electronics

MDPI

Article

Machine Learning-Based Intrusion Detection for Rare-Class Network Attacks

Yu Yang, Yuheng Gu * and Yu Yan

College of Information Engineering, Chinese People's Armed Police Force Engineering University, Xi'an 710086, China; yuxsky91039@163.com (Y.Y.); a13785062908@163.com (Y.Y.)
* Correspondence: a15734042989@163.com

Abstract: 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 the 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. The performance of the system was evaluated using two datasets, NSL-KDD and N-BaIoT, achieving 86.39% and 99.94% multiclassification accuracy, respectively. Through ablation experiments and comparison with the baseline model, it is found that the inherent limitations of a single machine-learning model directly affect the accuracy of the intrusion detection system, while the superiority of the proposed multi-module model in detecting unknown attacks and rare classes of attack traffic is demonstrated.

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

Citation: Yang, Y.; Gu, Y.; Yan, Y. Machine Learning-Based Intrusion Detection for Rare-Class Network Attacks. *Electronics* **2023**, *12*, 3911. https://doi.org/10.3390/electronics12183911

Academic Editor: Harald Vranken

Received: 22 August 2023
Revised: 14 September 2023
Accepted: 15 September 2023
Published: 16 September 2023

1. Introduction

The Internet of Things (IoT) is one of the biggest technological advances in the last few years [1]. 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. Cisco estimates that, by 2030, there will be approximately 203 billion IoT devices in use worldwide [2]. The explosive growth of IoT devices has made the form of IoT security increasingly critical, and general IoT systems are vulnerable to various cyber attacks [3,4], and common attacks in various fields are shown in Table 1. While the IoT brings many advantages to our production and life, there are serious security challenges with this technology. The Intrusion Detection System (IDS) [5–8] is an intelligent and proactive security method which can effectively face all kinds of cyber attacks within the IoT. IDSs are able to identify unauthorized attacks and are one of the most effective methods to ensure the security of the IoT.

Compared with traditional rule- or signature-based intrusion detection methods [9,10], machine learning-based Intrusion Detection Systems (IDSs) [11,12] can identify anomalous attacks by learning a large amount of IoT security data. Although some achievements have been made, there are still some shortcomings.

First, data imbalance can affect the performance of intrusion detection classifiers [13]. For example, in the NSL-KDD dataset, there are 67,343 normal data out of 125,973 training

data, and 58,630 attack data. The attack data can be further divided into four major categories, among which the rarest User to Root attack occurs only 52 times. The rare class of attack data often accounts for only a small fraction of the normal data, and when using a single machine-learning algorithm for classification, the classifier tends to favor the majority class of data and misclassify the rare class of attack data into normal data. Common solutions include the use of oversampling algorithms such as Synthetic Minority Over-sampling Technique (SMOTE), Borderline-SMOTE, Adaptive Synthetic Sampling (ADASYN), Generative Adversarial Networks (GAN), etc. to generate rare classes of attack data. Although IDSs based on these traditional oversampling techniques alleviate the imbalance problem to some extent, the generated attack samples lack diversity and tend to generate bad samples.

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

Second, due to the influence of data dimensionality, the trained model has a low detection rate for certain classes of attacks, especially unknown attacks. In the face of IoT security data with high-dimensional characteristics [14], IDSs based on techniques such as principal component analysis and Pearson correlation have improved detection performance to some extent. However, these traditional dimensionality reduction methods have a limited ability to extract effective information from high-dimensional data and poor generalization of extracted features, which affect the overall detection rate of IDSs.

To overcome these drawbacks, this paper proposes an IDS for high-dimensional unbalanced IoT traffic, called KGMS-IDS. The contributions of this paper are threefold:

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

The concept of intrusion detection was first introduced by Anderson [15]. Intrusion detection can be used to detect any attack that damages the host system. The main objective of implementing intrusion detection is to find an effective algorithm for network data analysis and detection. Wang et al. [16] preprocessed the dataset and constructed 98 models such as Deep Neural Network (DNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) to determine whether the network traffic constitutes a malicious attack. The proposed intrusion detection mechanism was evaluated by using the comprehensive and advanced CSE-CIC-IDS2018 dataset, and good detection results were achieved. Khraisat et al. [17] proposed a taxonomy of IoT attacks and classified various types of IoT-based intrusion detection techniques based on deployment and verification policies. The authors also discussed the availability of IDS datasets and the challenges faced by IoT-based IDSs. In addition to IDSs, Dutta et al. [18] have also discussed Intrusion Prevention Systems (IPS) and Intrusion Response Systems (IRS). The authors included works that depend on security in the IoT standardized protocol stack by considering nine different dimensions and characteristics. These dimensions are (1) detection techniques, (2) layout strategies, (3) location/visibility, (4) frequency of use, (5) authentication methods, (6) types of attacks, (7) evaluation metrics, (8) datasets and data preprocessing, and (9) feature reduction techniques.

2.1. Dealing with High-Dimensional Data

In recent years, IoT security has received increasingly widespread attention due to the explosive growth of IoT traffic. Machine learning-based IDSs as a defense method capable of proactively detecting attacks remain a hot research topic in the field of IoT security [19]. The network traffic generated by IoT systems has a high-dimensional character. Since the high-dimensional features contain a lot of redundant information, this poses a great challenge for machine learning-based IDSs. For high-dimensional IoT data, efficient feature extraction has become a hot topic in the field of intrusion detection. Mehmood et al. [20] applied the Random Forest Recursive Feature Elimination (RFRFE) method to filter important features in network traffic and improve the overall detection rate of network attacks.

Due to the high computational overhead of RFRFE, in order to solve this problem, Hammad et al. [21] used correlated feature selection and T-distributed random neighborhood embedding for feature reduction. The method achieved a good attack detection rate at CSE-CIC-IDS2018 (CIC-IDS2018). Xie et al. [22] proposed a network intrusion detection algorithm based on dynamic intuitionistic fuzzy sets and used the cardinality test to select the best features. Experiments showed that the intrusion detection model after fusion feature selection outperforms the method using a single classification algorithm. Prajisha et al. [23] focused on cyber attacks in the IoT domain and proposed an enhanced chaotic swarm optimization algorithm to reduce the dimensionality of attack data. By experimenting on three IoT datasets, it is shown that the proposed feature selection method effectively improves the detection rate of IoT attacks. To address the difficulty of DDoS attack detection within the IoT, Kumar et al. [24] proposed training Random Forest (RF) and an optimal gradient tree boosting system (XGBoost) on distributed fog nodes to detect DDoS attacks against mining pools within the IoT, which was validated on public IoT datasets.

However, all the above IDSs use feature selection methods to downscale the high-dimensional network data. These feature selection methods remove a portion of useful features and are sensitive to noise and redundant features. To address this problem, Kunang et al. [25] used deep autoencoder for feature extraction, which reduced the redundancy of high-dimensional data and better preserved the information of the original network data. Lv et al. [26] proposed an intrusion detection model based on Stacked Denoising Autoencoder-Support Vector Machine (SDAE-SVM), and experiments showed that the network data after SDAE dimensionality reduction improved the detection effect of the classifier. Wang et al. [27] extracted representative features of normal data by applying AE, and then trained One-Class Support Vector Machine (OCSVM) and Gaussian Mixture

Model (GMM) on the reduced features. The experimental results show that AE improves the detection rate and the combination of two detectors is better than a single detector. Muhammad et al. [28] proposed an IDS based on stacked AE and DNN. The stacked AE learns the features recorded by the input network in an unsupervised manner to reduce the feature width. The DNN was then trained in a supervised manner to extract deep-learning features for classifiers and was validated on three publicly available cybersecurity datasets.

2.2. Machine-Learning Ensembles

The rapidly evolving and diverse range of attack data within the IoT leads to a high false positive rate in intrusion detection models based on single machine-learning algorithms. To address this issue, Khan et al. [29] proposed an intrusion detection method based on AutoML and soft voting, which improved the detection accuracy on two network security datasets. Aburomman et al. [30] introduced an ensemble intrusion detection model, combining k-nearest neighbors, artificial neural networks, and naive Bayes. This model achieved a 83.43% detection rate for five-class attack detection on the complete NSL-KDD dataset, resolving the issue of unstable training with single classifiers. Hossain et al. [31] proposed a novel integration-based machine-learning technique for intrusion detection, where the integration strategies include Random Forest, Gradient Boost, Adaboost, Gradient XGBoost, Bagging, and Simple Stacking. The performance of the proposed method was evaluated on ten public cybersecurity datasets and good detection results were achieved.

2.3. Dealing with Imbalanced Data

The imbalanced distribution of attacks is one of the challenging research problems in intrusion detection and IoT security. In response to this issue, Zhang et al. [32] utilized Borderline-SMOTE to oversample rare-class attack traffic within the IoT, effectively improving the accuracy of IoT attack detection under sample imbalance conditions. To address the low detection rate of traditional classifiers for rare-class attack traffic within the IoT, Andresini et al. [33] employed GAN to learn the distribution of rare-class attack traffic and generate synthetic rare-class attack data, which partially improved the detection efficiency of intrusion detection models based on Convolutional Neural Networks (CNNs). Kumar et al. [34] utilized Wasserstein Conditional Generative Adversarial Network (WCGAN) to generate rare-class attack samples and then trained an XGBoost classifier with a balanced dataset. Through testing on publicly available IoT datasets, the training data oversampled by WCGAN demonstrated better classifier training and improved the recognition rate of intrusion detection models for rare-class attacks.

SMOTE [35,36], Borderline-SMOTE, ADASYN, and other algorithms [37,38] are classical oversampling methods. However, they all generate synthetic samples along the line segments connecting minority class instances, making it difficult to improve the distribution of minority class samples. G- SMOTE [39] is a synthetic minority oversampling technique designed to address the issue of imbalanced data. It is an improved version based on the SMOTE algorithm. Unlike the aforementioned SMOTE-based algorithms, G-SMOTE does not simply generate synthetic samples along the line segments connecting minority class instances. Instead, it expands the linear interpolation mechanism by introducing geometric transformations in the feature space, better simulating the distribution of rare-class samples.

2.4. Our Approach

Although the above methods have improved the detection rate of IoT attack traffic, they have shown less satisfactory performance in training and detecting unknown attacks. Additionally, the IDSs mentioned above have not simultaneously addressed the challenges posed by high-dimensional class imbalance. In this study, based on previous research, we propose an integrated IDS to address these issues. The system consists of three main modules.

The MDSAE module is used for feature dimensionality reduction. Based on [26–28], by adding the self-attention mechanism and multiple noise, MDSAE can enhance the robustness of the dimensionality reduction data and improve the detection rate of SVEDM for unknown attacks. The KGSMOTE module is used to generate rare-class attack data. On the basis of [35–39], by improving the G-SMOTE mechanism, rare-class attack samples with diversity are generated to enhance the detection rate of SVEDM on rare-class attacks. Based on [29–31], through a large number of experiments, the combination model with superior performance is selected, and the low-dimensional balanced data are fed into SVEDM for multiclassification anomaly detection. These modules are interconnected and enhance each other to improve the detection rate of IDS for unknown and rare-class attack traffic. Table 2 compares the detection methods of the various models mentioned above with the proposed method presented in this study.

Table 2. Comparison with relevant survey results.

Problem Solved	Methods	Datasets	Ensemble Leaning	Unknown Attack	Year
Feature Dimensionality Reduction.	RFRFE-FGSVM [20]	NSL-KDD	-	-	2022
	MMM-RF [21]	CSE-CIC-IDS2018	-	-	2022
	IFSs [22]	KDD 99/NSL-KDD/UNSW-NB15	-	-	2021
	ECSSA-LightGBM [23]	MC-IoT/MQTTset/MQTT-IoT-IDS2020	-	-	2022
	RF/XGBoost [24]	BoT-IoT	-	-	2022
	PTDAE-DNN [25]	NSL-KDD/CSE-CIC-IDS2018	-	-	2021
	SDAE-SVM [26]	NSL-KDD	-	-	2020
	AE-OCSVM-GMM [27]	NF-BoT-IoT-V2/NF-CSE-CIC-IDS2018-V2	-	-	2023
	SAE-DNN [28]	KDDCup99/NSL-KDD/aegean WIFI	-	-	2023
Machine-learning classifier.	OE-IDS [29]	UNSW-NB15/CIC-IDS2017	✓	-	2023
	ACOR-WMV [30]	NSL-KDD	✓	-	2022
	PCA-EL [31]	Ten public cybersecurity datasets	✓	-	2023
Imbalance processing.	ICVAE-BSM [32]	NSL-KDD/CIC-IDS2017/ CIC-IDS2018	-	-	2022
	GAN-CNN [33]	KDDCUP99/UNSW-NB15/ CIC-IDS2017/AAGM17	-	-	2021
	WCGAN-XGBoost [34]	NSL-KDD/BoT-IoT/UNSW-NB15	-	-	2023
	SCADA-IDS [35]	Morris power dataset/CIC-IDS2017	-	-	2023
Both	Proposed method	NSL-KDD/N-BaIoT	✓	✓	2023

3. Method

3.1. Model Structure

The structure of the KGMS-IDS proposed here is shown in Figure 1. It mainly consists of three modules: imbalance processing module, feature dimensionality reduction module, and classification module. Each module is optimized by a hyper-parameter search through experience and a large number of experiments, and good detection results are obtained, with specific parameters 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. 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 un-

known 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 within the 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.

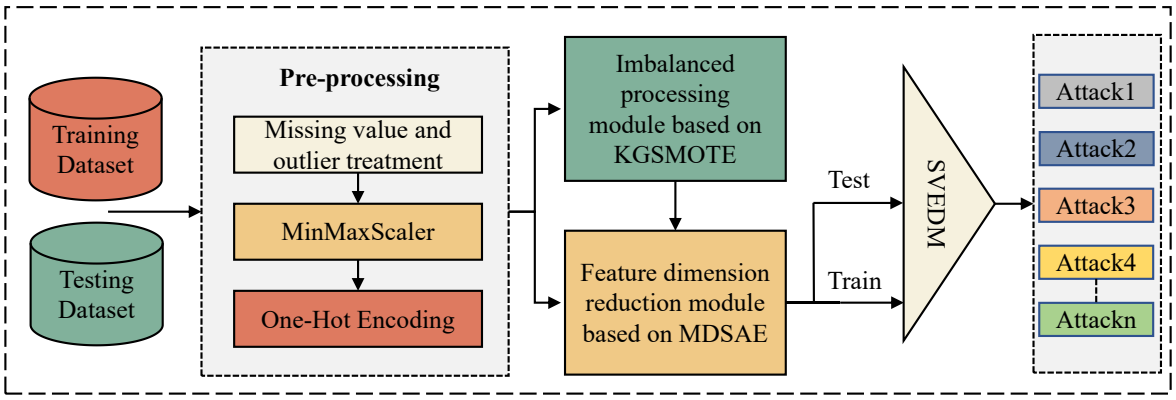

Figure 1. Framework of the proposed model.

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

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 are input to the next imbalance processing module for imbalance processing.

2. Imbalance processing module: The imbalance processing module is mainly based on the random downsampling algorithm and the KGSMOTE algorithm. The training set in the data after pre-processing is taken out, and the training set is input 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 are generated by the KGSMOTE algorithm. It should be noted that the KGSMOTE model only performs 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 are input to the feature downsampling module for feature downsampling.
3. Feature reduction module: The training data processed by the imbalance processing module are input 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 are input into 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 is input into the trained classification module for multi-classification anomaly detection, and the final detection results are obtained.

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.

G-SMOTE introduces a geometric region, i.e., a hypersphere, wherein rare-class attack data are synthesized around each rare-class attack sample with a safe radius. Typically, this geometric region in input space is a truncated hyperellipsoid. Specifically, instead of synthesizing new rare-class attack data along line segments connecting minority class instances, G-SMOTE defines a flexible geometric region around each selected rare-class attack sample and increases the diversity of generated samples by expanding the area of the minority class.The core idea of G-SMOTE is to generate a random point, e_{sphere}, on the surface of the unit hypersphere using Equation (1). By applying Equation (2), the e_{sphere} point is transformed into a randomly generated x_{gen} point within the unit hypersphere. The end result of this process is the generation of a uniformly distributed random point within the unit hypersphere.

$$e_{\text{sphere}} \leftarrow \frac{v_{\text{normal}}}{\| v_{\text{normal}} \|} \tag{1}$$

$$x_{\text{gen}} \leftarrow r^{1/p} e_{\text{sphere}} \tag{2}$$

v_{normal} is generated by p random numbers from the normal distribution N (0, 1). r is a random number from the uniform distribution U (0, 1).

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. Algorithm 1 illustrates the operation process of the KGSMOTE module.

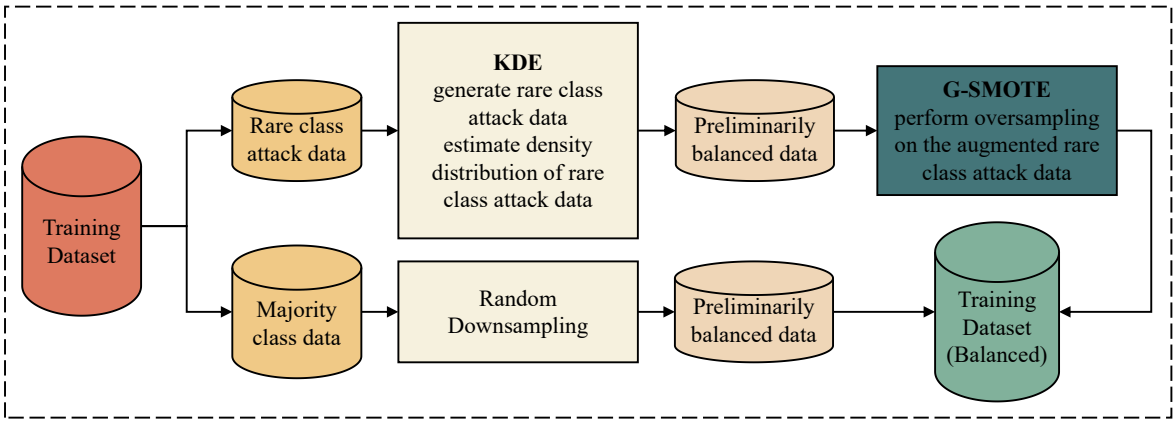

Figure 2. Imbalanced data processing module based on KGSMOTE.

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**

Firstly, the KDE [40] 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 (3). 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, x_i 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.

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3}$$

3.3. Feature Dimension Reduction Module Based on MDSAE

The Autoencoder (AE) [41,42] 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 means that it cannot effectively improve the multi-classification accuracy of the detection model. The Denoising Autoencoder (DAE) [43,44] 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 a self-attention mechanism [45,46] 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 Equations (4) and (5).

$$z = f(\omega_e x + b^e) \tag{4}$$

$$y = g\left(\omega_d z + b^d\right) \tag{5}$$

Equation (4) demonstrates how the encoder compresses high-dimensional raw data into latent features, where we represent the weight matrix and bias vector of the encoder, respectively. Equation (5) illustrates how the decoder restores low-dimensional latent features to high-dimensional raw data, where w_d and b_d 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). Based on experience and a large number of experiments, it is concluded that MDSAE-IDS achieves excellent classification results when it contains two hidden layers. The hidden layer (z_1) contains 80 neurons and the final layer (z) contains 30 neurons. The self-attention mechanism layer in the encoder is added between the hidden layer (z_1) and the final layer (z). The number of output nodes of the self-attention mechanism layer is 50. The self-attention mechanism layer can improve the ability of AE to capture the relationship between high-dimensional data, so that it can take more into account the interactions between individual features in the feature dimensionality reduction process. Meanwhile, adding the self-attention

mechanism layer to the encoder can alleviate the overfitting problem of the self-encoder and improve the ability of AE to characterize high-dimensional data. Therefore, adding the self-attention mechanism layer can improve the robustness of AE together with multiple noise. 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 b_e and b_d 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 exhibit improved robustness. Additionally, the MDSAE module employs the Huber Loss [47] as the loss function. Finally, by extracting the trained encoder part from the MDSAE, high-dimensional features are reduced to 30 dimensions.

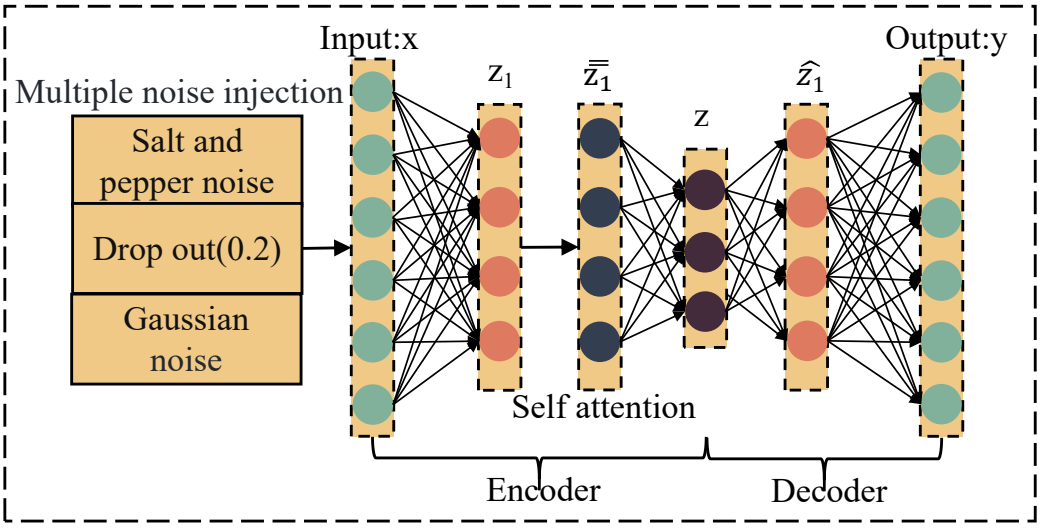

Figure 3. Traditional autoencoders and multiple denoising autoencoders improved by the self-attention mechanism.

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 were chosen as the base classifiers for SVEDM. The flow chart of SVEDM is depicted in Figure 4.

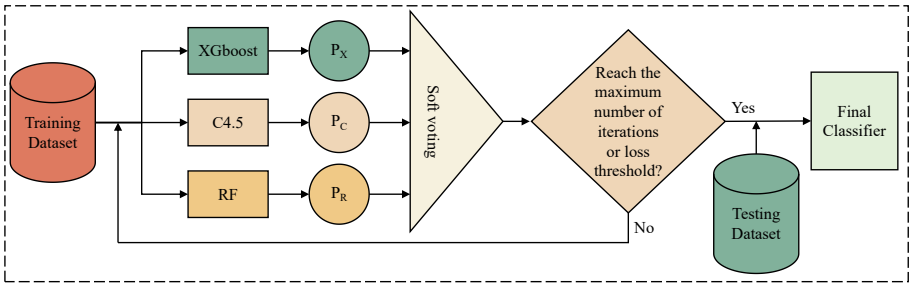

Figure 4. A multi-class anomaly detection module based on SVEDM.

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 [48] is an improved version of the KDD Cup 99 [49]. 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.This setup aims to better simulate real-world scenarios because, in the real world, new types of attacks are constantly emerging and many of them evolve and modify to evade existing defense systems. Therefore, in order to effectively evaluate the performance of intrusion detection systems in the face of unknown attacks, these unknown attacks are included in the test set. For the unknown attacks in the test set, machine-learning algorithms and detection methods are usually used for classification and detection. Using the information of known attacks in the training set, attack patterns and features are learned to try to identify and classify unknown attacks.

Table 4. Categories and partitioning of the NSL-KDD dataset.

Class	KDDTrain+	Number	KDDTest+	(Unknow Attack)	Number
Normal	normal	67,343	normal	\	9711
DoS	back, land, neptune, pod, smurf, teardrop	45,927	back, land, neptune, smurf, teardrop, pod	apache2, mailbomb, processtable, udpstorm	7458
Probe	ipsweep, nmap, portsweep, satan	11,656	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	125,973	21	17	22,544

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.

3.5.2. N-BaIoT

The N-BaIoT dataset [50,51] was specifically developed for intrusion 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 are 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 is 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. However, detecting these unknown attacks can be challenging due to their previously unseen features and behaviors. When evaluating with the N-BaIoT dataset, for the unknown attacks in the test set, we focus more on metrics such as the accuracy, recall, and false alarm rate of the system. These metrics will evaluate whether the system is able to correctly identify unknown attacks in order to validate the system's detection capability in the face of novel attacks.

Table 5. Categories and partitioning of the N-BaIoT dataset.

Class	N-BaIoT Train	Number	N-BaIoT Test	(Unknow Attack)	Number
Normal	normal	34,806	normal	\	14,917
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	47,726	8	3	26,358

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 2.30 GHz
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:

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{9}$$

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. In addition, this paper evaluates the performance of the proposed model using the confusion matrix and Kappa coefficient. The confusion matrix is a situation analysis table used in machine learning to summarize the predictions of classification models. It summarizes the records in the dataset in a matrix form based on two criteria, i.e., the true categories and the category judgments predicted by the classification model. Kappa is a statistical metric used to measure the consistency between classifiers or evaluators. It evaluates classification tasks by calculating the level of consistency as a means of evaluating more than accuracy. Its core idea is shown in Equation (10).

$$\text{Kappa } = \frac{(p_o - p_e)}{1 - p_e} \tag{10}$$

where P_o is the observed exact agreement between classifiers or evaluators and P_e is the consistency achieved by the classifier or evaluator in a randomized situation.

4.2. Imbalanced Processing Based on KGSMOTE

First, the training data are 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 Figures 5 and 6.

Figures 5a and 6a 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 5b and 6b. 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 the 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 Figures 7 and 8.

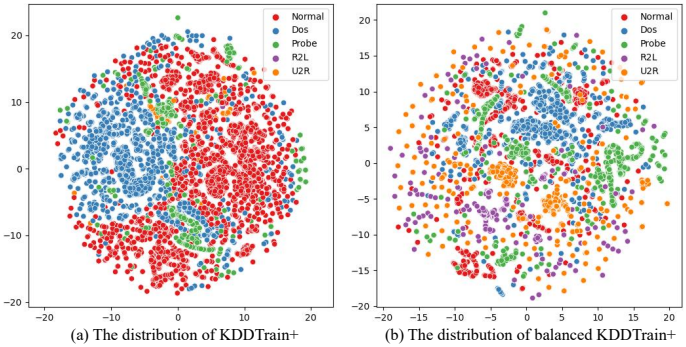

Figure 5. UMAP visualization based on KDDTrain+.

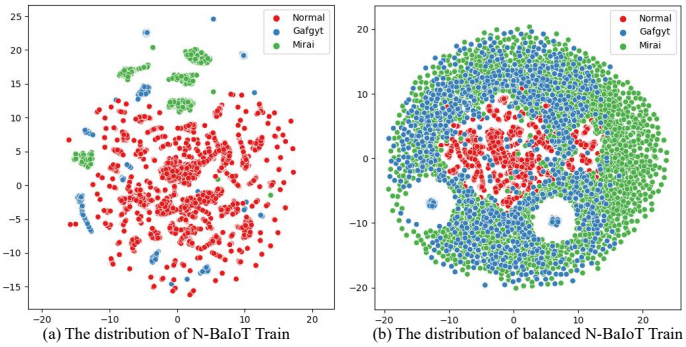

Figure 6. UMAP visualization based on N-BaIoT Train.

The 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 Figures 7 and 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 the 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.

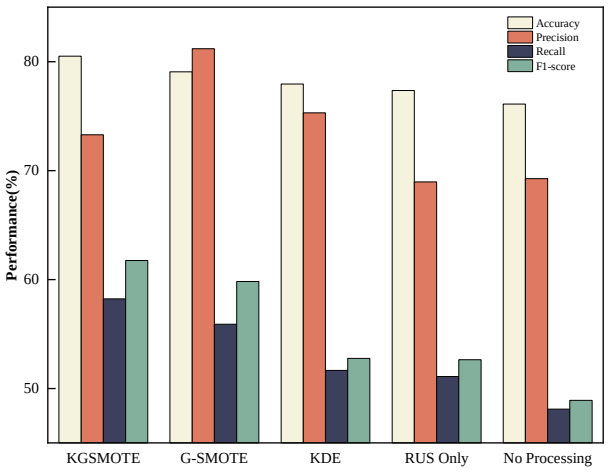

Figure 7. The classification results after performing imbalance treatment on NSL-KDD dataset using KGSMOTE module and its components.

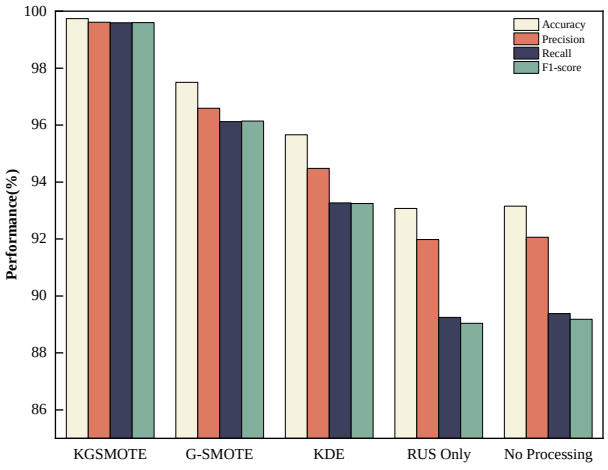

Figure 8. The classification results after performing imbalance treatment on N-BaIoT dataset using KGSMOTE module and its components.

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 Figures 9 and 10.

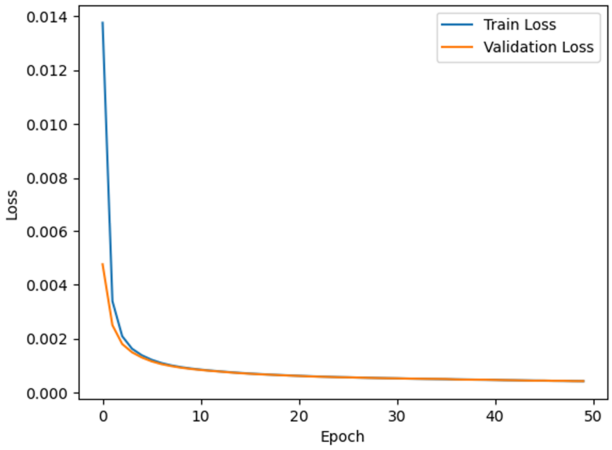

Figure 9. MDSAE loss-epoch curves trained on the NSL-KDD dataset.

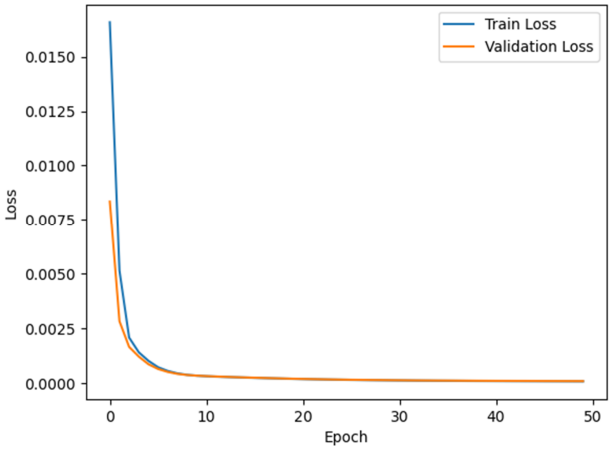

Figure 10. MDSAE loss-epoch curves trained on the N-BaIoT dataset.

Figures 9 and 10 display 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 Figures 11 and 12.

Figures 11 and 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 the baseline model AE in all four metrics. In addition, 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.

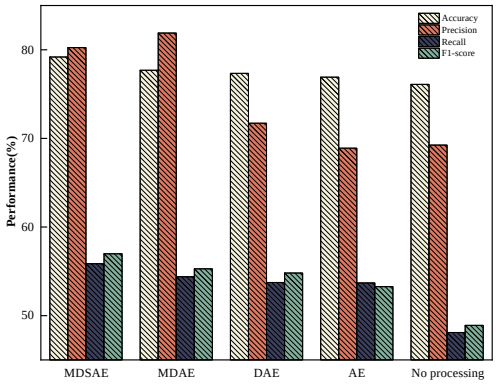

Figure 11. The classification results after performing dimensionality reduction on NSL-KDD dataset using MDSAE module and its components.

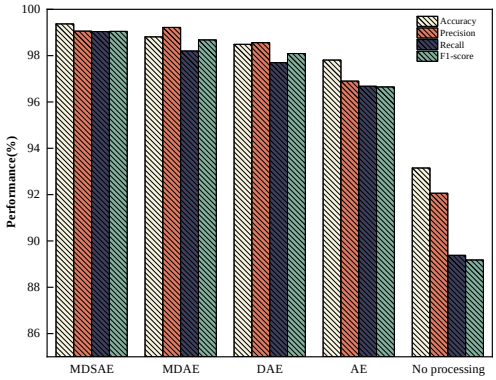

Figure 12. The classification results after performing dimensionality reduction on N-BaIoT dataset using MDSAE module and its components.

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 Figures 13 and 14. Figures 13a and 14a show 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. Figures 13b and 14b show 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.

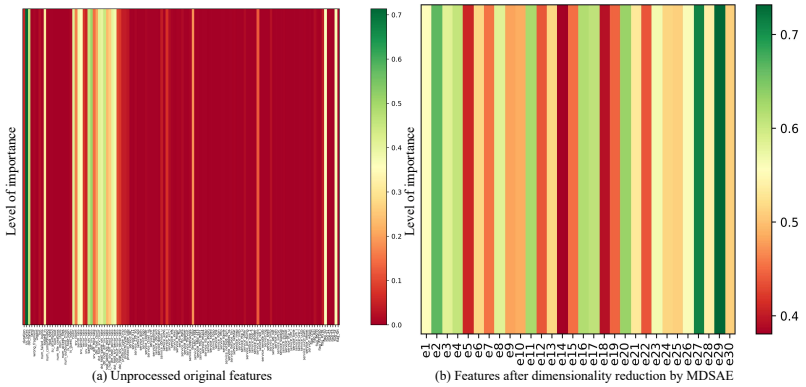

Figure 13. The importance level between each feature and the target variable on the NSL-KDD dataset.

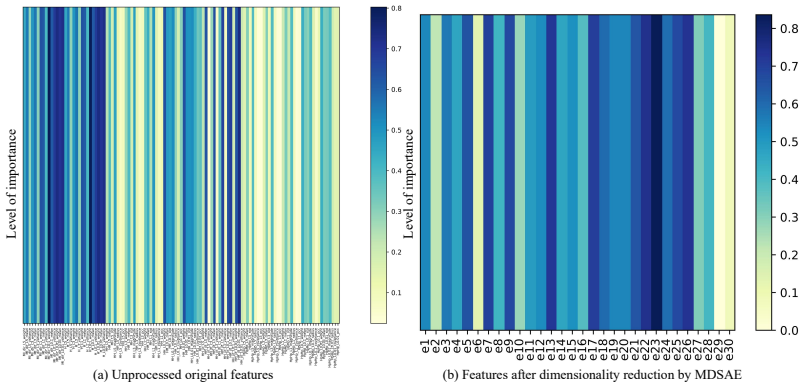

Figure 14. The importance level between each feature and the target variable on the N-BaIoT dataset.

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.

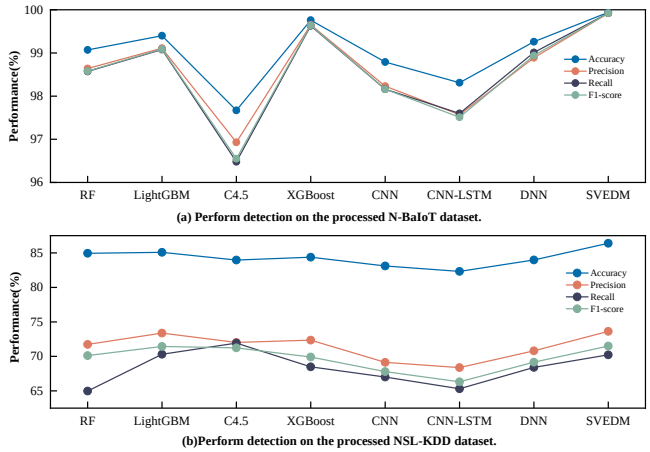

Figure 15. Using the SVEDM module for multi-class anomaly detection on dimensionality reduced and balanced dataset.

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 Tables 7 and 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%

Tables 7 and 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-score. On the N-BaIoT dataset, the model using XGBoost, C4.5, and RF as base classifiers achieved the highest Acc, Pre, Re, and F1-score. 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

Tables 9 and 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, F1-score and Kappa, 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, F1-score, and Kappa. 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 Re of 95.85%, while using the CNN algorithm achieved an Acc of 99.68% and 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 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 Figures 16 and 17.

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

Method	Accuracy	Precision	Recall	F1-Score	Kappa
RF	75.88%	88.68%	49.11%	50.36%	61.40%
C4.5	75.62%	79.03%	49.68%	49.48%	65.70%
XGBoost	75.52%	68.08%	46.81%	48.01%	60.54%
LightGBM	75.70%	80.00%	49.36%	52.69%	60.91%
CNN	79.75%	86.46%	56.96%	59.71%	67.90%
CNN-LSTM	78.14%	67.27%	53.52%	53.10%	65.70%
DAE-SMOTE-DNN	81.31%	68.18%	61.59%	63.78%	71.62%
Proposed method	86.39%	73.62%	70.22%	71.49%	79.74%

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

Method	Accuracy	Precision	Recall	F1-Score	Kappa
RF	93.79%	92.64%	90.36%	90.24%	89.39%
C4.5	97.28%	96.20%	95.85%	95.84%	95.36%
XGBoost	90.36%	89.59%	85.04%	84.55%	83.45%
LightGBM	88.42%	88.46%	82.03%	80.94%	80.14%
CNN	99.68%	99.52%	99.61%	99.56%	99.46%
CNN-LSTM	88.13%	87.77%	81.71%	80.34%	79.73%
DAE-SMOTE-DNN	99.78%	99.86%	99.69%	99.77%	99.64%
Proposed method	99.94%	99.92%	99.92%	99.92%	99.90%

Figures 16 and 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.

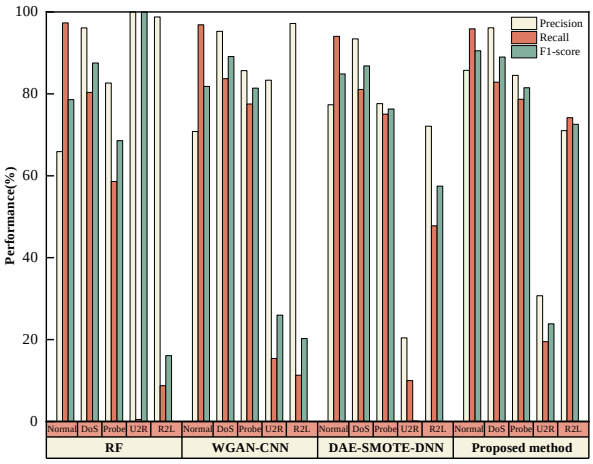

Figure 16. The performance of KGMS-IDS on the NSL-KDD dataset.

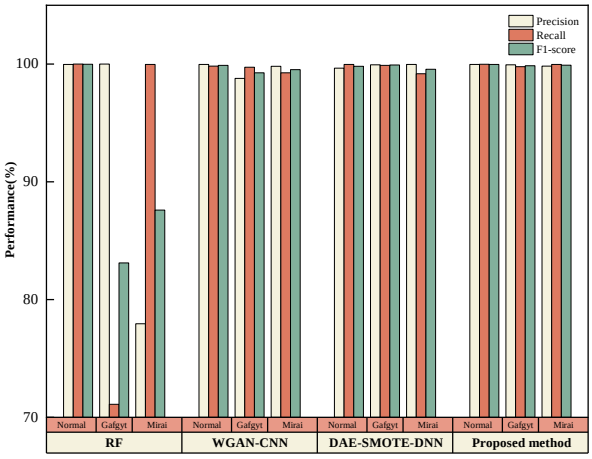

Figure 17. The performance of KGMS-IDS on the N-BaIoT dataset.

This result is further illustrated in the confusion matrices shown in Figures 18 and 19. Figures 18 and 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 Tables 11 and 12.

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

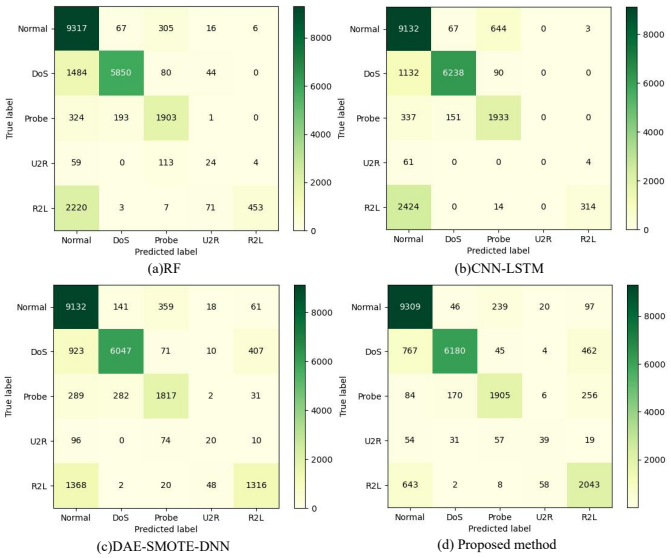

Figure 18. Confusion matrix on the NSL-KDD dataset.

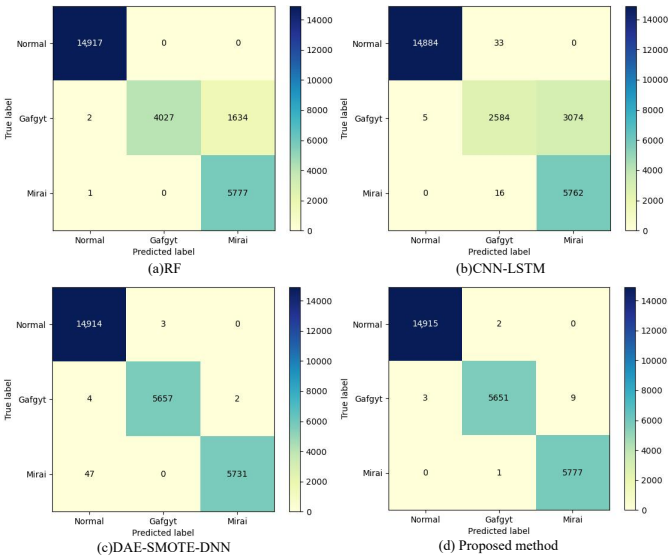

Figure 19. Confusion matrix on the N-BaIoT dataset.

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.

Tables 11 and 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 has been compared to state-of-the-art intrusion detection methods in recent years, as shown in Table 13. It can be seen that all the detection methods [52–56] for the NSL-KDD dataset are effective in detecting unknown attacks. In particular, CS-NN achieved an accuracy of 85.56%, which is better than several other models. However, the above models are less effective in detecting rare-class attack data. Methods [57–60] were tested on the N-BaIoT dataset, and all of them achieved high detection rates, but the above methods did not validate the detection effect on unknown attacks. Comparing the best performance of the proposed KGMS-IDS with the above state-of-the-art intrusion detection methods, KGMS-IDS achieves the highest accuracy rate on both datasets. In addition, by comparing with the baseline model (GSMOTE-AE-RF), the proposed model improves the 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 the IoT, achieving intelligent intrusion detection.

Table 13. Comparison results between different detection models and KGMS-IDS.

Model	Year	Datasets	Accuracy	Classification	Unknown Attack
MDPCA-DBN [52]	2019	NSL-KDD	82.08	Multi (5)	✓
LCVAE [53]	2020	NSL-KDD	85.51	Multi (5)	✓
CAFE-CNN [54]	2021	NSL-KDD	83.34	Multi (5)	✓
ID-UL [55]	2022	NSL-KDD	81.48	Multi (5)	✓
CS-NN [56]	2022	NSL-KDD	85.56	Multi (5)	✓
LGBA-NN [57]	2022	N-BaIoT	90.00	Multi (11)	-
SGAN-IDS [58]	2022	N-BaIoT	99.89	Binary (2)	-
EL-DTs [59]	2022	N-BaIoT	99.60	Multi (10)	-
Cu-DNNGRU [60]	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. Conclusions and Future Work

IDS is a critical service used to monitor networks for malicious activity, such as security attacks. Difficulties are encountered due to the complex and dynamic environment of the network, which may severely impact the performance of existing IDS. These difficulties include unbalanced, high-dimensional, and asymmetrically distributed datasets. In this paper, we detect anomalies by proposing an integrated intrusion detection framework, KGMS-IDS. KGMS-IDS detects anomaly based intrusions in the IoT and also improves the class imbalance problem that is prevalent in many domain datasets. The multiple noise reduction technique is implemented to better extract the valid information and improve the model robustness for the detection of unknown attacks. 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 the 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. Ensemble learning is costly in terms of training time, testing time, and computational overhead. As a result, this leads to high latency and resource utilization, which significantly affects the real-time functionality of smart IDS. In addition, KGMS-IDS is an integrated model, and other intrusion detection methods using a single machine-learning model, to a certain degree, increased 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.

In order to meet the performance requirements of IoT's high speed and low latency, IDSs also need to face diverse network attacks. In this case, intrusion detection of large-scale network traffic is imperative for IoT edge networks. Therefore, lightweight and high-precision intrusion detection methods for edge networks may be a major research direction in the future. Distributed intelligent IDS architecture helps to reduce the computational overhead, which in turn improves the detection rate and performance of KGMS-IDS by reducing the false alarm rate and training and testing time. Since IoT has a huge amount of data and is limited by the computational power of IoT devices, our future plan is to continue researching faster and better feature reduction methods. We also plan to address issues related to latency and resource utilization, adaptation to dynamic IoT environments, and deployment architecture. The goal is to minimize the complexity of the model while ensuring the detection accuracy. Meanwhile, in 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 the IoT.

Author Contributions: Conceptualization, Y.Y. (Yu Yang); methodology, Y.G.; software, Y.G.; validation, Y.Y. (Yu Yan); formal analysis, Y.Y. (Yu Yan); writing—original draft preparation, Y.Y. (Yu Yang) and Y.G.; writing—review and editing, Y.Y. (Yu Yan), Y.Y. (Yu Yang), and Y.G. 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. 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 Things J.* **2023**, *10*, 4059–4092. [CrossRef]
2. Evans, D. How the Next Evolution of the Internet Is Changing Everything. Internet Things 2011. Available online: http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf (accessed on 22 September 2021).
3. Wang, M.; Yang, N.; Weng, N. Securing a Smart Home with a Transformer-Based IoT Intrusion Detection System. *Electronics* **2023**, *12*, 2100. [CrossRef]
4. Alazab, A.; Khraisat, A.; Singh, S.; Bevinakoppa, S.; Mahdi, O.A. Routing attacks detection in 6lowpan-based internet of things. *Electronics* **2023**, *12*, 1320. [CrossRef]
5. Alani, M.M.; Awad, A.I. An Intelligent Two-Layer Intrusion Detection System for the Internet of Things. *IEEE Trans. Ind. Inform.* **2022**, *19*, 683–692. [CrossRef]

6. Qu, Y.; Ma, H.; Jiang, Y.; Bu, Y. A Network Intrusion Detection Method Based on Domain Confusion. *Electronics* **2023**, *12*, 1255. [CrossRef]
7. Kim, T.; Pak, W. Scalable Inline Network-Intrusion Detection System with Minimized Memory Requirement. *Electronics* **2023**, *12*, 2061. [CrossRef]
8. Yang, L.; Moubayed, A.; Shami, A. MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles. *IEEE Internet Things J.* **2021**, *9*, 616–632. [CrossRef]
9. Zhang, Y.; Liu, H.; Dong, X.; Li, C.; Zhang, Z. HyIDSVis: Hybrid intrusion detection visualization analysis based on rare category and association rules. *J. Vis.* **2022**, *25*, 175–190. [CrossRef]
10. Erlacher, F.; Dressler, F. On high-speed flow-based intrusion detection using snort-compatible signatures. *IEEE Trans. Dependable Secur. Comput.* **2020**, *19*, 495–506. [CrossRef]
11. Zhang, C.; Jia, D.; Wang, L.; Wang, W.; Liu, F.; Yang, A. Comparative research on network intrusion detection methods based on machine learning. *Comput. Secur.* **2022**, *121*, 102861. [CrossRef]
12. Apruzzese, G.; Pajola, L.; Conti, M. The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 5152–5169. [CrossRef]
13. Liu, C.; Antypenko, R.; Sushko, I.; Zakharchenko, O. Intrusion Detection System After Data Augmentation Schemes Based on the VAE and CVAE. *IEEE Trans. Reliab.* **2022**, *71*, 1000–1010. [CrossRef]
14. Telikani, A.; Shen, J.; Yang, J.; Wang, P. Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing. *IEEE Internet Things J.* **2022**, *9*, 23260–23271. [CrossRef]
15. Ganesan, R.; Jajodia, S.; Cam, H. Optimal scheduling of cybersecurity analysts for minimizing risk. *ACM Trans. Intell. Syst. Technol. (TIST)* **2017**, *8*, 1–32. [CrossRef]
16. Wang, Y.C.; Houng, Y.C.; Chen, H.X.; Tseng, S.M. Network Anomaly Intrusion Detection Based on Deep Learning Approach. *Sensors* **2023**, *23*, 2171. [CrossRef] [PubMed]
17. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. *Cybersecurity* **2021**, *4*, 18. [CrossRef]
18. Dutta, M.; Granjal, J. Towards a secure Internet of Things: A comprehensive study of second line defense mechanisms. *IEEE Access* **2020**, *8*, 127272–127312.
19. 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**, *10*, 121173–121192. [CrossRef]
20. 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. [CrossRef]
21. Hammad, M.; Hewahi, N.; Elmedany, W. MMM-RF: A novel high accuracy multinomial mixture model for network intrusion detection systems. *Comput. Secur.* **2022**, *120*, 102777. [CrossRef]
22. Xie, J.; Wang, H.; Garibaldi, J.M.; Wu, D. Network Intrusion Detection Based on Dynamic Intuitionistic Fuzzy Sets. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 3460–3472. [CrossRef]
23. Prajisha, C.; Vasudevan, A. An efficient intrusion detection system for MQTT-IoT using enhanced chaotic salp swarm algorithm and LightGBM. *Int. J. Inf. Secur.* **2022**, *21*, 1263–1282. [CrossRef]
24. 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. *J. Parallel Distrib. Comput.* **2022**, *164*, 55–68. [CrossRef]
25. Kunang, Y.N.; Nurmaini, S.; Stiawan, D.; Suprapto, B.Y. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *J. Inf. Secur. Appl.* **2021**, *58*, 102804. [CrossRef]
26. Lv, Z.; Qiao, L.; Li, J.; Song, H. Deep-learning-enabled security issues in the internet of things. *IEEE Internet Things J.* **2020**, *8*, 9531–9538. [CrossRef]
27. Wang, C.; Sun, Y.; Lv, S.; Wang, C.; Liu, H.; Wang, B. Intrusion Detection System Based on One-Class Support Vector Machine and Gaussian Mixture Model. *Electronics* **2023**, *12*, 930. [CrossRef]
28. Muhammad, G.; Hossain, M.S.; Garg, S. Stacked autoencoder-based intrusion detection system to combat financial fraudulent. *IEEE Internet Things J.* **2020**, *10*, 2071–2078. [CrossRef]
29. Khan, M.A.; Iqbal, N.; Jamil, H.; Kim, D.H. An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection. *J. Netw. Comput. Appl.* **2023**, *212*, 103560. [CrossRef]
30. Albashish, D.; Aburomman, A. Weighted heterogeneous ensemble for the classification of intrusion detection using ant colony optimization for continuous search spaces. *Soft Comput.* **2022**, *27*, 4779–4793. [CrossRef]
31. Hossain, M.A.; Islam, M.S. Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array* **2023**, *19*, 100306. [CrossRef]
32. Zhang, Y.; Liu, Q. On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. *Future Gener. Comput. Syst.* **2022**, *133*, 213–227. [CrossRef]
33. Andresini, G.; Appice, A.; De Rose, L.; Malerba, D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. *Future Gener. Comput. Syst.* **2021**, *123*, 108–127. [CrossRef]
34. Kumar, V.; Sinha, D. Synthetic attack data generation model applying generative adversarial network for intrusion detection. *Comput. Secur.* **2023**, *125*, 103054. [CrossRef]

35. 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. [CrossRef] [PubMed]
36. 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. *J. Inf. Secur. Appl.* **2023**, *72*, 103405. [CrossRef]
37. Lavanya, T.; Rajalakshmi, K. Heterogenous ensemble learning driven multi-parametric assessment model for hardware Trojan detection. *Integration* **2023**, *89*, 217–228. [CrossRef]
38. Liu, J.; Gao, Y.; Hu, F. A fast network intrusion detection system using adaptive synthetic oversampling and LightGBM. *Comput. Secur.* **2021**, *106*, 102289. [CrossRef]
39. Douzas, G.; Bacao, F. Geometric SMOTE a geometrically enhanced drop-in replacement for SMOTE. *Inf. Sci.* **2019**, *501*, 118–135. [CrossRef]
40. Kamalov, F.; Moussa, S.; Avante Reyes, J. KDE-Based Ensemble Learning for Imbalanced Data. *Electronics* **2022**, *11*, 2703. [CrossRef]
41. Boppana, T.K.; Bagade, P. GAN-AE: An unsupervised intrusion detection system for MQTT networks. *Eng. Appl. Artif. Intell.* **2023**, *119*, 105805. [CrossRef]
42. Mushtaq, E.; Zameer, A.; Umer, M.; Abbasi, A.A. A two-stage intrusion detection system with auto-encoder and LSTMs. *Appl. Soft Comput.* **2022**, *121*, 108768. [CrossRef]
43. 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. *Comput. Commun.* **2022**, *194*, 55–65. [CrossRef]
44. 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 Trans. Reliab.* **2022**, *72*, 498–510. [CrossRef]
45. Tseng, S.M.; Yeh, Z.T.; Wu, C.Y.; Chang, J.B.; Norouzi, M. Video Scene Detection Using Transformer Encoding Linker Network (TELNet). *Sensors* **2023**, *23*, 7050. [CrossRef]
46. Islam, M.M.; Hasan, M.; Athrey, K.S.; Braskich, T.; Bertasius, G. Efficient Movie Scene Detection using State-Space Transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 18749–18758.
47. Xie, J.; Liu, S.; Chen, J.; Jia, J. Huber loss based distributed robust learning algorithm for random vector functional-link network. *Artif. Intell. Rev.* **2023**, *56*, 8197–8218. [CrossRef]
48. Revathi, S.; Malathi, A. A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection. *Int. J. Eng. Res. Technol. (IJERT)* **2013**, *2*, 1848–1853.
49. 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, Ottawa, ON, Canada, 8–10 July 2009; pp. 1–6.
50. 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 Comput.* **2018**, *17*, 12–22. [CrossRef]
51. 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 Things J.* **2021**, *9*, 3930–3944. [CrossRef]
52. 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. *Appl. Sci.* **2019**, *9*, 238. [CrossRef]
53. Xu, X.; Li, J.; Yang, Y.; Shen, F. Toward effective intrusion detection using log-cosh conditional variational autoencoder. *IEEE Internet Things J.* **2020**, *8*, 6187–6196. [CrossRef]
54. Shams, E.A.; Rizaner, A.; Ulusoy, A.H. A novel context-aware feature extraction method for convolutional neural network-based intrusion detection systems. *Neural Comput. Appl.* **2021**, *33*, 13647–13665. [CrossRef]
55. Li, X.; Kong, K.; Shen, H.; Wei, Z.; Liao, X. Intrusion detection method based on imbalanced learning classification. *J. Exp. Theor. Artif. Intell.* **2022**, 1–21. [CrossRef]
56. Rani, M. Effective network intrusion detection by addressing class imbalance with deep neural networks multimedia tools and applications. *Multimed. Tools Appl.* **2022**, *81*, 8499–8518. [CrossRef]
57. Om Kumar, C.; Marappan, S.; Murugeshan, B.; Beaulah, P.M.R. Intrusion Detection Model for IoT Using Recurrent Kernel Convolutional Neural Network. *Wirel. Pers. Commun.* **2023**, *129*, 783–812. [CrossRef]
58. 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), Barcelona, Spain, 1–3 August 2022; pp. 1–5.
59. Abu Al-Haija, Q.; Al-Dala'ien, M. ELBA-IoT: An ensemble learning model for botnet attack detection in IoT networks. *J. Sens. Actuator Netw.* **2022**, *11*, 18. [CrossRef]
60. 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. [CrossRef]