CSAGC-IDS: A Dual-Module Deep Learning Network Intrusion Detection Model for Complex and Imbalanced Data

Yifan Zeng

School of Cyber Science and Engineering, Southeast University, Nanjing , China
yifanzeng0615@foxmail.com

Abstract. As computer networks proliferate, the gravity of network intrusions has escalated, emphasizing the criticality of network intrusion detection systems for safeguarding security. While deep learning models have exhibited promising results in intrusion detection, they face challenges in managing high-dimensional, complex traffic patterns and imbalanced data categories. This paper presents CSAGC-IDS, a network intrusion detection model based on deep learning techniques. CSAGC-IDS integrates SC-CGAN, a self-attention-enhanced convolutional conditional generative adversarial network that generates high-quality data to mitigate class imbalance. Furthermore, CSAGC-IDS integrates CSCA-CNN, a convolutional neural network enhanced through cost sensitive learning and channel attention mechanism, to extract features from complex traffic data for precise detection. Experiments conducted on the NSL-KDD dataset. CSAGC-IDS achieves an accuracy of 84.55% and an F1-score of 84.52% in five-class classification task, and an accuracy of 91.09% and an F1 score of 92.04% in binary classification task.Furthermore, this paper provides an interpretability analysis of the proposed model, using SHAP and LIME to explain the decision-making mechanisms of the model.

Keywords: Network Intrusion Detection · Data Imbalance · Deep Learning

1 Introduce

1.1 Background

With the widespread adoption of network technology, the consequences of cyberattacks have become increasingly severe [1], and traditional network security techniques [2] are no longer adequate to meet the demands. Network-based Intrusion Detection System can effectively monitor network traffic and detect anomalies [1]. Machine learning, especially deep learning [3,4], has demonstrated exceptional performance in intrusion detection, but it faces challenges in dealing with imbalanced data and high-dimensional complex data. While deep learning Network-based Intrusion Detection Models (NIDMs) can identify common attacks, their ability to detect rare attacks is insufficient, affecting overall performance [5, 6]. Moreover, deep learning NIDMs still encounter difficulties when

handling high-dimensional and complex data [7]. High-dimensional traffic data implies a large number of features, complex data patterns, as well as intricate relationships between features, which, along with the increasing complexity of models, pose challenges to the capabilities and structures of deep learning NIDMs. Therefore, further research is needed to enhance the performance of NIDMs in detecting rare attacks under high-dimensional, complex, and imbalanced data conditions.

1.2 Research Content and Contributions

In response to the challenge of analyzing high-dimensional, intricate, and imbalanced intrusion traffic data, CSAGC-IDS intrusion detection model is proposed.

SC-CGAN. To tackle the issue of data imbalance, the imbalanced data processing algorithm SC-CGAN is proposed. This approach leverages self-attention mechanisms and CNNs to effectively fuse conditional information and capture intricate feature dependencies, ultimately leading to the generation of higher-quality new data. This balanced dataset serves as a valuable resource for subsequent traffic classification tasks. Experimental evaluations have verified that SC-CGAN outperforms other comparative methods.

CSCA-CNN. For the handling of complexly high-dimensional traffic data, the traffic classification algorithm CSCA-CNN is proposed. This approach integrates channel attention with cost-sensitive learning to extract features and assigns higher costs to minority classes to mitigate imbalanced bias. Experimental results demonstrate that CSCA-CNN surpasses other comparative methods.

CSAGC-IDS. By integrating SC-CGAN and CSCA-CNN, CSAGC-IDS is constructed. The experimental results indicate that the model surpasses other comparative methods, demonstrating effectiveness and progressiveness in network intrusion detection tasks with high-dimensional, complex, and imbalanced traffic data.

1.3 Paper Structure

Section 2 specifically introduces the relevant work in this field. Section 3 details the proposed intrusion detection model, CSAGC-IDS, and its two integral components: SC-CGAN and CSCA-CNN. Section 4 demonstrates the evaluation. It compares the performance of the proposed algorithms and model with existing methods, while also conducting ablation experiments on CSCA-CNN to further analyze its effectiveness. Section 5 concludes the paper with a summary and provides insights into potential directions for future improvements.

2 Related Work

2.1 Deep Learning Intrusion Detection Methods

Gupta et al. proposed CSE-IDS by combining cost sensitive deep learning and ensemble learning and achieved good performance on imbalanced data [6]. Li et al. combined multiple CNNs [8] to achieve better accuracy and low complexity [9]. Shams et al. proposed CAFE-CNN, which converts traffic data into grayscale images and extracts context aware features [10]. Fu et al. combined CNN and bidirectional LSTM to enhance detection performance [11]. Cui et al. combined CNN and LSTM [12] to form a traffic classifier after extracting features from stacked autoencoder (SAE), fully considering the correlation between data and exhibiting good performance [7].

2.2 Imbalanced Data Processing Methods

Synthetic Minority Oversampling (SMOTE) [13] can synthesize new minority samples to achieve relative class balance. Jiang et al. used SMOTE to address the data imbalance in network intrusion detection [14]. Ma et al. combined adversarial reinforcement learning with SMOTE for network intrusion detection [15]. Generative Adversarial Network (GAN) is a generative model proposed by Goodfellow [16]. Lee et al. oversampled the minority class of network intrusion data using GAN which performed better than SMOTE [17]. Douzas et al. used CGAN [18] to handle imbalanced data, which performed better than other methods [19]. Cui et al. used WGAN [20] combined with GMM for network intrusion detection data balancing, achieving significant performance improvement [7].

3 Proposed Model for Network Intrusion Detection

3.1 CSAGC-IDS Architecture

CSAGC-IDS consists of two sub module algorithms, SC-CGAN and CSCA-CNN. The former is used for traffic data balancing to reduce imbalance, while the latter classifies and detects traffic.

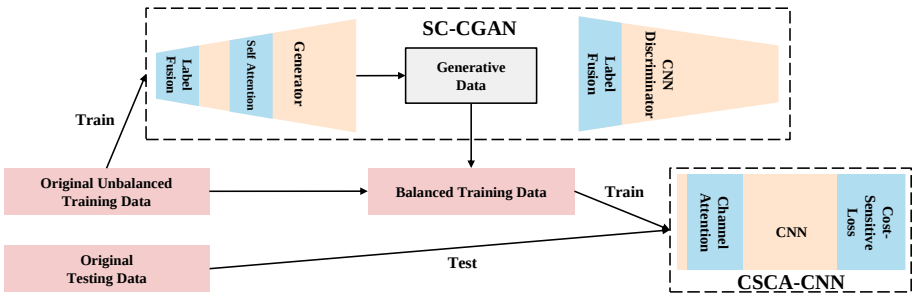

Fig. 1. CSAGC-IDS architecture

The overall architecture is demonstrated in Fig. 1. SC-CGAN uses the original training set to generate new data similar to the original data, and forms a class balanced data with the original training set to train CSCA-CNN. After training, CSCA-CNN was tested to obtain the final detection result. The model operation process is demonstrated in Fig. 2.

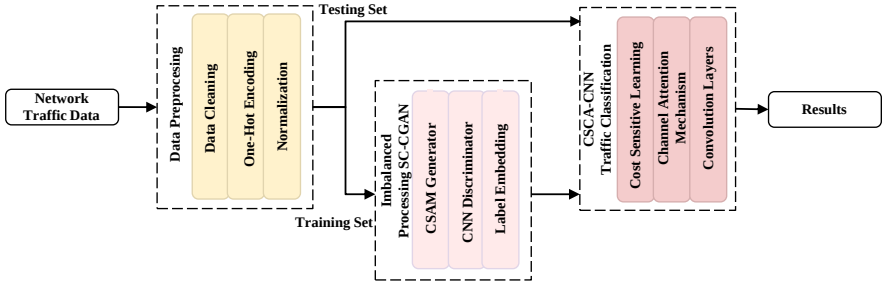

Fig. 2. CSAGC-IDS process

Data preprocessing is an important initial step. Numerical processing transforms features into One Hot Encoding that is easily accepted by the model. That only preserves category difference information to avoid misleading the model, allowing the model to better understand the features.Normalization transforms feature values to a certain range, such as [0,1], eliminating the influence of different ranges of feature values. Normalization can make parameter updates more stable and converge faster. Standardization is a kind of normalization:

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

It converts the original data into a distribution with a mean of 0 and a standard deviation of 1, while retaining the original characteristics of data.

The remaining steps will be introduced below.

3.2 Imbalanced Data Processing Algorithm SC-CGAN

SC-CGAN (Self Attention Mechanism Convolution Conditional Generative Adversarial Network) is a generator that integrates a self attention mechanism module [21] on the basis of a regular Conditional GAN [18] , and the discriminator uses CNN [8] to distinguish true or false. The generator, discriminator, and self attention module integrate conditional information into their input, namely the traffic data categorical labels.

SC-CGAN is employed to generate high-quality traffic data, with the objective of balancing the training set, augmenting samples from minority classes, and mitigating model bias stemming from data imbalance. The evaluation results have demonstrated that SC-CGAN exhibits significant advantages over existing methods in the generation of high-quality network traffic data. The architecture of SC-CGAN is demonstrated in Fig. 3.

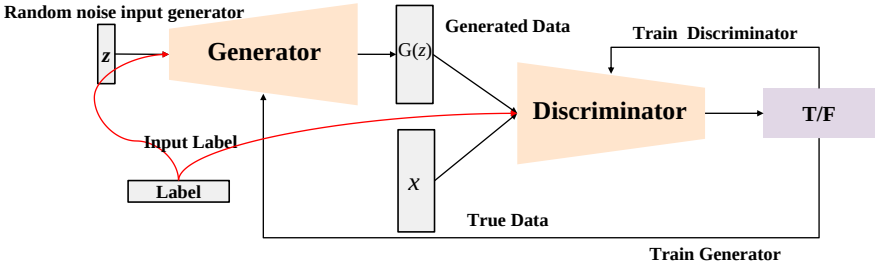

Fig. 3. SC-CGAN architecture

Generative Adversarial Nets with Fusion of Conditional Information. While GANs exhibit remarkable generative capabilities, their sole reliance on noise input falls short when dealing with multi-category training data, as it lacks the ability to control the generation of specific categories. To address this limitation, Conditional Generative Adversarial Networks (CGANs) [18] introduce additional conditional information into both the generator and the discriminator.

SC-CGAN adopts this approach for the generation of traffic data, where the generator integrates conditional information with random noise, and the discriminator combines conditional information with the input sample to be evaluated. Both the generator and discriminator incorporate this conditional information in their respective tasks of generating and discriminating. Specifically, the conditional information in SC-CGAN takes the form of one hot encoded category labels.The loss function of SC-CGAN is defined as follows:

$$\mathcal{L}_D = -\mathbb{E}_{x,y\sim p_{\text{data}}(x,y)}[\log D(x,y)] - \mathbb{E}_{z\sim p_z(z),y\sim p_{\text{data}}(y)}[\log(1 - D(G(z,y),y))] \tag{2}$$

$$\mathcal{L}_G = -\mathbb{E}_{z\sim p_z(z),y\sim p_{\text{data}}(y)}[\log D(G(z,y),y)] \tag{3}$$

The generator's loss minimization objective is to produce data that the discriminator deems as authentic (i.e., with an output close to 1), whereas the discriminator aims to minimize its loss by accurately distinguishing between generated data (outputting 0) and real data (outputting 1). Both the generator and discriminator incorporate the conditional information, y, during this process.

Utilizing the category information, the SC-CGAN generator produces samples of specified classes. By generating additional samples from minority classes, it aims to mitigate the imbalance present in the original dataset.

Conditional Self Attention Mechanism Generator. The SC-CGAN generator is integrated with the Conditional Self Attention Mechanism (CSAM). Transformer [21] represents a significant milestone in the realm of artificial intelligence, and the SAM serves as its cornerstone. Remarkably, the SAM has not only been implemented in the realm of Natural Language Processing (NLP) [22],

but it has also achieved significant success in the domain of image generation [23].

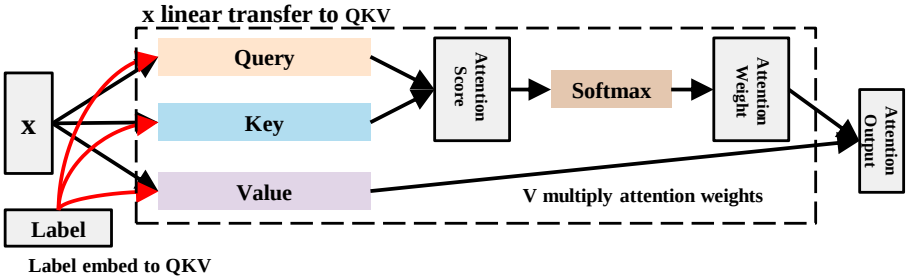

Fig. 4. CSAM architecture

The integration of CSAM in SC-CGAN generators is beneficial for generating higher quality traffic data. Fig. 4 demonstrates the CSAM architecture in the SC-CGAN generator, where Query (Q), Key (K), and Value (V) are all obtained through linear transformation of the input. Query retrieves and queries relevant information in the traffic feature sequence. Key is used for similarity matching with Query, while Value is associated with Key. Conditional information is embedded into Query, Key, and Value, and a dot product of Query and Key is computed to determine the similarity between traffic data features. Following a Softmax operation, the attention weight P is obtained and applied to Value for attention-weighted scaling, ultimately yielding the output. The calculation process of the SAM is outlined below:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \tag{4}$$

By incorporating a residual connection for this module, can mitigate the issue of vanishing gradients in the model [24], as illustrated in Fig. 5. Additionally, this residual connection ensures that any raw input information that may be lost during the flow through the CSAM is preserved.

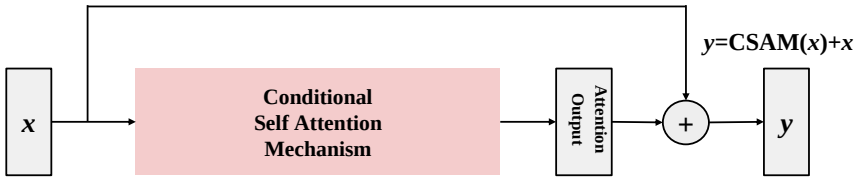

Fig. 5. Residual connection of CSAM

CSAM contributes to the generation of high-quality traffic data in the following significant ways:

Capture Long-Range Dependencies. CSAM can effectively captures the dependency and correlation relationships among traffic data features, regardless of how far they are in the sequence. Traffic data exhibits numerous dependencies, such as the association between protocol type and port number. When generating traffic data, consider dependency relationships and generate data that matches reality. In complex network scenarios, CSAM is used to adaptively learn dependency patterns.

Add Condition Information. CSAM embeds conditional information into the Q, K, and V. This approach enables more precise control over the generation of specific sample categories.

Enhance Model Learning Ability. Q, K, and V are obtained from learnable parameters. The model introduces more parameters to enhance learning ability.

3.3 Traffic Classification algorithm CSCA-CNN

The CSCA-CNN (Cost Sensitive Channel Attention Mechanism Convolutional Neural Network) framework effectively integrates Cost Sensitive Learning (CSL) [6] and Channel Attention Mechanism (CAM) [25] within a CNN architecture. CSL addresses the issue of bias towards majority classes, ensuring a more balanced treatment of all classes. On the other hand, CAM enhances the representation of crucial channel features, thereby boosting the overall performance of traffic classification. Fig. 6 illustrates the structure of the CSCA-CNN, showcasing how these two mechanisms are integrated.

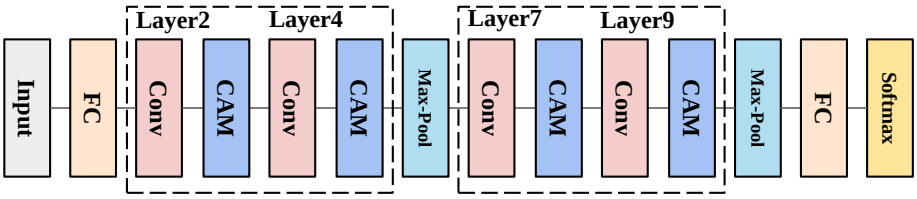

Fig. 6. CSCA-CNN architecture

Cost Sensitive learning. In ordinary classification tasks, it is often assumed that all misclassifications incur an equal cost, but in practical applications, misclassifying instances from different classes can lead to vastly disparate losses. To address this issue, Cost Sensitive Learning (CSL) [6] has been introduced, which assigns distinct weights to various types of errors and prioritizes the minimization of errors with higher weights during the training process.

CSCA-CNN utilizes CSL to modify the cross-entropy loss function. Specifically, a cost weight matrix is implemented to assign differential weights to the loss

functions corresponding to different categories. This approach imposes heavier penalties for misclassifying instances from minority classes, thereby increasing the model's focus on these classes during parameter updates. The cost-sensitive cross-entropy loss function is formulated as follows:

$$L = -\sum_{i=1}^{C} w_i \cdot y_i \cdot \log(p_i) \tag{5}$$

Channel Attention Mechanism. CSCA-CNN employs the Channel Attention Mechanism (CAM) feature extraction from the CBAM (Convolutional Block Attention Module) [25] framework for traffic classification. This approach aims to enhance the representation of effective and crucial channel features, while minimizing attention to redundant and irrelevant channel features, ultimately improving the overall classification performance. The specific process of CAM is illustrated in Fig. 7.

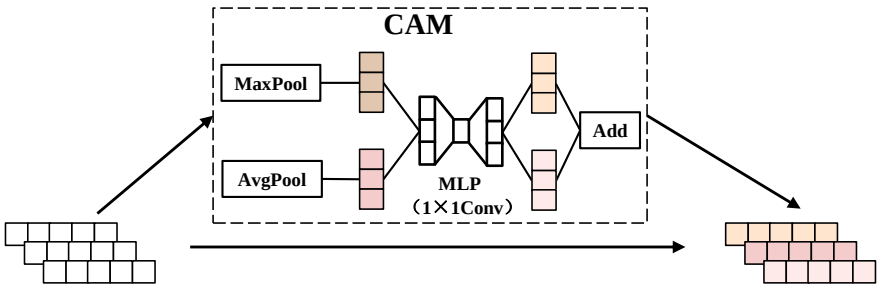

Fig. 7. CAM architecture

4 Evaluation

This section conduct extensive experiments to evaluate the proposed algorithms and model in terms of their data generation quality, classification performance, and model complexity. The results obtained validate the advantages of our proposed solution.

4.1 Experimental Configuration Environment

The experiments were conducted on a computing environment with Intel (R) Xeon (R) Gold 6240 CPU @ 2.60GHz. The GPU used was Tesla V100S-PCIE-32GB, and the operating system was Ubuntu 18.04.3 LTS. All code was implemented in Python 3.7.6. The framework was employed PyTorch 1.13.1+cu117.

4.2 Evaluation Metrics

Experiments utilize multiple classification performance indicators to provide a comprehensive evaluation, including Accuracy (Acc), Precision (Pre), Recall, and F1-score. Given the significant imbalance in the data, relying solely on Accuracy as a metric is insufficient. Therefore, I also include precision, recall, and F1-score to obtain a more thorough assessment. Notably, the F1-score is particularly valuable as it considers both precision and recall, rendering it a reliable and robust indicator [7].

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

In multi-class classification scenarios, it is crucial to consider the global performance across all classes. To achieve this, calculate various class of Pre, Recall, and F1-score separately, and use the ratio of each type of quantity as the weighted average. The F1-score calculation is as follows:

$$\text{Weighted F1} = \sum_{i=1}^{N} \left(\frac{\text{Number}_i}{\sum_{j=1}^{N} \text{Number}_j} \cdot 2 \cdot \frac{\text{Pre}_i \cdot \text{Recall}_i}{\text{Pre}_i + \text{Recall}_i} \right) \tag{7}$$

The indicators for measuring complexity are Params and FLOPs.

4.3 Dataset

Experiments utilize the NSL-KDD [26] benchmark dataset, a widely recognized resource in the field of network intrusion detection. This dataset provides comprehensive and authentic network intrusion traffic data, exhibiting a natural imbalance in data distribution as well as high-dimensional and complex features. These make NSL-KDD an excellent candidate for evaluating the effectiveness and robustness of intrusion detection models. All of the following evaluations were conducted on KDDTest+.

Table 1. NSL-KDD Description

Class	Description	Quantity	CI Ratio
Normal	Normal traffic (no attack)	77054	1
DoS	Denial-of-Service attack (Overloading to disrupt service)	53385	1.44
Probe	Probe attack (information gathering)	14077	5.47
R2L	Remote-to-Local attack (Unauthorized remote access)	3749	20.55
U2R	User-to-Root attack (attempt to gain superuser privileges)	252	305.77

4.4 Evaluation of Imbalanced Processing Algorithms

Comparative Experiments. To measure the quality of traffic data generation for various imbalanced processing algorithms, I evaluate the performance

of classifiers trained on data processed by these algorithms. This approach is justified as the performance of the classifier serves as a proxy for the quality of the training data [19]. Experiments evaluate the proposed SC-CGAN method and compare it with 8 other imbalance processing algorithms.

Table 2. Number of samples generated for each class

Data Source	Normal	DoS	Probe	R2L	U2R
Original Data	67343	45927	11656	995	52
Generated Data	0	21416	55687	66348	67291

The approach for balancing the experimental data involves generating additional samples for each category and integrating them into the original dataset. This process ensures that the number of samples in each category is equalized. In the case of the KDDTrain+ dataset, since the Normal class originally contained the highest number of samples at 67,343, I generated an equivalent number of samples for each other category to match this figure. The corresponding number of samples generated for each category is presented in the Table 2.

Table 3. SC-CGAN, CVAE, and CSCA-CNN hyperparameters

Hyperparameters	Generator	Discriminator	CVAE	CSCA-CNN
Hidden node	100	60	60	40
Noise dimension	123	-	-	-
Attention dimension	30	-	-	-
Activation	LeakyReLU	LeakyReLU	LeakyReLU	LeakyReLU
Initialization	He	Xavier	He	Xaiver
Batch size	128	128	128	128
Learning rate	0.001	0.000005	0.0001	0.01
Epoch	30	30	120	-
Optimizer	Adam [27]	Adam	Adam	Adam
Loss function	BCELoss	BCELoss	MSELoss	CSL-CELoss
Convolution kernel size	-	3	-	3
Dropout	-	0.3	-	0.3
Maxpool size	-	2	-	2
Latent dimension	-	-	32	-
Number of layers	8	8	10	12
CAM squeeze ratio	-	-	-	8

To conduct these comparisons, I implement 5 baseline classifiers and train them using the original imbalanced data, as well as balanced data processed by SC-CGAN and the aforementioned 8 algorithms. Subsequently, I test the classification performance of these trained classifiers to determine the effectiveness of each data balancing method. This comprehensive evaluation allows to gain in-

sights into the quality of data generated by different algorithms and their impact on classifier performance.

The evaluation results, presented in Tables 4 and 5, demonstrate that SC-CGAN exhibits noteworthy advantages across various classification performance indicators. These findings indicate the effectiveness of SC-CGAN in generating balanced training data that significantly enhances the performance of classifiers.

Table 4. Performance of different imbalanced processing algorithms (%)

Algorithm	CNN				Multilayer Perceptron			
	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1
Original Data	75.44	77.44	75.44	71.74	72.14	64.70	72.14	67.45
ROS	78.63	78.50	78.63	77.26	73.19	76.49	73.19	73.90
SMOTE [13]	79.40	80.58	79.40	78.27	72.37	75.93	72.37	73.64
Borderline SMOTE [28]	77.86	78.61	77.86	75.38	72.72	79.41	72.72	72.91
KMeans SMOTE [29]	77.45	77.67	77.45	76.01	69.54	76.76	69.54	71.74
SVM SMOTE [30]	79.62	80.46	79.62	77.75	75.26	77.78	75.26	75.06
CVAE [31, 32]	76.75	76.56	76.75	74.31	63.40	70.12	63.40	62.68
CBN-CVAE [33]	77.07	80.98	77.07	74.49	60.52	79.24	60.52	65.62
CGAN [18]	77.72	80.48	77.72	75.09	72.25	**82.15**	72.25	75.46
SC-CGAN	**80.96**	**83.04**	**80.96**	**78.78**	**78.28**	81.74	**78.28**	**78.81**

Table 5. Performance of different imbalanced processing algorithms(%)

Algorithm	Decision Tree				Random Forest				K-Nearest Neighbor			
	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1	Acc	Pre	Recall	F1
Original Data	75.88	79.32	75.88	72.74	77.07	80.81	77.07	73.64	72.82	72.61	72.82	67.98
ROS	77.02	79.14	77.02	73.84	76.54	81.50	76.54	73.00	74.71	78.55	74.71	71.70
SMOTE	76.11	78.04	76.11	73.33	75.89	80.53	75.89	72.99	75.42	78.92	75.42	73.29
Borderline SMOTE	75.89	78.73	75.89	73.59	76.63	79.91	76.63	73.59	74.94	78.41	74.94	72.08
KMeans SMOTE	76.13	79.03	76.13	72.81	76.29	79.07	76.29	72.56	75.25	79.22	75.25	72.76
SVM SMOTE	78.11	79.09	78.11	76.02	77.18	78.08	77.18	73.79	74.99	78.47	74.99	72.12
CVAE	78.45	79.10	78.45	77.18	77.36	81.71	77.36	74.07	77.87	80.07	77.87	74.85
CBN-CVAE	79.40	80.69	79.40	78.19	77.78	81.42	77.78	74.95	72.57	77.10	72.57	69.52
CGAN	79.50	80.85	79.50	76.80	77.36	81.48	77.36	75.09	78.43	81.00	78.43	75.97
SC-CGAN	**80.19**	**82.18**	**80.19**	**79.01**	**78.80**	**82.93**	**78.80**	**75.85**	**79.37**	**81.29**	**79.37**	**76.85**

Dimensionality Reduction for Visualization. The process entails diminishing the complexity of high-dimensional traffic data by employing techniques such as t-Distributed Stochastic Neighbor Embedding (t-SNE) [34] and Stacked Autoencoder (SAE) [7], ultimately projecting the data into a two-dimensional space for intuitive visualization through scatter plots.

The comparative visualization of the original imbalanced data and the SC-CGAN-balanced data, achieved through the utilization of t-SNE (depicted on

the left) and SAE (displayed on the right) dimensionality reduction algorithms, is presented in Fig. 8.

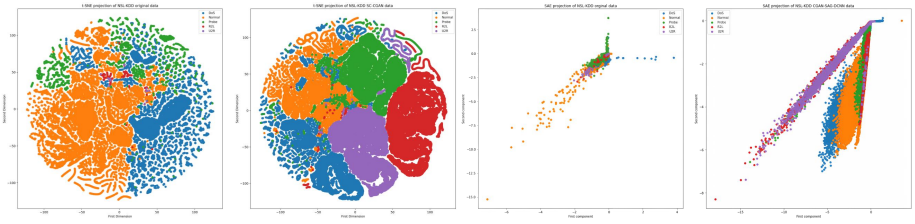

Fig. 8. NSL-KDD original imbalanced data and SC-CGAN balanced data dimensionality reduction visualization by t-SNE and SAE

Observing the results, it is evident that the balanced data generated by SC-CGAN notably augments the presence of samples from rare classes compared to the original data, thus effectively mitigating data imbalance. Consequently, the decision boundaries between various classes become more distinct, favoring the classification task. Furthermore, the data augmentation achieved through SC-CGAN widens the data coverage, potentially enhancing the model's generalization capabilities.

4.5 Evaluation of Traffic Classification Algorithms

All classification algorithms are trained on the balanced data generated by SC-CGAN. To assess the effectiveness of the CSL and CAM components, ablation experiments are conducted. Following this, a comparative analysis is performed between the baseline classification algorithm and the proposed CSCA-CNN. Finally, a comparative evaluation of the algorithm complexity of CSCA-CNN is undertaken with respect to the complexity reported in other relevant studies.

Table 6. Ablation experiment results for CSCA-CNN(%)

Algorithm	CSL	CAM	CNN	Acc	Pre	Recall	F1
CSCA-CNN	✓	✓	✓	**84.55**	**85.70**	**84.55**	**84.52**
CNN-Only	-	-	✓	80.96	83.04	80.96	78.78
w/o CAM	✓	-	✓	82.66	83.37	82.66	82.30
w/o CSL	-	✓	✓	81.72	83.41	81.72	79.60

Table 7. Comparative experiment results between CSCA-CNN and baseline classifiers(%)

Classifier	Acc	Pre	Recall	F1
Naive Bayes	53.80	48.44	53.80	44.25
Logistic Regression	77.04	79.68	77.04	73.67
K-Nearest Neighbor	79.37	81.29	79.37	76.85
Decision Tree	80.19	82.18	80.19	79.01
Random Forest	78.80	82.93	78.80	75.85
XGBoost [35]	78.94	81.31	78.94	76.73
Multilayer Perceptron	78.28	81.74	78.28	78.81
CSCA-CNN	**84.55**	**85.70**	**84.55**	**84.52**

Table 8. Comparative experiment results on complexity of CSCA-CNN

Indicator	CNN	1D-CNN	DNN 2 layers	DNN 3 layers	DNN 4 layers	DNN 5 layers	**CSCA-CNN**
Cite	[36]	[37]	[37]	[37]	[37]	[37]	-
Params	126826	90373	841221	1235717	1366789	1399557	**49469**
FLOPs	-	6886280	1680670	2469150	2731038	2796446	**729000**

The ablation study unequivocally demonstrated the effectiveness of the proposed enhancement. In the realm of traffic classification, the CSCA-CNN stands out, exhibiting considerable superiority compared to conventional baseline classifiers. Notably, the CSCA-CNN boasts lower Params and FLOPs, translating to reduced storage requirements and a diminished risk of overfitting. Furthermore, its efficient design ensures it necessitates less computational resources, making it a cost-effective and efficient solution for traffic classification tasks.

4.6 Evaluation of Intrusion Detection Models

In evaluating the performance of the CSAGC-IDS, a comprehensive comparison is conducted with both classical models and the start-of-the-art artificial intelligence models that have been proposed by researchers in recent years. Initially, I analyze the binary classification capabilities of the CSAGC-IDS, distinguishing between normal traffic and attack patterns. Subsequently, I delve deeper into comparing the performance of the five-class classification models, which categorize traffic into five distinct classes: Normal, DoS, Probe, R2L, and U2R.

Binary Classification. To evaluate the binary classification performance of the CSAGC-IDS model, a comparison has been conducted against various benchmark models, including LR, NB, SVM-rbf, DNN 1 layer, DNN 5 layers [38], Multi-CNN [9], and DLNID [11].

Table 9. Models binary classification performance comparative results(%)

Model	Acc	Pre	Recall	F1
LR [38]	82.60	91.50	74.40	82.00
NB [38]	82.90	86.50	80.50	83.40
SVM-rbf [38]	83.70	76.90	**99.30**	86.70
DNN 1 layer [38]	80.10	69.20	96.90	80.70
DNN 5 layers [38]	78.90	68.00	96.30	79.70
Multi-CNN [9]	86.95	89.56	87.25	88.41
DLNID [11]	90.73	86.38	93.17	89.65
CSAGC-IDS	**91.09**	**93.68**	90.45	**92.04**

Five-Class Classification. For Siam-IDS [40], I-SiamIDS [41], and LIO-IDS [42], where only various classes of Pre, Recall, and F1-score are provided, I utilized weighted approach to calculate overall indicators for comparison.

Table 10. Models five classification performance comparative results(%)

Model	Acc	Pre	Recall	F1
J48 [26]	81.05	-	-	-
NBTree [26]	82.02	-	-	-
RandomTree [26]	81.59	-	-	-
SVM [26]	69.52	-	-	-
AlexNet [14]	77.02	78.54	77.24	77.88
LeNet-5 [14]	79.91	82.95	80.01	80.45
BiLSTM [14]	79.43	81.14	79.65	80.39
DNN 5 layers [38]	78.50	81.00	78.50	76.50
CNN [36]	80.13	-	-	-
Multi-CNN [9]	81.33	-	-	-
CAFE-CNN [10]	83.34	85.35	83.44	82.60
SCAD-RNN [39]	82.61	-	-	-
Siam-IDS [40]	-	77.39	77.41	75.65
I-SiamIDS [41]	-	78.77	80.32	78.81
LIO-IDS [42]	-	81.13	80.80	80.77
DQN [43]	81.80	-	-	-
SSDDQN [44]	79.43	82.81	79.43	76.22
AE-RL [45]	80.16	79.74	80.16	79.40
AESMOTE [15]	82.09	84.11	82.09	82.43
AE-SAC [37]	84.15	84.27	84.15	83.97
CSAGC-IDS	**84.55**	**85.70**	**84.55**	**84.52**

Based on the results presented in Tables 9 and 10, the CSAGC-IDS demonstrates exceptional performance, highlighting its progressiveness and effectiveness. The CSAGC-IDS excels in learning deep feature representations of data, making it adept at managing complex, high-dimensional, and imbalanced data compared to other deep neural network architectures.

4.7 Interpretability Analysis

While CSAGC-IDS exhibits remarkable performance, its deep learning structure and vast parameters hinder interpretability [46], a crucial aspect in network intrusion detection. Given the high error costs, administrators often require more than just labels to make informed decisions [47].

To enhance interpretability, I utilize LIME [48] and SHAP [49] on the NSL-KDD to dissect CSAGC-IDS to provide with insights into the model's decision-making process, fostering trust and enabling detection of potential errors.

LIME. LIME (Local Interpretable Model-agnostic Explanations) trains interpretable models to approximate complex decision boundaries on individual samples, providing localized explanations [48].

For the binary classification task of CSACG-IDS on the NSL-KDD, LIME employs a linear model to mimic the CSACG-IDS. The coefficients of this interpretable model quantify each feature's impact on predictions. I select 4 normal samples and generate 50,000 perturbed samples for analysis.

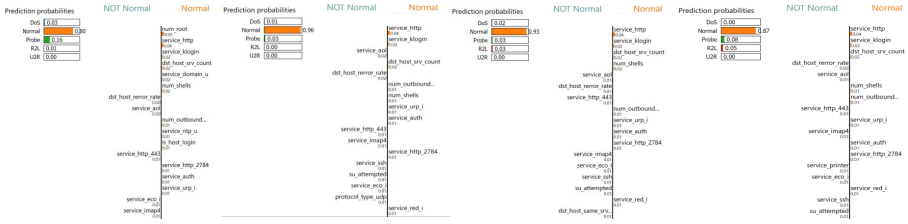

Fig. 9. 4 normal samples interpretability analysis by LIME

As illustrated in Fig. 9, by integrating the analysis of 4 normal samples, it is evident that features such as service_http, dst_host_srv_count, service_klogin, service_urp_i and num_shells have a positive effect on predicting normal classes in CSACG-IDS. Conversely, features such as service_aol, service_eco_i, service_imap4 and service_ssh have a negative effect. The service_http represents common HTTP services that appear frequently in the Internet. It may indicate behaviors such as DoS. The dst_host_srv_count reflects the usage of services on the host, which may reveal attempts at port scanning and exploiting vulnerabilities in multiple services. The feature service_klogin indicates Kerberos login, which may suggest legitimate user authentication behavior. The num_shells may reveal that malicious users or scripts are attempting to control the system, which could be related to U2R and R2L attacks. The service_ssh represents the SSH service used for remote management, which may reveal the existence of unauthorized remote access and control (U2R, R2L).

SHAP. SHAP (SHapley Additive exPlanations) assesses each feature's contribution, quantifying its average marginal impact on the model's outcome via the Shapley value, offering a quantitative analysis for each feature [49].

SHAP offers a binary interpretation for CSAGC-IDS, visualizing the Shapley values of each feature in a force plot (Fig. 10). Here, the Shapley values are depicted as forces, indicating their impact on the results, with red indicating positive contributions and blue representing negative effects. In prediction of the attack sample, features such as service_http have a positive effect on predicting attack class and features such as flag_REJ have a negative effect.

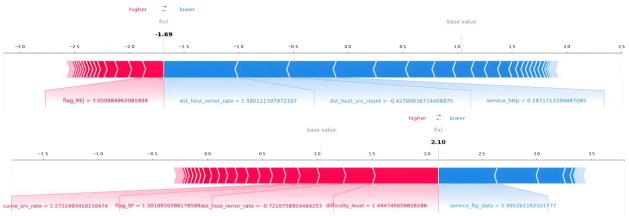

Fig. 10. Attack and normal samples SHAP force plot

By taking 100 samples and plotting force plot in horizontal stack. As illustrated in Fig. 11, Sample 56 is classified as an attack sample, based on the values of features including service_http, dst_host_rerror_rate and service_imap4. The service_imap4 may reveal behaviors that exploit the IMAP4 mail service for attack. The dst_host_rerror_rate indicates the rate of connection errors on the destination host, which could suggest that the host is under substantial invalid or malicious requests (DoS).

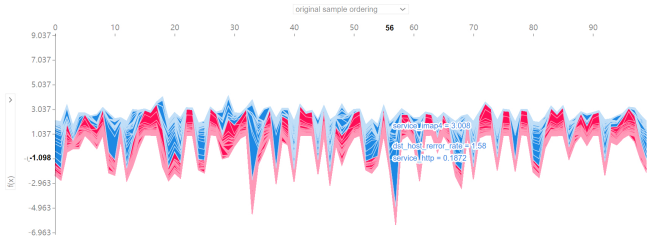

Fig. 11. 100 samples SHAP force plot

5 Conclusion

CSAGC-IDS, a deep learning network intrusion detection model that leverages cost sensitive learning and a mixed attention mechanism to tackle the challenges of high-dimensional, complex, and imbalanced data distributions in network intrusion detection. Experimental results demonstrate its effectiveness in achieving superior performance for these issues.

CSAGC-IDS includes two algorithms. SC-CGAN integrates CGAN with CSAM and CNN to fuse conditional information, capture feature dependencies, generate high-quality data. CSCA-CNN for traffic classification, which integrates CAM and CSL to extract deep features from complex and high-dimensional data, assign higher costs to minority classes to reduce bias caused by data imbalance.

Finally, enhancing interpretability of the model provided explanations for the decision-making processes.

Based on this paper, there are several prospective directions for future work. Firstly, enhancing robustness [43, 50]. Secondly, reducing parameter and computational complexity [51, 52]. Thirdly, considering temporal characteristics of network traffic [47].

Acknowledgments. I would like to express my sincere gratitude to my girl friend, Gui Tian.

References

1. Khraisat, A., Gondal, I., Vamplew, P., et al. Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, **2**(1), 22 (2019)
2. Yuan, L., Mai, J., Su, Z., et al. FIREMAN: a toolkit for firewall modeling and analysis. In: Proceedings of the 2006 IEEE Symposium on Security and Privacy. IEEE (2006)
3. LeCun, Y., Bengio, Y., Hinton, G. Deep learning. Nature, **521**(7553), 436–444 (2015)
4. Javaid, A., Niyaz, Q., Sun, W., et al. A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies. pp. 21–26 (2016)
5. He, H., Garcia, E. A. Learning from imbalanced data. IEEE Transactions on Knowledge & Data Engineering, **20**(9), 1263–1284 (2008)
6. Gupta, N., Jindal, V., Bedi, P. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Computers & Security, **112**, 102499 (2022)
7. Cui, J., Zong, L., Xie, J., et al. A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data. Applied Intelligence, **53**(1), 272–288 (2023)
8. LeCun, Y., Bottou, L., Bengio, Y., et al. Gradient-Based Learning Applied to Document Recognition. Proceedings of the IEEE, **86**(11), 2278–2324 (1998)
9. Li, Y., Xu, Y., Liu, Z., et al. Robust Detection for Network Intrusion of Industrial IoT Based on Multi-CNN Fusion. Measurement, **154**(2), 107450 (2019)
10. 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, **33**(18), 13647–13665 (2021)
11. Fu, Y., Du, Y., Cao, Z., et al. A deep learning model for network intrusion detection with imbalanced data. Electronics, **11**(6), 898 (2022)
12. Hochreiter, S., Schmidhuber, J. Long short-term memory. Neural Computation, **9**(8), 1735–1780 (1997)
13. Chawla, N. V., Bowyer, K. W., Hall, L. O., et al. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, **16**(1), 321–357 (2002)
14. Jiang, K., Wang, W., Wang, A., et al. Network Intrusion Detection Combined Hybrid Sampling With Deep Hierarchical Network. IEEE Access, **8**, 32464–32476 (2020)
15. Ma, X., Shi, W. AESMOTE: Adversarial Reinforcement Learning With SMOTE for Anomaly Detection. IEEE Transactions on Network Science and Engineering, **8**(2), 943–956 (2021)

16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al. Generative Adversarial Nets. In: Advances in Neural Information Processing Systems 27 (NIPS 2014), Montreal, Canada (2014)
17. Lee, J. H., Park, K. H. GAN-based imbalanced data intrusion detection system. Personal and Ubiquitous Computing, **25**(1), 121–128 (2021)
18. Mirza, M., Osindero, S. Conditional Generative Adversarial Nets. arXiv preprint arXiv:1411.1784 (2014)
19. Douzas, G., Bacao, F. Effective data generation for imbalanced learning using Conditional Generative Adversarial Networks. Expert Systems with Applications, **82**, 74–86 (2017)
20. Arjovsky, M., Chintala, S., Bottou, L. Wasserstein generative adversarial networks. In: Proceedings of the 34th International Conference on Machine Learning (ICML 2017), Sydney, Australia (2017)
21. Vaswani, A., Shazeer, N., Parmar, N., et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017)
22. Brown, T. B., Mann, B., Ryder, N., et al. Language models are few-shot learners. In: Proceedings of the NeurIPS (2020)
23. Zhang, H., Goodfellow, I., Metaxas, D., et al. Self-attention generative adversarial networks. In: Proceedings of the 36th International Conference on Machine Learning (2019)
24. He, K., Zhang, X., Ren, S., et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
25. Woo, S., Park, J., Lee, J. Y., et al. CBAM: Convolutional Block Attention Module. In: Proceedings of the 15th European Conference on Computer Vision (2018)
26. Tavallaee, M., Bagheri, E., Lu, W., et al. 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, pp. 1-6 (2009)
27. Kingma, D. P., Ba, J. Adam: A Method for Stochastic Optimization. arxiv preprint arxiv:1412.6980 (2014)
28. Han, H., Wang, W. Y., Mao, B. H. Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Proceedings of the International Conference on Intelligent Computing (2005)
29. Douzas, G., Bacao, F., Last, F. Improving Imbalanced Learning Through a Heuristic Oversampling Method Based on K-means and SMOTE. Information Sciences, **465**, 1-20 (2018)
30. Nguyen, H. M., Cooper, E. W., Kamei, K. Borderline Over-sampling for Imbalanced Data Classification. International Journal of Knowledge Engineering and Soft Data Paradigms (2009)
31. Kingma, D. P., Welling, M., Auto-Encoding Variational Bayes. In: Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014). Banff, Canada, April 14-16 (2014)
32. Sohn, K., Yan, X., & Lee, H. (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In: Advances in Neural Information Processing Systems 28 (NIPS 2015). Montreal, Canada, December 7-12, pp. 3483-3491 (2015)
33. Yin, G. J., Liu, B., Sheng, L., et al. Semantics disentangling for text-to-image generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)

34. Laurens, V. D. M., Hinton, G. Visualizing Data using t-SNE. Journal of Machine Learning Research, **9**(2605): 2579-2605 (2008)
35. Chen, T., Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York (2016)
36. Ding, Y., Zhai, Y. Intrusion detection system for NSL-KDD dataset using convolutional neural networks. In: Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence. Shenzhen, China, pp. 81–85 (2018)
37. Li, Z. F., Huang, C. H., Deng, S. H., et al. A Soft Actor-Critic Reinforcement Learning Algorithm for Network Intrusion Detection. Computers & Security, **135**: 103502 (2023)
38. Vinayakumar, R., Alazab, M., Soman, K. P., et al. Deep learning approach for intelligent intrusion detection system. IEEE Access, **7**, 41525–41550 (2019)
39. Singh, P., et al. "Soft-computing-based false alarm reduction for hierarchical data of intrusion detection system." International Journal of Distributed Sensor Networks. **15**(10) (2019)
40. Bedi, P., Gupta, N., Jindal, V. Siam-IDS: Handling Class Imbalance Problem in Intrusion Detection Systems Using Siamese Neural Network. In: Proceedings of the Third International Conference on Computing and Network Communications, Trivandrum (2019)
41. Bedi, P., Gupta, N., Jindal, V. I-SiamIDS: An Improved Siam-IDS for Handling Class Imbalance in Network-Based Intrusion Detection Systems. Applied Intelligence, **51**(2): 1133-1151 (2021)
42. Gupta, N., Jindal, V., Bedi, P. LIO-IDS: Handling Class Imbalance Using LSTM and Improved One-vs-One Technique in Intrusion Detection System. Computer Networks,**192**: 108076 (2021)
43. Sethi, K., Rupesh, E. S., Kumar, R., et al. A Context-Aware Robust Intrusion Detection System: A Reinforcement Learning-Based Approach. International Journal of Information Security, **19**(6): 657-678 (2020)
44. Dong, S., Xia, Y., Peng, T. Network Abnormal Traffic Detection Model Based on Semi-Supervised Deep Reinforcement Learning. IEEE Transactions on Network and Service Management, **18**(4): 4197-4212 (2021)
45. Caminero, G., Lopez-Martin, M., Carro, B. Adversarial Environment Reinforcement Learning Algorithm for Intrusion Detection. Computer Networks, **159**: 96-109 (2019)
46. Adadi, A., Berrada, M. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). IEEE Access, **6**, 52138-52160 (2018)
47. Wei, F., Li, H.D., Zhao, Z.M., et al. xNIDM: Explaining Deep Learning-based Network Intrusion Detection Systems for Active Intrusion Responses. In: Proceedings of the 32nd USENIX Security Symposium (USENIX Security 2023). Anaheim, CA, USA (2023)
48. Ribeiro, M.T., Singh, S., Guestrin, C. "Why Should I Trust You?" Explaining the Predictions of Any Classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135-1144 (2016)
49. Lundberg, S.M., Lee, S.I. A Unified Approach to Interpreting Model Predictions. In: Advances in Neural Information Processing Systems 30 (2017)
50. Merzouk, M. A., Delas, J., Neal, C.,et al. Evading deep reinforcement learning-based network intrusion detection with adversarial attacks. In: Proceedings of the 17th International Conference on Availability, Reliability and Security, pp. 1-6 (2022)

51. Hinton, G., Vinyals, O., Dean, J. Distilling the knowledge in a neural network. arxiv preprint arxiv:1503.02531 (2015)
52. Yim, J., Joo, D., Bae, J., et al. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4133-4141 (2017)