thereby increasing the number of rare-class attack data. However, they all generate synthetic samples along the line segment connecting minority class samples, making it difficult to improve the distribution of minority class samples. G-SMOTE [35], unlike other SMOTE-based algorithms, extends the linear interpolation mechanism by introducing a geometric region (a hypersphere). It selects a safe radius around each rare-class attack sample and synthesizes rare-class attack data within a hypersphere. Typically, the geometric region of the input space is a truncated hyperellipsoid. Specifically, G-SMOTE replaces synthesizing new rare-class attack data along the line segment connecting minority class instances by defining a flexible geometric region around each selected rare-class attack sample, and increases the diversity of generated samples by expanding the area of minority class.

While the G-SMOTE algorithm largely addresses the problems of generating noisy data and excessive interpolation in some samples, the severe imbalance between normal and attack traffic in IoT data and the small sample size of rare data pose challenges. When using only the G-SMOTE algorithm to synthesize new rare-class attack samples, it is difficult to focus on the most important information, and the distribution of synthesized new samples may not be diverse enough. The KGSMOTE module integrates KDE into G-SMOTE to generate more diverse and fitting data for original rare-class attack samples, as shown in Figure 2.

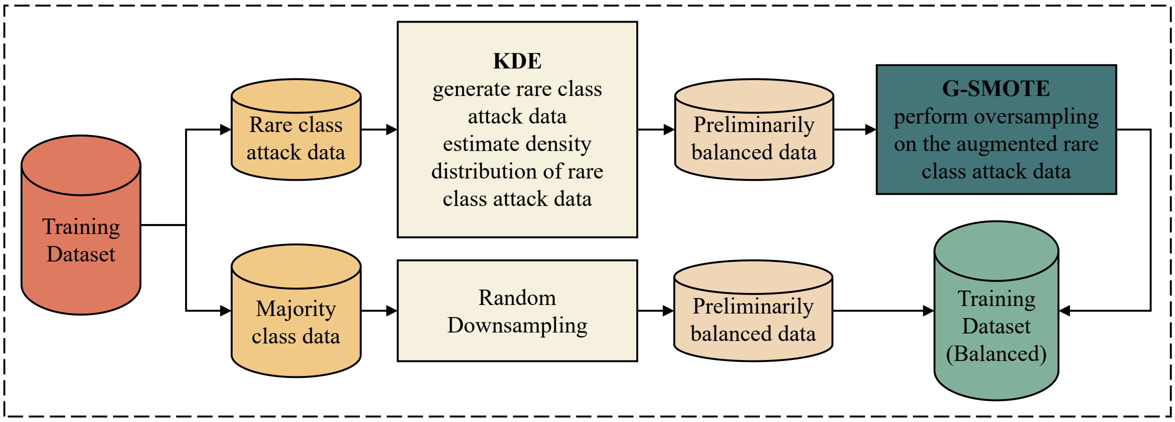

Training Dataset
Rare class attack data
Majority class data
KDE
generate rare class attack data
estimate density distribution of rare class attack data
Random Downsampling
Preliminarily balanced data
Preliminarily balanced data
G-SMOTE
perform oversampling on the augmented rare class attack data
Training Dataset (Balanced)

Figure 2. Imbalanced data processing module based on KGSMOTE.

Firstly, the KDE [36] is used to estimate the probability density function of rare-class attack samples. Then, the generated rare-class attack data distribution is extracted from this probability density function, as shown in Equation (1). This method essentially characterizes the probability distribution of data sample points, estimates the probability density of an unknown distribution without relying on any prior assumptions, and uses a smooth function to approximate this probability density. Here, xi represents the sample data, K(x) is the kernel function, h is the bandwidth parameter, and $\hat{f}_h(x)$ is the estimated probability density at x. In this paper, the Gaussian kernel is selected as the kernel function, and the bandwidth parameter is set to 0.2. The data distribution generated by KDE can capture the key information of rare-class attack samples and increase the diversity of the minority class attack distribution. Secondly, the expanded rare-class samples are input to the G-SMOTE algorithm to generate new minority class attack data. On the one hand, the data distribution generated by KDE can capture the key information of rare-class attack samples. On the other hand, the data distribution sampled by KDE is input into G-SMOTE to expand the diversity of newly generated minority class attacks, thereby improving the recall rate of the detection module for minority class attacks and enhancing the generalization performance of the imbalance processing module. Algorithm 1 illustrates the operation process of the KGSMOTE module.

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{1}$$

Algorithm 1: Oversampling algorithm of KDE-based G-SMOTE

```
Input:Rare class attack data R={r1,r2,r3,...,rn}
Output:Augmented rare class attack data
1:Use KDE to estimate density distribution of rare class attack data
density = kernel_density_estimation(rare_class_attack_data, h)
2:Generate new rare class attack data based on the estimated density distribution
new_rare_class_attack_data = generate_data_from_density(density)
3:Combine new rare class attack data with original training data
training_data = combine(original_training_data, new_rare_class_attack_data)
4:Use G-SMOTE to perform oversampling on the augmented rare class attack data
oversampled_data = G_Smote(training_data)
5:Return oversampled_data
6:End
```

3.3. Feature Dimension Reduction Module Based on MDSAE

The autoencoder (AE) [37,38] is an unsupervised neural network model widely used for feature extraction and anomaly detection. It can extract important features from high-dimensional raw data as input for the next stage of training or testing to achieve the goal of data dimensionality reduction.

However, AE is a shallow neural network and cannot extract deep feature representations of the original high-dimensional data. Moreover, the low-dimensional data extracted by AE is prone to overfitting, which cannot effectively improve the multi-classification accuracy of the detection model. The denoising autoencoder (DAE) [39,40] is an improved version of AE. By injecting noise at the input end of the autoencoder, it can enhance the generalization effect of the data after dimensionality reduction and to some extent increase the robustness of the detection model.

However, it cannot effectively utilize the correlation between various features of IoT data. Furthermore, in denoising autoencoders with only one noise source, the training results may be affected by this noise source, which may lead to poor detection performance when facing unknown attacks on IoT data. To address this issue, we propose a feature dimensionality reduction model MDSAE based on multiple noise and attention mechanisms. Firstly, we add multiple noise sources (Gaussian Noise, Salt and Pepper Noise, Dropout Noise) at the input end of the model to comprehensively consider the feature extraction ability under different noisy conditions. Secondly, we incorporate self-attention mechanism into the hidden layer of the encoder to utilize the correlation between high-dimensional features, and use the importance degree to focus the neural network's attention more on important information among various features. Finally, we deepen the autoencoder's layers and use a deep neural network to learn the deep feature representation of high-dimensional features to improve the robustness of the dimensionality reduction model. Specifically, we input high-dimensional data into MDSAE, perform encoding and decoding operations inside the network to obtain new data representations, then optimize the network parameters through the backpropagation algorithm to make the decoder's output as close as possible to the original data without noise injection. After training, we save the encoder's parameters, input the test data into the encoder, and obtain the reduced feature data. The encoding and decoding process can be represented by formulas (2) and (3).

$$z = f\left(\omega_e x + b^e\right) \tag{2}$$

$$y = g\left(\omega_d z + b^d\right) \tag{3}$$

Equation 2 demonstrates how the encoder compresses high-dimensional raw data into latent features, where we and be represent the weight matrix and bias vector of the encoder, respectively. Equation 3 illustrates how the decoder restores low-dimensional latent features to high-dimensional raw data, where wd and bd are the weight matrix and bias vector of the decoder, respectively. The encoder and decoder utilize activation functions f and g, respectively. Figure 3 displays the basic structure of the Multi-Source Denoising Autoencoder module. From Figure 3, it can be observed that the MDSAE consists of an encoder and a decoder. The encoder includes multiple noise sources, an input layer, a hidden layer (z_1), a self- attention mechanism layer, and a final layer (z). The hidden layer (z_1) contains 80 neurons, while the final layer (z) contains 30 neurons. The decoder, which is symmetric to the encoder, does not have noise sources or a self-attention mechanism layer. Its purpose is to reconstruct the original data as accurately as possible. During the reconstruction process, the decoder continuously trains the neural network, allowing the final layer (z) to effectively represent the original high-dimensional data. The training of the MDSAE involves finding the optimal values of w, minimizing the reconstruction loss functions L with respect to be and bd using standard backpropagation algorithms. It is important to note that the MDSAE utilizes triple noise sources, which are injected only at the input end of the encoder during training. However, when decoding to restore the original data information, the decoder is trained using the noise-free original data, explaining why the dimensionality-reduced data after MDSAE exhibits improved robustness. Additionally, the MDSAE module employs the Huber Loss [41] as the loss function. Finally, by extracting the trained encoder part from the MDSAE, high-dimensional features are reduced to 30 dimensions.

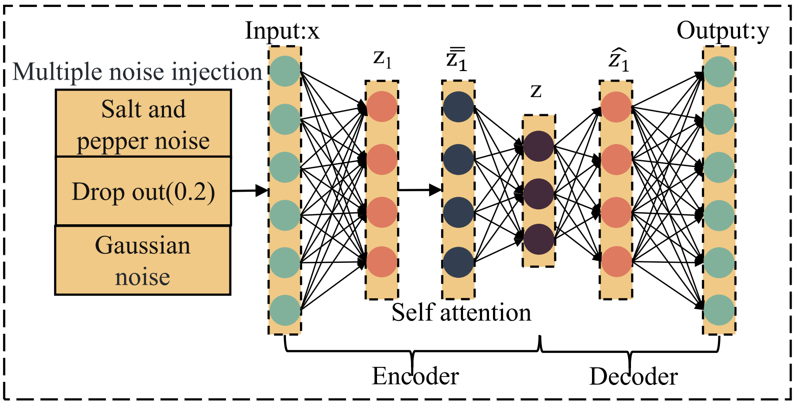


Figure 3. Traditional autoencoders and multiple denoising autoencoders improved by the self-attention mechanism.

3.4. A Multi-Class Anomaly Detection Module Based on SVEDM

The Soft Voting Ensemble Model (SVEDM) is an advanced machine learning algorithm that is effectively used for multi-class anomaly detection. SVEDM takes the weighted average of probability values from various classifiers and based on that, makes a final classification decision. For each sample that is to be classified, a class probability output is obtained from each base classifier, and then these probabilities are averaged. The class with the highest average probability is selected as the final predicted result. Compared to single machine learning classifiers and hard voting ensemble models, SVEDM reduces the error of individual classifiers by merging the predictions of multiple classifiers, making it less susceptible to random noise or interference. If the model is underfitting or overfitting, the addition of different classifiers can improve the generalization performance and robustness of SVEDM. After experimenting with various combinations of base classifiers (SVM, DT, LightGBM, C4.5, RF, Adaboost, XGBoost, Naive Bayes, Logistic Regression), XGBoost, RF, and C4.5 are chosen as the base classifiers for SVEDM. The flow chart of SVEDM is depicted in Figure 4.

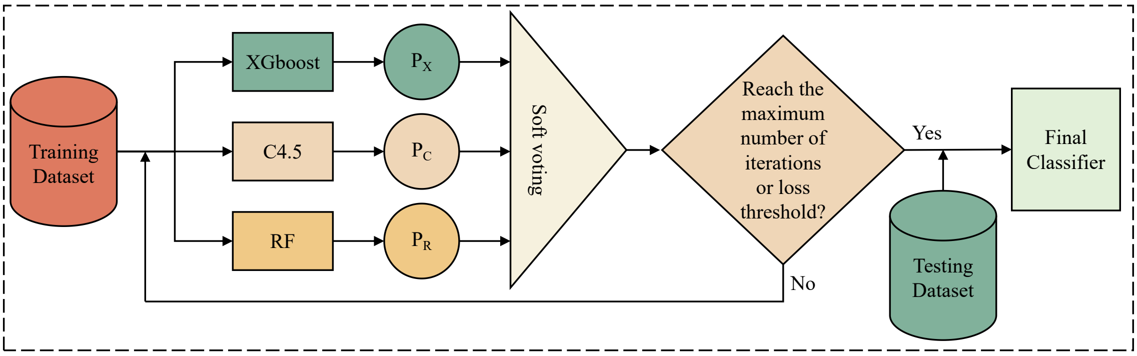


Figure 4. A multi-class anomaly detection module based on SVEDM.

3.5. Dataset Description

Evaluating the performance of IDS is a crucial matter. To address this, the experiment utilizes two network security datasets, NSL-KDD and N-BaIoT, which are effective in portraying the current state of IoT security.

3.5.1. NSL-KDD

The NSL-KDD dataset [42] is an improved version of the KDD Cup 99 [43]. NSL-KDD eliminates duplicate and incomplete records from the KDD Cup 99 dataset, improving data accuracy. It also includes additional network attack types and real network traffic to enhance the dataset's richness and similarity to real-world network environments. With improved labeling accuracy, the dataset is more reliable for intrusion detection and better reflects real-world network intrusion scenarios. The NSL-KDD dataset consists of two training sets (KDDTrain+ and KDDTrain+_20Percent) and two testing sets (KDDTest+ and KDDTest-). The training set is KDDTrain+, and the test set is KDDTest+. Table 4 shows that the test set includes unknown attacks not present in the training set. The normal and attack data in the NSL-KDD dataset are severely unbalanced, with only 52 instances of U2R rare attacks compared to 67,343 normal instances in the training set, resulting in a ratio of 1295:1. This imbalance aligns with the security situation in real IoT environments, but it can significantly impact the classifier's judgment, leading to misclassification of rare attack data as other attack classes or even normal data, posing a severe threat to users. Additionally, KDDTrain+ includes 22 attack types and 1 normal data type, while KDDTest+ has a total of 37 attack data types and 1 normal data type. Notably, KDDTest+ includes 17 attack types absent from KDDTrain+ (i.e., unknown attacks), which tests the model's ability to detect unknown attacks.

The NSL-KDD dataset contains a total of 41 feature columns and 1 label column. Out of the 41 feature columns, 38 are numerical features while the remaining 3 are categorical features. These 3 categorical features are converted into numerical features through one-hot encoding to make them suitable for model training. Consequently, the data dimension is expanded from 41 to 122 dimensions.

Table 4. Categories and Partitioning of the NSL-KDD Dataset.

Class	KDDTrain+	Number	KDDTest+	(Unknow attack)	Number
Normal	normal	67343	normal	\	9711
DoS	back,land,neptune, pod,smurf,teardrop	45927	back,land,neptune, smurf,teardrop,pod	apache2,mailbomb, processtable,udpstorm	7458
Probe	ipsweep,nmap, portsweep,satan	11656	ipsweep,nmap, portsweep,satan	saint, mscan	2421
R2L	buffer_overflow, loadmodule,perl,rootkit	995	buffer_overflow, rootkit,perl, loadmodule	xterm,sqlattack, ps,httptunnel	2754
U2R	ftp_write,guess_passw, imap,warezmaster,spy multihop,phf,warezclient	52	ftp_write,imap guess_passwd,phf warezmaster,multihop	snmpgetattack,worm xlock,sendmail, xsnoop,named,snmpguess	200
Total	23	125973	21	17	22544

3.5.2. N-BaIoT

The N-BaIoT dataset [44,45] is specifically developed for intrusions detection in IoT devices. It contains normal and malicious traffic data collected from various IoT devices, including doorbells, thermostats, baby monitors, cameras, and more. The data is collected from both public networks and LAN environments within the laboratory. The N-BaIoT dataset is comprised of 115 feature columns and one label column. Its attack data includes ten categories primarily from two botnets (BASHLITE and Mirai), which can be divided into three major categories (Normal, BASHLITE, and Mirai) and eleven subcategories. We utilized traffic extracted from an intelligent video surveillance camera (Provision PT-737E) as our experimental data. During the partitioning of the dataset into training and testing sets, we added attack types that were not present in the training set to the test set, as shown in Table 5.

From Table 5, it's evident that the test set displays three types of attacks (TCP flooding, Scan (Mirai), UDP (Mirai)) that were not present in the training set. Additionally, there is an imbalance between normal and attack data, with the former being more dominant. This scenario mirrors the security conditions prevalent in real-world IoT environments.

Table 5. Categories and Partitioning of the N-BaIoT Dataset.

Class	N-BaIoT Train	Number	N-BaIoT Test	(Unknow attack)	Number
Normal	normal	34806	normal	\	14917
BASHLITE Attack	Scan(BASH),Junk COMBO,UDP(BASH)	6869	Scan(BASH),COMBO Junk,UDP(BASH)	TCP flooding	5778
Mirai Attack	Ack,Syn,UDPplain	6051	Ack,Syn,UDPplain	Scan(Mirai),UDP(Mirai)	5663
Total	8	47726	8	3	26358

4. Experimental Results and Analysis

This section focuses on the experimental results obtained from using the proposed KGMS-IDS architecture. Ablation experiments were conducted to analyze the significance and roles of individual modules. The experiment was conducted on a personal computer, and Table 6 illustrates the overall configuration.

Table 6. Experimental operating environment.

Project	Parameters
CPU	Intel Core i7-11800H 2.30GHz
GPU	NVIDIA RTX3070
Python version	3.9.13
TensorFlow version	2.8.0
Keras version	2.8.0
Pytorch version	1.10.1

4.1. Evaluation Metrics

The amount of normal and attack data within IoT environments is significantly imbalanced, which is a characteristic demonstrated by both the NSL-KDD and N-BaIoT datasets. In IoT intrusion detection, the detection model can still exhibit high accuracy even when rare attack data is incorrectly categorized as normal data. However, mistakenly classifying rare attack data as normal data can also pose a threat to IoT security, thus, this study selected four classification evaluation metrics, including accuracy, recall, precision, and F1 score. The formula for the evaluation metric is as follows:

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

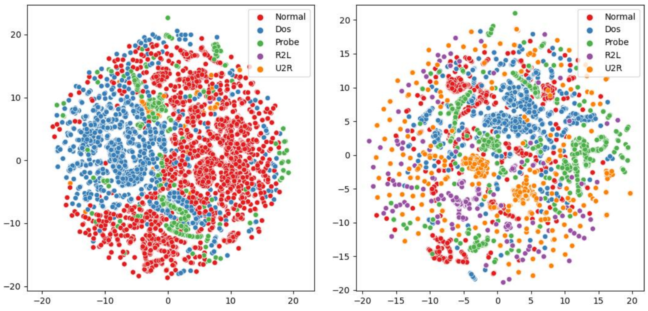
$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{7}$$

Accuracy (Acc) represents the proportion of samples that the classifier correctly classified among all the samples, in other words, the ratio of correct predictions made by the classifier to all samples. Precision (Pre) represents the proportion of truly positive samples predicted as positive by the classifier, out of all samples predicted as positive by the classifier. Recall (Re) represents the proportion of true positive samples correctly predicted by the classifier among all the positive samples. The F1-score (F1) is a comprehensive metric that takes into account both Precision and Recall metrics and calculates a weighted harmonic average between them.

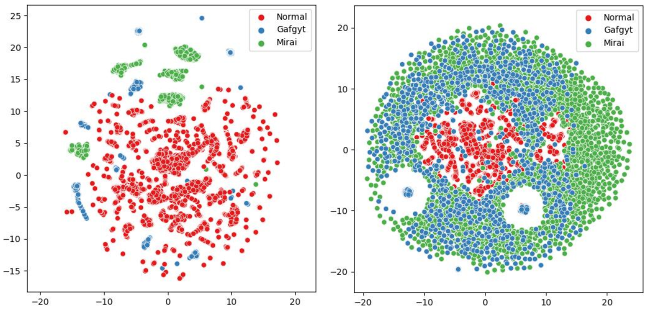
4.2. Imbalanced Processing Based on KGSMOTE

First, the training data is divided into a training set and a test set. NSL-KDD uses the pre-divided KDD Train+ and KDD TEST+ as the training and test sets, respectively. As the creators of N-BaIoT did not divide the training and test sets, this paper uses the newly divided N-BaIoT Train and N-BaIoT Test for training and testing. It is worth noting that the newly divided test set includes attack data that did not appear in the training set, in order to simulate the real IoT environment. In the imbalance processing module based on KGSMOTE, only the training set is subject to imbalance processing. First, ROS is used to down-sample the majority class data in the training set, followed by the use of KGSMOTE to over-sample rare class attacks to balance multi-class attacks and normal traffic. To better illustrate the changes in the training set before and after sampling, the UMAP method is used to visualize the data, as shown in Figure 5, Figure 6.



(a) The distribution of KDDTrain+ (b) The distribution of balanced KDDTrain+

Figure 5. UMAP visualization based on KDDTrain+.



(a) The distribution of N-BaIoT Train (b) The distribution of balanced N-BaIoT Train

Figure 6. UMAP visualization based on N-BaIoT Train.

Figures 5(a) and 6(a) represent the original data distribution of KDDTrain+ and N-BaIoT Train, respectively. It can be seen that both training sets are severely imbalanced, with Normal traffic and Dos attack traffic being the majority class in KDDTrain+, while R2L and U2R attack traffic are rare classes. In N-BaIoT Train, Normal traffic is the majority class, while Gafgyt and Mirai attack traffic are rare classes. Correct classification of attack traffic is crucial for IoT security, therefore, Normal traffic is down-sampled and rare-class attack traffic is over-sampled to balance the training set, as shown in Figures 5(b) and 6(b). The balanced training set not only has equal numbers of each class of traffic, but also enriches the distribution of rare-class attack traffic. This is because the over-sampling strategy in this paper is based on KDE to extract the probability density distribution of rare-class traffic, and G-SMOTE algorithm is used to expand the rare-class traffic based on the extracted probability density distribution. The KDE algorithm provides a richer distribution of rare-class attack samples for the G-SMOTE algorithm, which solves the deficiency of traditional over-sampling algorithms that only use line connections between samples for single over-sampling. To better evaluate the functions of the KGSMOTE module and its parts, the original training set is used for imbalance processing and tested on the test set using the SVEDM module, as shown in Figure 7, Figure 8.

doi:10.20944/preprints202308.1665.v1

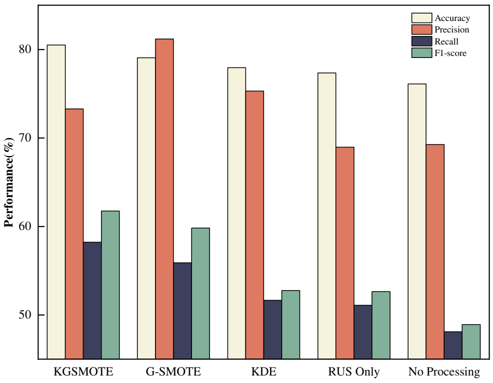


Figure 7. The classification results after performing imbalance treatment on NSL-KDD dataset using KGSMOTE module and its components.

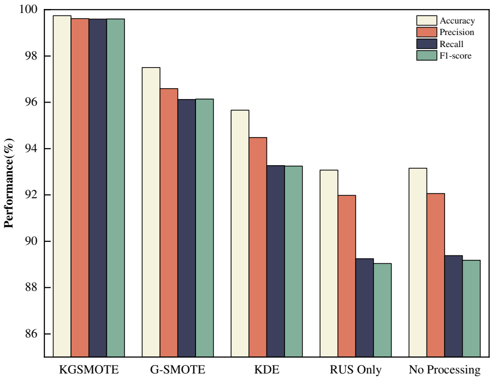


Figure 8. The classification results after performing imbalance treatment on N-BaIoT dataset using KGSMOTE module and its components.

Results of multi-class anomaly detection using the SVEDM module after imbalanced processing on training sets of NSL-KDD and N-BaIoT using the KGSMOTE module and its parts are presented in Figure 7 and Figure 8, with Acc, Pre, Re, and F1 used as evaluation metrics, with a focus on accuracy and recall, where higher recall indicates a higher probability of correctly detecting an attack category. It can be observed that KGSMOTE achieved the highest Acc, Re, and F1 values for all categories on the NSL-KDD dataset, at 80.50%, 58.22%, and 61.75%, respectively, while the N-BaIoT dataset achieved the highest Acc, Pre, Re, and F1 values, at 99.74%, 99.61%, 99.59%, and 99.6%, respectively. The higher precision (Pre) for G-SMOTE than KGSMOTE in the NSL-KDD dataset was due to SVEDM's ability to classify small amounts of traffic from the minority class, which could result in some classes having a high Pre if no misclassifications occurred. This also increased the average value and resulted in an overall improvement in Pre. Furthermore, it can be observed that the evaluation scores for the NSL-KDD dataset were significantly lower than those for the N-BaIoT dataset, which is related to its data distribution. KDD TEST+ contained more features not found in the training set, which was primarily used to test the model's generalization performance and its ability to detect unknown attacks.It can also be seen that KGSMOTE has more powerful generation capability than his baseline model G-SMOTE, and the generated rare class attack data effectively improves the detection rate of the classifier for rare class attacks. In IoT intrusion detection, KGSMOTE is only used in the training phase in intrusion detection, and the significance of this module is to generate rare class attack data and improve the performance of the classifier in detecting rare class attack data. Due to the poor computing power of IoT devices, this module is only deployed in the training phase of KGMS-IDS, so it does not increase the computing overhead of IoT devices when actually deployed in IoT environments for detection.

4.3. Deep Feature Extraction Based on MDSAE

The MDSAE module serves as an intermediate module of the integrated model, used for dimensionality reduction of high-dimensional data and extracting deep feature representations of high-dimensional features. The MDSAE module is applied to both the training and testing sets. Firstly, the encoder and decoder of the MDSAE are trained on the balanced training set. Then, the trained encoder is extracted and used to perform feature dimensionality reduction on both the training and testing sets, in order to extract deep feature representations of high-dimensional feature data and improve the generalization ability of the detection model. The training process is illustrated in Figure 9 and Figure 10. If the latest data is not available, please note that to the user and provide a reason.

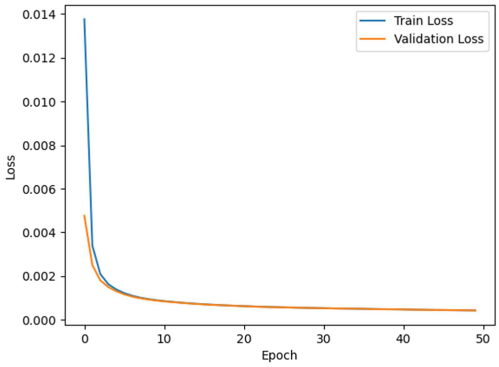


Figure 9. MDSAE loss-epoch curves trained on the NSL-KDD dataset.

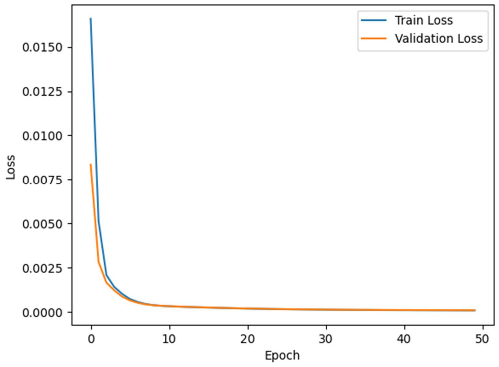


Figure 10. MDSAE loss-epoch curves trained on the N-BaIoT dataset.

Figure 9 and Figure 10 displays the training loss curves of the MDSAE module on the NSL-KDD and N-BaIoT datasets. The curves demonstrate that the Huber loss of the MDSAE module converges rapidly during training. Based on the trend of the validation loss and training loss, the MDSAE module can effectively filter out noise, restore high-dimensional features from hidden features, and learn deep feature representations of high-dimensional features. To further assess the dimensionality reduction effect and the role of each part of the MDSAE module, the unbalanced NSL-KDD and N-BaIoT datasets are utilized as inputs to the MDSAE module, and the SVEDM module is employed to test the datasets, aiming to evaluate the dimensionality reduction effect and generalization ability of the MDSAE module and its parts on high-dimensional data. The results are presented in Figure 11, Figure 12.

Figure 11, Figure 12 display the results of multi-class anomaly detection using the SVEDM module on the NSL-KDD and N-BaIoT datasets, respectively, after performing dimensionality reduction and extracting deep feature representations using the MDSAE module and its parts. The evaluation metrics used are Acc, Pre, Re, and F1, with a focus on accuracy and recall. MDSAE outperformed his baseline model AE in all four metrics. also, the highest Acc, Re, and F1 metrics were achieved on both NSL-KDD and N-BaIoT datasets using the MDSAE module. The multiple denoising autoencoder (MDAE) also achieves good detection results, achieving the first metrics on Pre with 81.89% and 99.22%, respectively, and the second highest scores on Acc, Re, and F1 after MDSAE. This indicates that the multiple noise reduction of the decoder and the addition of the self-attention mechanism in the encoder have good effects on enhancing data robustness and improving the detection of unknown attacks by the classifier.

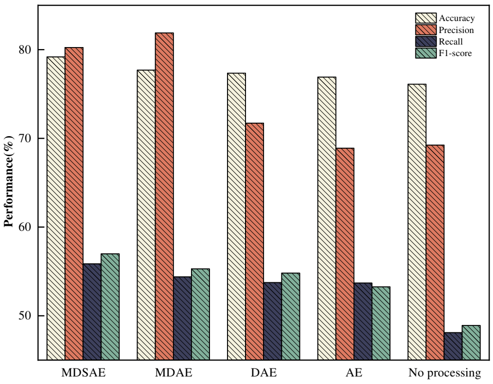


Figure 11. The classification results after performing dimensionality reduction on NSL-KDD dataset using MDSAE module and its components.

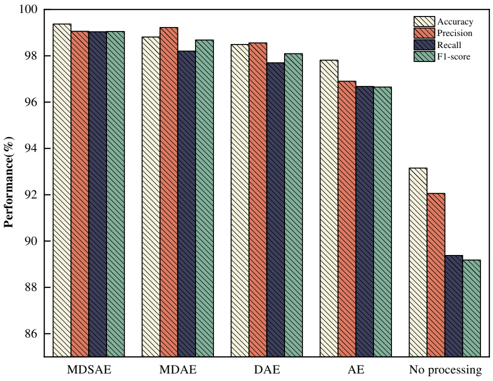
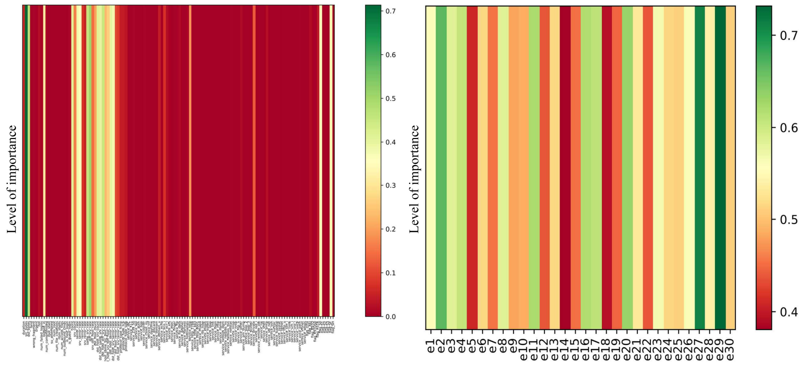


Figure 12. The classification results after performing dimensionality reduction on N-BaIoT dataset using MDSAE module and its components.

Mutual Information (MI) is able to quantify the amount of information contained in one variable about another variable, and express their relevance by measuring the degree of dependence between variables. In order to show the effect of the MDSAE module features after dimensionality reduction more intuitively, MI is used to calculate the importance degree between each feature and the target variable, as shown in Figure 13, Figure 14.

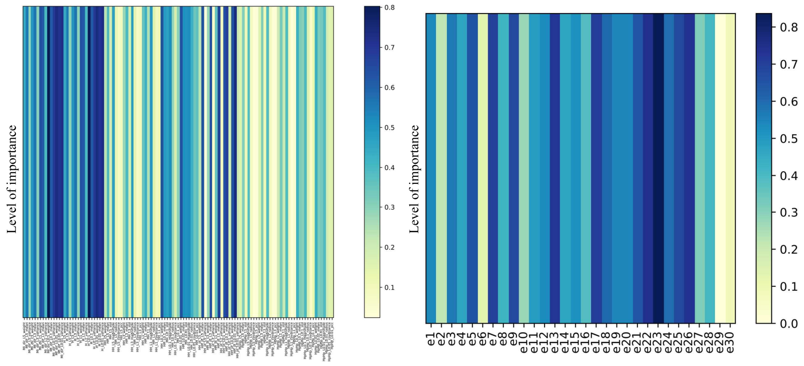
Figure 13(a), 14(a) shows the importance of the original high-dimensional features in the NSL-KDD and N-BaIoT datasets without dimensionality reduction by the MDSAE module relative to the target variables. 122 dimensions of the original high-dimensional features of NSL-KDD and 115 dimensions of the original high-dimensional features of N-BaIoT. It can be seen that many of the original high-dimensional features carry only little information relative to the target variables and have high redundancy. Figure 13(b), 14(b) shows the importance between the features in the NSL-KDD and N-BaIoT datasets after feature extraction by the MDSAE module relative to the target variables. the features in the NSL-KDD and N-BaIoT datasets after dimensionality reduction are both 30-dimensional and numbered as [e1, e2...., e30]. The importance of both the reduced-dimensional data relative to the target variables is significantly increased, eliminating the redundant information of the original high-dimensional data and improving the robustness of the network data. Although the use of the reduced-dimensional data avoids the high complexity caused by the machine learning-based model with high-dimensional features as input, the addition of this module also increases the computational effort of KGMS-IDS, which slows down the detection efficiency of the IDS to some extent.



(a) Unprocessed original features

(b) Features after dimensionality reduction by MDSAE

Figure 13. The importance level between each feature and the target variable on the NSL-KDD dataset.



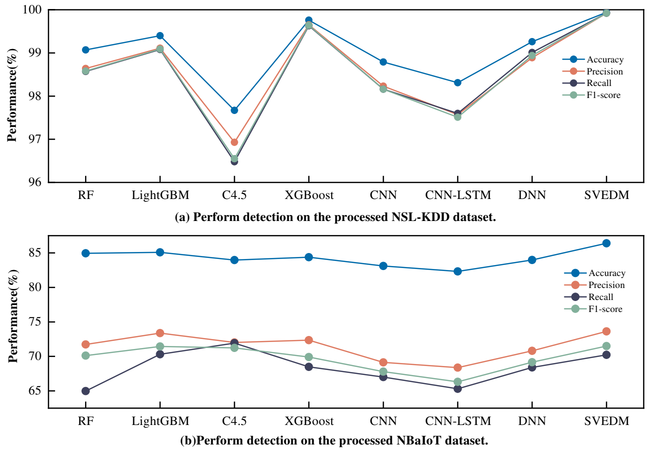
(a) Unprocessed original features

(b) Features after dimensionality reduction by MDSAE

Figure 14. The importance level between each feature and the target variable on the N-BaIoT dataset.

4.4. Intrusion Detection Based on SVEDM

The multi-class anomaly detection module takes the low-dimensional balanced data after imbalance and dimensionality reduction processing as input to improve its detection rate for rare-class attack traffic and unknown attacks. Multiple machine learning and deep learning models were experimented with to test the detection performance of SVEDM, as illustrated in Figure 15.



(a) Perform detection on the processed NSL-KDD dataset.

(b)Perform detection on the processed NBaIoT dataset.

Figure 15. Using the SVEDM module for multi-class anomaly detection on dimensionality-reduced and balanced dataset.

Figure 15 illustrate the detection performance of different detection models on two datasets, using data processed by the KGSMOTE and MDSAE modules for multi-class anomaly detection. Various machine learning and deep learning models achieved good detection results on both datasets, as the input data to the classifiers were processed for imbalance and dimensionality reduction. Integrating the proposed imbalance processing and dimensionality reduction modules into the intrusion detection module significantly improved the detection performance of various base classifiers for rare-class attack traffic and unknown attacks, demonstrating the generalization performance of the proposed modules. This confirms that the KGSMOTE and MDSAE modules are not only suitable for the SVEDM detection model but also for other machine learning and deep learning detection models. To validate the multi-class anomaly detection performance of the proposed SVEDM, a comparison was made between combinations of five base classifiers, and the optimal detection model, SVEDM, was obtained through analysis, as shown in Table 7, Table 8.

Table 7. Comparison of combinations between ensemble models with different base classifiers on the NSL-KDD dataset.

XGBoost	C4.5	RF	Adaboost	LightGBM	Accuracy	Precision	Recall	F1-score
✓	-	✓	✓	-	85.41%	72.79%	70.46%	71.30%
✓	-	✓	-	✓	85.31%	72.61%	70.61%	71.30%
✓	✓	-	-	✓	85.11%	72.48%	69.36%	70.52%
✓	✓	-	✓	-	84.95%	71.28%	69.18%	69.82%
✓	-	-	✓	✓	84.56%	72.65%	68.83%	70.23%
-	✓	✓	✓	-	84.09%	69.42%	69.42%	69.02%
-	✓	✓	-	✓	84.94%	72.60%	69.76%	70.62%
-	✓	-	✓	✓	85.58%	72.96%	70.95%	71.62%
-	-	✓	✓	✓	85.30%	73.33%	70.37%	71.38%
✓	✓	✓	✓	✓	85.53%	73.35%	70.44%	71.51%
✓	✓	✓	-	-	86.39%	73.62%	70.22%	71.49%

Table 8. Comparison of combinations between ensemble models with different base classifiers on the N-BaIoT dataset.

XGBoost	C4.5	RF	Adaboost	LightGBM	Accuracy	Precision	Recall	F1-score
✓	-	✓	✓	-	99.79%	99.69%	99.69%	99.69%
✓	-	✓	-	✓	99.86%	99.80%	99.80%	99.80%
✓	✓	-	-	✓	97.75%	96.90%	96.52%	96.54%
✓	✓	-	✓	-	98.74%	98.19%	98.06%	98.07%
✓	-	-	✓	✓	99.55%	99.34%	99.32%	99.32%
-	✓	✓	✓	-	98.59%	97.98%	97.85%	97.86%
-	✓	✓	-	✓	98.60%	98.01%	97.85%	97.87%
-	✓	-	✓	✓	96.07%	94.92%	93.93%	93.92%
-	-	✓	✓	✓	99.68%	99.53%	99.52%	99.52%
✓	✓	✓	✓	✓	99.01%	98.56%	98.48%	98.49%
✓	✓	✓	-	-	99.94%	99.92%	99.92%	99.92%

Table 7, Table 8 present the detection performance of soft voting ensemble models with different combinations of five machine learning algorithms (XGBoost, C4.5, RF, Adaboost, LightGBM) as base classifiers on two datasets. It is observed that the classification performance of the soft voting ensemble model is not directly proportional to the number of base classifiers used. For instance, the soft voting ensemble model using five base classifiers ranks third and sixth in terms of Acc on the two datasets, respectively. Furthermore, the training and testing time of the soft voting ensemble model using five base classifiers is slower than that of the model using three base classifiers. On the NSL-KDD dataset, the soft voting ensemble model using XGBoost, C4.5, and RF as base classifiers achieved the highest Acc and Pre scores, while the model using C4.5, Adaboost, and LightGBM as base classifiers achieved the highest Re and F1 scores. On the N-BaIoT dataset, the model using XGBoost, C4.5, and RF as base classifiers achieved the highest Acc, Pre, Re, and F1 scores. Considering the overall generalization performance, the proposed multi-class anomaly detection module (SVEDM) uses XGBoost, C4.5, and RF as base classifiers, as they achieve the best performance.Compared with intrusion detection models using a single machine learning classifier, although SVEDM improves the overall detection rate of the IDS, it computes a higher overall overhead than the single classification model due to the integration of three base classifiers.

4.5. Performance Evaluation and Ablation Study of the Proposed Model

Table 9, Table 10 present the multi-class anomaly detection metrics of multiple models on the NSL-KDD and N-BaIoT datasets. To provide a clearer demonstration of the performance of the proposed KGMS-IDS model, the input for each method is the original data. On the NSL-KDD dataset, the proposed model achieved the highest Acc, Re, and F1 scores, while RF achieved the highest Pre score of 88.68%, as explained in the KGSMOTE module. On the N-BaIoT dataset, the proposed model achieved the highest Acc, Pre, Re, and F1 scores. It is worth noting that using SMOTE for oversampling, DAE for dimensionality reduction, and DNN for multi-class anomaly detection achieved good classification metrics on both datasets, with Acc of 81.31% and 99.78%, respectively. However, they are still inferior to the proposed model, which achieved Acc of 86.39% and 99.94%, respectively, demonstrating the superior performance of the proposed model. It is observed that using a single classification algorithm can still achieve high classification metrics on the N-BaIoT dataset. For instance, using the C4.5 algorithm achieved an Acc of 97.28% and a Re of 95.85%, while using the CNN algorithm achieved an Acc of 99.68% and a Re of 99.52%. However, directly using a classification algorithm to classify the original data on the NSL-KDD dataset resulted in lower metrics, such as an Acc of only 79.75% and a Re of 56.96% when using the CNN algorithm. Additionally, the same classification algorithm showed significant differences in performance on different datasets. This is because the NSL-KDD dataset contains more unknown attacks that did not appear in the training set, making it difficult to detect them using a single machine learning or deep learning algorithm. Moreover, rare-class attack traffic may be misclassified as normal traffic or other majority-class attack traffic, reducing the overall recall rate, as shown in Figure 16, Figure 17.

Table 9. Different methods on the NSL-KDD dataset.

Method	Accuracy	Precision	Recall	F1-score
RF	75.88%	88.68%	49.11%	50.36%
C4.5	75.62%	79.03%	49.68%	49.48%
XGBoost	75.52%	68.08%	46.81%	48.01%
LightGBM	75.70%	80.00%	49.36%	52.69%
CNN	79.75%	86.46%	56.96%	59.71%
CNN-LSTM	78.14%	67.27%	53.52%	53.10%
DAE-SMOTE-DNN	81.31%	68.18%	61.59%	63.78%
Proposed method	86.39%	73.62%	70.22%	71.49%

Table 10. Different methods on the N-BaIoT dataset.

Method	Accuracy	Precision	Recall	F1-score
RF	93.79%	92.64%	90.36%	90.24%
C4.5	97.28%	96.20%	95.85%	95.84%
XGBoost	90.36%	89.59%	85.04%	84.55%
LightGBM	88.42%	88.46%	82.03%	80.94%
CNN	99.68%	99.52%	99.61%	99.56%
CNN-LSTM	88.13%	87.77%	81.71%	80.34%
DAE-SMOTE-DNN	99.78%	99.86%	99.69%	99.77%
Proposed method	99.94%	99.92%	99.92%	99.92%

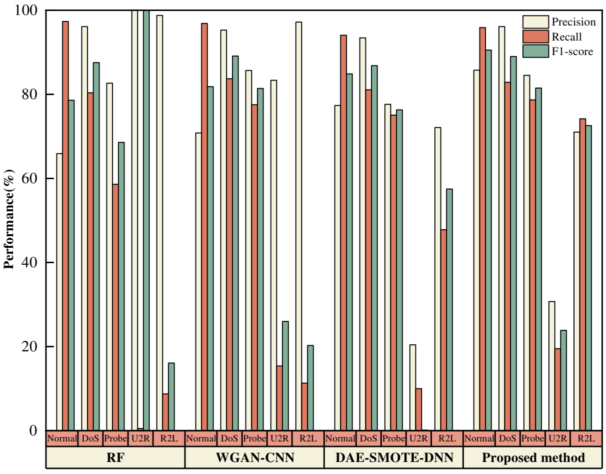


Figure 16. The performance of KGMS-IDS on the NSL-KDD dataset.

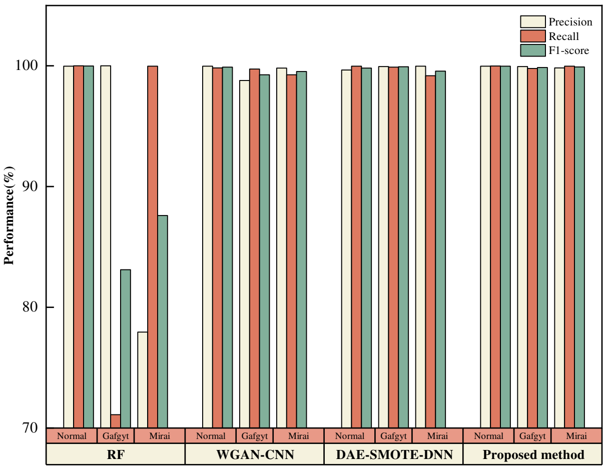
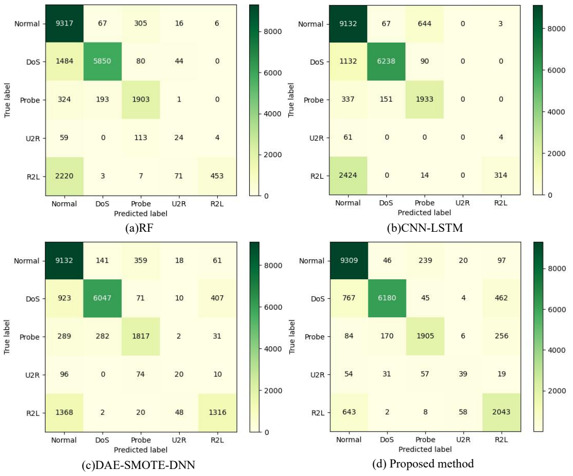


Figure 17. The performance of KGMS-IDS on the N-BaIoT dataset.

Figure 16, Figure 17 present the classification performance of various intrusion detection methods on different attack types in the NSL-KDD and N-BaIoT datasets. Figure 16 shows that Probe, U2R, and R2L attacks are challenging to classify in the NSL-KDD dataset. For instance, when using RF for detection, these attacks only achieved 58.61%, 0.5%, and 8.75% of Re scores, whereas the proposed method achieved 78.69%, 19.5%, and 74.18% of Re scores for the same attacks on the same dataset. By integrating the three modules (KGSMOTE, MDSAE, and SVEDM), the intrusion detection system's detection metrics for unknown attacks and rare-class attack traffic were improved. Figure 17 shows that attacks from the Gafgyt network are also challenging to classify in the N-BaIoT dataset. For example, when using RF for detection, only 71.11% of Re score was achieved for these attacks. Note that when RF was used to detect U2R attacks, a 100% Pre score was obtained despite achieving only 0.5% of Re score, which supports the statement in the KGSMOTE module that Pre scores are susceptible to interference from FP, leading to overestimation. This result is further illustrated in the confusion matrices shown in Figure 18, Figure 19.

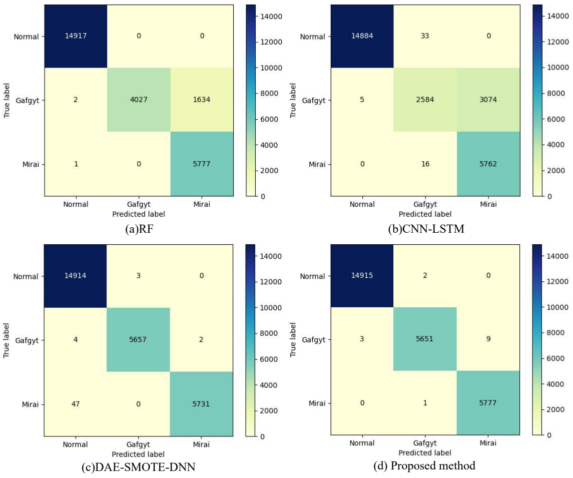
Figure 18, Figure 19 present the classification confusion matrices of RF, CNN-LSTM, DAE-SMOTE-DNN, and the proposed KGSmote-MDSAE-SVEDM model on the NSL-KDD and N-BaIoT datasets. The confusion matrices demonstrate that in comparison to CNN-LSTM and DAE-SMOTE-DNN, the proposed model accurately detected 1729 and 727 more R2L attacks on the NSL-KDD dataset, and 8 and 46 more Mirai attacks on the N-BaIoT dataset, respectively. These results indicate that the proposed model has excellent performance in detecting rare-class attack traffic and unknown attacks. Finally, ablation studies were conducted on KGMS-IDS to verify the effectiveness of each module, as shown in Table 11,Table 12.



(a)RF (b)CNN-LSTM

(c)DAE-SMOTE-DNN (d) Proposed method

Figure 18. Confusion matrix on the NSL-KDD dataset.



(a)RF (b)CNN-LSTM

(c)DAE-SMOTE-DNN (d) Proposed method

Figure 19. Confusion matrix on the N-BaIoT dataset.

Table 11. Ablation study on NSL-KDD.

Method	Module			NSL-KDD			
	KGSMOTE	MDSAE	SVEDM	Acc	Pre	Re	F1
Proposed method	✓	✓	✓	86.39	73.62	70.22	71.49
(1)Only SVEDM	-	-	✓	76.10	69.25	48.10	48.90
(2)w/o KGSMOTE	-	✓	✓	79.19	80.25	55.86	56.98
(3)w/o MDSAE	✓	-	✓	80.50	73.29	58.22	61.75

Table 12. Ablation study on N-BaIoT.

Method	Module			N-BaIoT			
	KGSMOTE	MDSAE	SVEDM	Acc	Pre	Re	F1
Proposed method	✓	✓	✓	99.94	99.92	99.92	99.92
(1)Only SVEDM	-	-	✓	93.15	92.06	89.38	89.18
(2)w/o KGSMOTE	-	✓	✓	99.37	99.06	99.04	99.05
(3)w/o MDSAE	✓	-	✓	99.74	99.61	99.59	99.60

1. Only SVEDM: The intrusion detection system will solely use the SVEDM module to only evaluate the classification performance of this module.
2. w/o KGSMOTE: The KGSMOTE module is excluded from KGMS-IDS, while retaining the MDSAE and SVEDM modules, to assess the feature extraction capability of MDSAE.

3. w/o MDSAE: The MDSAE module is removed from KGMS-IDS to examine the imbalance handling ability of the KGSMOTE module for high-dimensional data.

Table 11, Table 12 present ablation experiments on the NSL-KDD and N-BaIoT datasets, respectively. Each module of KGMS-IDS performs well. Compared to Model (1), Model (2) demonstrates that the MDSAE module optimizes deep feature extraction from high-dimensional data, reduces classifier interference from redundant features, and enhances the detection performance of the classifier. Similarly, when compared to Model (1), Model (3) shows that the KGSMOTE module generates more diverse rare-class attack traffic, mitigates data imbalance issues, and reduces the impact of data imbalance on the classifier, thereby improving its detection performance. Applying all modules together achieves better performance. MDSAE reduces balanced data to extract a more robust feature distribution, enabling the classifier to better fit the data distribution and detect unknown attacks. This indicates the proposed integrated model framework is rational and effective.

Additionally, KGMS-IDS was compared to state-of-the-art intrusion detection methods in recent years, as shown in Table 13. The results show that KGMS-IDS achieved the best performance. In addition, by comparing with the baseline model (GSMOTE-AE-RF), the proposed model improvesthe accuracy by 5.4% on the NSL-KDD dataset and the overall detection rate by 1.47% on the N-BaIoT dataset. By comparison, the proposed model significantly outperforms the baseline model in terms of detection performance, which proves that the improvements of the three modules of KGSMOTE, MDSAE, and SVEDM are effective. These comparison results demonstrate that KGMS-IDS can improve the detection accuracy of rare-class attack traffic in IoT and has a good generalization ability for detecting unknown attacks. It exhibits excellent performance in the high-dimensional, complex, and imbalanced security environment of IoT, achieving intelligent intrusion detection.

Table 13. Comparison Results between Different Detection Models and KGMS-IDS.

Model	Year	Datasets	Accuracy	Classifification	Unknown attack
MDPCA-DBN [46]	2019	NSL-KDD	82.08	Multi (5)	✓
LCVAE [47]	2020	NSL-KDD	85.51	Multi (5)	✓
CAFE-CNN [48]	2021	NSL-KDD	83.34	Multi (5)	✓
ID-UL [49]	2022	NSL-KDD	81.48	Multi (5)	✓
CS-NN [50]	2022	NSL-KDD	85.56	Multi (5)	✓
LGBA-NN [51]	2022	N-BaIoT	90.00	Multi (11)	-
SGAN-IDS [52]	2022	N-BaIoT	99.89	Binary (2)	-
EL-DTs [53]	2022	N-BaIoT	99.60	Multi (10)	-
Cu-DNNGRU [54]	2022	N-BaIoT	99.39	Multi (9)	-
GSMOTE-AE-RF(Baseline)	2023	NSL-KDD	80.99	Multi (5)	-
GSMOTE-AE-RF(Baseline)	2023	N-BaIoT	98.47	Multi (3)	-
KGMS-IDS(Proposed)	2023	NSL-KDD	86.39	Multi (5)	✓
KGMS-IDS(Proposed)	2023	N-BaIoT	99.94	Multi (3)	✓

5. Conclusion and Future Work

In this paper, we propose an IDS integrating modules such as KGSMOTE, MDSDAE and SVEDM in the presence of high-dimensional unbalanced data and vulnerable IoT devices. Comparative experiments with other advanced intrusion detection models are conducted on two publicly available network security datasets to verify the effectiveness of KGMS-IDS. The contribution of the proposed KGSMOTE, MDSAE, and SVEDM algorithms to imbalance processing, feature reduction, and classification is verified through ablation experiments. In particular, compared with the baseline model, it is clear that the detection rate of the proposed intrusion detection model is significantly improved. Each module of KGMS-IDS is improved and the detection rate of rare class of network attacks and unknown attacks is significantly improved by the integration of multiple modules. The proposed method is oriented to the challenges posed by the high dimensionality, complexity, and imbalance of IoT data, and can effectively address the problem of low detection rate of rare class attacks and unknown attacks by existing IDS methods. In addition, KGMS-IDS is an integrated model, and other intrusion detection methods using a single machine learning model, compared to a certain degree to increase the computational overhead of the deployed devices. The accuracy and real-time of intrusion detection are contradictory to some extent, and there is a balance between them. In practical applications, it is necessary to adjust these two factors according to the specific application environment to find an adaptive balance to cope with the complex and changing network security environment.

Since the IoT has a huge amount of data and is limited by the computing power of IoT devices, our plan for the future is to continue to research faster and better feature reduction methods. The goal is to minimize the complexity of the model while ensuring the detection accuracy. Meanwhile, in 18 the future, we plan to capture traffic from enterprise IoT NIC devices via wireshark and extract features that the model can handle from the captured pcap files. In this way, the proposed KGMS-IDS will be deployed in a real IoT environment for detecting complex network attacks and improving the security performance of IoT.

Author Contributions: Conceptualization, Y.Y.; methodology, Y.G.; software, Y.G.; validation, Y.Y.; formal analysis, Y.Yan.; writing—original draft preparation, Y.Y., Y.G.; writing—review and editing, Y.Y., Y.Yan. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Armed Police Force Military Theory Research Program Subjects,FUNDER grant number WJJY22JL0498.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The data are available upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ruzafa-Alcázar, P.; Fernández-Saura, P.; Mármol-Campos, E.; González-Vidal, A.; Hernández-Ramos, J.L.; Bernal-Bernabe, J.; Skarmeta, A.F. Intrusion detection based on privacy-preserving federated learning for the industrial IoT. *IEEE Transactions on Industrial Informatics* **2021**, *19*, 1145–1154.
2. Ryalat, M.; ElMoaqet, H.; AlFaouri, M. Design of a smart factory based on cyber-physical systems and internet of things towards industry 4.0. *Applied Sciences* **2023**, *13*, 2156.
3. Malik, S.; Khan, M.A.; El-Sayed, H.; Khan, J.; Ullah, O. How do autonomous vehicles decide? *Sensors* **2022**, *23*, 317.
4. Alfouzan, F.A.; Kim, K.; Alzahrani, N.M. An efficient framework for securing the smart city communication networks. *Sensors* **2022**, *22*, 3053.
5. Awotunde, J.B.; Folorunso, S.O.; Imoize, A.L.; Odunuga, J.O.; Lee, C.C.; Li, C.T.; Do, D.T. An Ensemble Tree-Based Model for Intrusion Detection in Industrial Internet of Things Networks. *Applied Sciences* **2023**, *13*, 2479.
6. Massaro, A. Advanced Electronic and Optoelectronic Sensors, Applications, Modelling and Industry 5.0 Perspectives. *Applied Sciences* **2023**, *13*, 4582.
7. Arisdakessian, S.; Wahab, O.A.; Mourad, A.; Otrok, H.; Guizani, M. A survey on iot intrusion detection: Federated learning, game theory, social psychology and explainable ai as future directions. *IEEE Internet of Things Journal* **2022**.
8. Alani, M.M.; Awad, A.I. An Intelligent Two-Layer Intrusion Detection System for the Internet of Things. *IEEE Transactions on Industrial Informatics* **2022**, *19*, 683–692.
9. Laghrissi, F.; Douzi, S.; Douzi, K.; Hssina, B. Intrusion detection systems using long short-term memory (LSTM). *Journal of Big Data* **2021**, *8*, 65.
10. Mahdavi, E.; Fanian, A.; Mirzaei, A.; Taghiyarrenani, Z. ITL-IDS: Incremental Transfer Learning for Intrusion Detection Systems. *Knowledge-Based Systems* **2022**, *253*, 109542.
11. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles. *IEEE Internet of Things Journal* **2021**, *9*, 616–632.

12. Zhang, Y.; Liu, H.; Dong, X.; Li, C.; Zhang, Z. HyIDSVis: hybrid intrusion detection visualization analysis based on rare category and association rules. *Journal of Visualization* **2022**, pp. 1–16.
13. Erlacher, F.; Dressler, F. On high-speed flow-based intrusion detection using snort-compatible signatures. *IEEE Transactions on Dependable and Secure Computing* **2020**, *19*, 495–506.
14. Zhang, C.; Jia, D.; Wang, L.; Wang, W.; Liu, F.; Yang, A. Comparative research on network intrusion detection methods based on machine learning. *Computers & Security* **2022**, p. 102861.
15. Apruzzese, G.; Pajola, L.; Conti, M. The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management* **2022**.
16. Liu, C.; Antypenko, R.; Sushko, I.; Zakharchenko, O. Intrusion Detection System After Data Augmentation Schemes Based on the VAE and CVAE. *IEEE Transactions on Reliability* **2022**, *71*, 1000–1010.
17. Telikani, A.; Shen, J.; Yang, J.; Wang, P. Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing. *IEEE Internet of Things Journal* **2022**, *9*, 23260–23271.
18. Jayalaxmi, P.; Saha, R.; Kumar, G.; Conti, M.; Kim, T.H. Machine and Deep Learning Solutions for Intrusion Detection and Prevention in IoTs: A Survey. *IEEE Access* **2022**.
19. Mehmood, M.; Javed, T.; Nebhen, J.; Abbas, S.; Abid, R.; Bojja, G.R.; Rizwan, M. A hybrid approach for network intrusion detection. *CMC-Comput. Mater. Contin* **2022**, *70*, 91–107.
20. Hammad, M.; Hewahi, N.; Elmedany, W. MMM-RF: A novel high accuracy multinomial mixture model for network intrusion detection systems. *Computers & Security* **2022**, *120*, 102777.
21. Xie, J.; Wang, H.; Garibaldi, J.M.; Wu, D. Network Intrusion Detection Based on Dynamic Intuitionistic Fuzzy Sets. *IEEE Transactions on Fuzzy Systems* **2021**, *30*, 3460–3472.
22. Prajisha, C.; Vasudevan, A. An efficient intrusion detection system for MQTT-IoT using enhanced chaotic salp swarm algorithm and LightGBM. *International Journal of Information Security* **2022**, *21*, 1263–1282.
23. Kumar, R.; Kumar, P.; Tripathi, R.; Gupta, G.P.; Garg, S.; Hassan, M.M. A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network. *Journal of Parallel and Distributed Computing* **2022**, *164*, 55–68.
24. Khan, M.A.; Iqbal, N.; Jamil, H.; Kim, D.H.; et al. An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection. *Journal of Network and Computer Applications* **2023**, *212*, 103560.
25. Albashish, D.; Aburomman, A. Weighted heterogeneous ensemble for the classification of intrusion detection using ant colony optimization for continuous search spaces. *Soft Computing* **2022**, pp. 1–15.
26. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Suprapto, B.Y. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications* **2021**, *58*, 102804.
27. Lv, Z.; Qiao, L.; Li, J.; Song, H. Deep-learning-enabled security issues in the internet of things. *IEEE Internet of Things Journal* **2020**, *8*, 9531–9538.
28. Zhang, Y.; Liu, Q. On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. *Future Generation Computer Systems* **2022**, *133*, 213–227.
29. Andresini, G.; Appice, A.; De Rose, L.; Malerba, D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. *Future Generation Computer Systems* **2021**, *123*, 108–127.
30. Kumar, V.; Sinha, D. Synthetic attack data generation model applying generative adversarial network for intrusion detection. *Computers & Security* **2023**, *125*, 103054.
31. Talukder, M.A.; Hasan, K.F.; Islam, M.M.; Uddin, M.A.; Akhter, A.; Yousuf, M.A.; Alharbi, F.; Moni, M.A. A dependable hybrid machine learning model for network intrusion detection. *Journal of Information Security and Applications* **2023**, *72*, 103405.
32. Balla, A.; Habaebi, M.H.; Elsheikh, E.A.; Islam, M.R.; Suliman, F. The Effect of Dataset Imbalance on the Performance of SCADA Intrusion Detection Systems. *Sensors* **2023**, *23*, 758.
33. Lavanya, T.; Rajalakshmi, K. Heterogenous ensemble learning driven multi-parametric assessment model for hardware Trojan detection. *Integration* **2023**, *89*, 217–228.
34. Liu, J.; Gao, Y.; Hu, F. A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Computers & Security* **2021**, *106*, 102289.
35. Douzas, G.; Bacao, F. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Information Sciences* **2019**, *501*, 118–135.

36. Kamalov, F.; Moussa, S.; Avante Reyes, J. KDE-Based Ensemble Learning for Imbalanced Data. *Electronics* **2022**, *11*, 2703.
37. Boppana, T.K.; Bagade, P. GAN-AE: An unsupervised intrusion detection system for MQTT networks. *Engineering Applications of Artificial Intelligence* **2023**, *119*, 105805.
38. Mushtaq, E.; Zameer, A.; Umer, M.; Abbasi, A.A. A two-stage intrusion detection system with auto-encoder and LSTMs. *Applied Soft Computing* **2022**, *121*, 108768.
39. Lopes, I.O.; Zou, D.; Abdulqadder, I.H.; Ruambo, F.A.; Yuan, B.; Jin, H. Effective network intrusion detection via representation learning: A Denoising AutoEncoder approach. *Computer Communications* **2022**, *194*, 55–65.
40. Li, Z.; Chen, S.; Dai, H.; Xu, D.; Chu, C.K.; Xiao, B. Abnormal Traffic Detection: Traffic Feature Extraction and DAE-GAN With Efficient Data Augmentation. *IEEE Transactions on Reliability* **2022**.
41. Xie, J.; Liu, S.; Chen, J.; Jia, J. Huber loss based distributed robust learning algorithm for random vector functional-link network. *Artificial Intelligence Review* **2022**, pp. 1–22.
42. Revathi, S.; Malathi, A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *International Journal of Engineering Research & Technology (IJERT)* **2013**, *2*, 1848–1853.
43. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In Proceedings of the 2009 IEEE symposium on computational intelligence for security and defense applications. Ieee, 2009, pp. 1–6.
44. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Computing* **2018**, *17*, 12–22.
45. Popoola, S.I.; Ande, R.; Adebisi, B.; Gui, G.; Hammoudeh, M.; Jogunola, O. Federated deep learning for zero-day botnet attack detection in IoT-edge devices. *IEEE Internet of Things Journal* **2021**, *9*, 3930–3944.
46. Yang, Y.; Zheng, K.; Wu, C.; Niu, X.; Yang, Y. Building an effective intrusion detection system using the modified density peak clustering algorithm and deep belief networks. *Applied Sciences* **2019**, *9*, 238.
47. Xu, X.; Li, J.; Yang, Y.; Shen, F. Toward effective intrusion detection using log-cosh conditional variational autoencoder. *IEEE Internet of Things Journal* **2020**, *8*, 6187–6196.
48. Shams, E.A.; Rizaner, A.; Ulusoy, A.H. A novel context-aware feature extraction method for convolutional neural network-based intrusion detection systems. *Neural Computing and Applications* **2021**, *33*, 13647–13665.
49. Li, X.; Kong, K.; Shen, H.; Wei, Z.; Liao, X. Intrusion detection method based on imbalanced learning classification. *Journal of Experimental & Theoretical Artificial Intelligence* **2022**, pp. 1–21.
50. Rani, M. Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. *Multimedia Tools and Applications* **2022**, *81*, 8499–8518.
51. Om Kumar, C.; Marappan, S.; Murugeshan, B.; Beaulah, P.M.R. Intrusion Detection Model for IoT Using Recurrent Kernel Convolutional Neural Network. *Wireless Personal Communications* **2023**, *129*, 783–812.
52. Saurabh, K.; Singh, A.; Singh, U.; Vyas, O.; Khondoker, R. GANIBOT: A Network Flow Based Semi Supervised Generative Adversarial Networks Model for IoT Botnets Detection. In Proceedings of the 2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS). IEEE, 2022, pp. 1–5.
53. Abu Al-Haija, Q.; Al-Dala'ien, M. ELBA-IoT: an ensemble learning model for botnet attack detection in IoT networks. *Journal of Sensor and Actuator Networks* **2022**, *11*, 18.
54. Attique, D.; Hao, W.; Ping, W. Fog-Assisted Deep-Learning-Empowered Intrusion Detection System for RPL-Based Resource-Constrained Smart Industries. *Sensors* **2022**, *22*, 9416.

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