Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

A multi-classification detection model for imbalanced data in NIDS based on reconstruction and feature matching

Yue Yang[1,3], Jieren Cheng[2,3*], Zhaowu Liu[2,3], Huimin Li[2] and Ganglou Xu[2]

Abstract

With the exponential growth of various data interactions on network systems, network intrusions are also increasing. The emergence of edge computing technology brings a new solution to network security. However, due to the difficulty of processing massive and unbalanced data at the edge, higher accuracy requirements are necessary for deployed detection models. This paper proposes a multi-classification model for network intrusion detection based on reconstruction and feature matching. This model can be deployed on small-scale edge nodes, effectively identifying various attack behaviors through the utilization of reconstruction errors and adaptive scaling. Furthermore, we proposed a model transfer method based on feature matching to enhance the training and detection efficiency of multi-classification models under different data distribution conditions. The proposed model has been evaluated on the CICIDS2017 dataset in terms of accuracy, recall, precision and F1 score. The model demonstrates high accuracy for normal flows in the network, majority class attacks, and minority class attacks, achieving an overall multi-class accuracy of 99.81%, outperforming similar models. Furthermore, this model demonstrates faster convergence and training speed after feature matching, exhibiting better robustness and outstanding performance at the edge.

Keywords Intrusion detection, Edge computing, Adaptive scaling, Feature matching

Introduction

With the rapid development of the Internet, network intrusion events occur frequently, and traditional network security solutions are usually concentrated on the data center or cloud services of enterprises, and these centralized security measures are difficult to meet the rapidly growing needs of network attacks and intrusions [1]. The edge computing technology distributes the intrusion detection model to the servers near the physical devices, routers or access points at the edge of the network, so as to realize real-time network security monitoring and threat prevention closer to the user terminal. This distributed security mechanism enables network intrusion detection to be carried out earlier and malicious behaviors can be located and isolated more precisely [2]. Therefore, edge computing brings a new solution to network security and provides a faster, more efficient and real-time security detection means.

Edge computing presents several advantages in the realm of network intrusion detection [3]. Firstly, deploying intrusion detection models at the edge offers lower latency, enabling real-time monitoring and faster response capabilities. This swift responsiveness is instrumental in minimizing losses incurred due to intrusions. Secondly, by decentralizing computing functions to the network edge, edge computing facilitates the realization of a more flexible and scalable network security architecture. This decentralization contributes to better

*Correspondence:
Jieren Cheng
cjr22@163.com
[1] School of Cyberspace Security, Hainan University, Haikou 570228, China
[2] School of Computer Science and Technology, Hainan University, Haikou 570228, China
[3] Hainan Blockchain Technology Engineering Research Center, Haikou 570228, China

management practices, enhancing the overall security posture. Lastly, edge computing brings forth greater machine learning and data analysis capabilities, enabling more accurate identification of new types of attacks compared to centralized solutions. In essence, the integration of edge computing enhances the efficiency and effectiveness of network intrusion detection systems.

Illustrated in Fig. 1, the potential for edge computing in network intrusion detection is expanding [4]. However, presently, intrusion detection encounters various risks and challenges within the realm of edge computing [5].

(a) At the periphery of the network environment, the computational and storage capacities of edge nodes are inherently limited, posing difficulties in effectively sampling extensive datasets [6]. Current intrusion detection technology faces challenges in extracting pertinent features from the resource-constrained edge nodes within vast and imbalanced datasets, thereby impeding accurate representation.
(b) Within the realm of edge computing, users have multiple avenues for transmitting network data. However, the openness of networks exposes edge nodes to a multitude of security threats. Advanced Persistent Threats (APT), challenging to detect due to their low sample size, prove elusive for existing intrusion detection techniques within massive and imbalanced datasets.
(c) When confronted with novel attacks, the efficiency of training detection models on edge nodes is suboptimal. This inefficiency hampers existing methods in swiftly and accurately identifying emerging types of attacks.

Aiming at the challenge of intrusion detection model in edge computing, this paper proposes a multi-class intrusion detection method based on reconstruction and feature matching. The major contributions of this paper are as follows:

(a) We propose an adaptive scaling method to handle massive and heterogeneous data collected at the edge, enhancing the discriminability of different features. This effectively addresses the challenging issue of data representation for models on edge devices.
(b) We introduce a deployable multi-class network intrusion model designed for the edge. This model leverages the fusion of reconstruction errors and extracted attack features to overcome the detection challenges of APT attacks in massive and imbalanced network flow data, which is a limitation observed in CNN-LSTM.
(c) We design a model transfer method based on feature matching. The edge model employs a feature matching transfer algorithm, facilitating the rapid construction of the target domain network. This effectively addresses the challenge encountered by edge models in promptly training and detecting novel attacks.

The paper is structured into several sections. In the second section, we briefly review existing attack detection

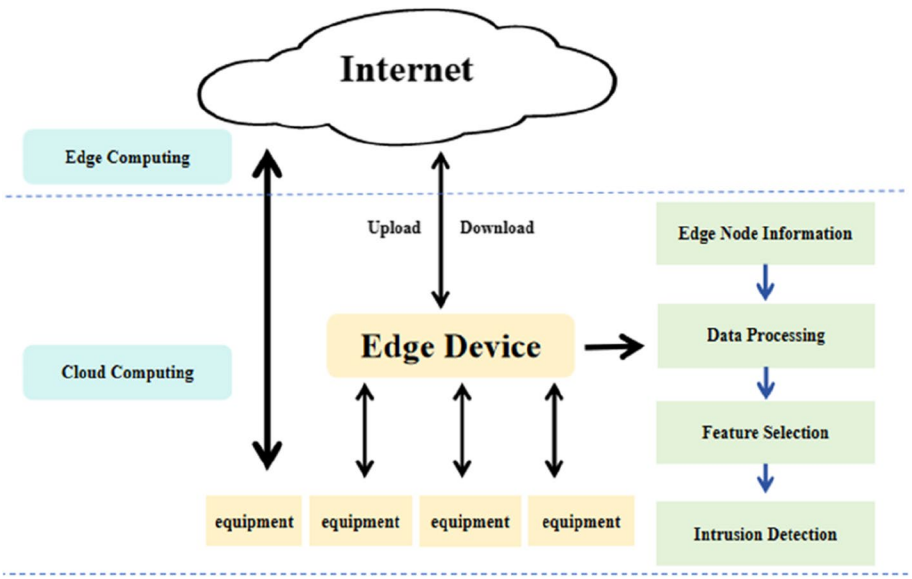


Fig. 1 Edge computing intrusion detection framework

models and transfer learning methods. The third section introduces the multi-classification model proposed in this paper and its specific algorithmic implementation. Our approach is evaluated using the CIC-IDS-2017, KDDCUP99, and UNSW-NB15 datasets in the fourth section. Lastly, the fifth section provides a summary of our research findings and outlines future directions for this work.

Related work

Network intrusion detection

Network intrusion detection leveraging edge computing entails implementing security measures and surveillance at the network periphery [7]. This strategy harnesses the potential of edge computing to locally process and analyze data, situated closer to the data source, rather than relying exclusively on centralized systems [8]. This methodology facilitates swift identification of potential intrusions, amplifies the efficacy of threat analysis, and furnishes a more agile defense mechanism against security threats.

In the realm of deep learning applications, researchers have explored various aspects of anomaly detection models, as summarized by Chalapathy et al. [9]. Furthermore, Yin et al. [10] introduced an enhanced mobile edge computing solution that combines federated learning with the CNN algorithm. Given the sequential nature of network traffic intrusion detection scenarios, many researchers have proposed models that amalgamate classical recurrent neural networks (RNN) and convolutional neural networks (CNN) to cater specifically to time series data. For instance, Xie et al. [11] developed a network intrusion detection algorithm employing dynamic IFS for data preprocessing and feature selection using the chi-square test on datasets such as KDD 99, NSL-KDD, and the high-dimensional UNSW-NB15. Liu et al. [12] introduced two attack detection methods, PL-CNN and PL-RNN, which support end-to-end attack detection of network data flows by considering the initial characters of the original effective load as input. Another noteworthy model, CANET [13], addressed the challenge of temporal information loss during high-level spatial feature extraction by simultaneously learning spatial-temporal features at multiple levels and considering the structural aspects of network attacks.

In the realm of anomaly detection, particularly when dealing with sample imbalance [14], Ngamba Thokchom et al. [15] proposed an ensemble learning-based classifier model incorporating Gaussian naive Bayes, logistic regression, and decision trees as fundamental classifiers. Additionally, a novel approach known as the Fence GAN model [16] was introduced to generate attack data using Generative Adversarial Networks (GANs). Traditional GANs posed a challenge as their loss function wasn't well-suited for anomaly detection since generated samples often overlapped with real data, rendering the generated discriminator ineffective for anomaly detection. To overcome this limitation, researchers modified the GAN loss function to position generated samples at the boundary of the real data distribution. This innovative approach enhanced abnormal data, expanded the feature distribution of anomaly samples, and ultimately boosted the model's anomaly detection capabilities.

The described method has enhanced the accuracy and efficiency of intrusion detection. Nevertheless, in multi-class scenarios with imbalanced data, especially when implemented in edge computing, it demonstrates restricted detection capabilities and resilience, particularly in the suboptimal detection of APT attacks.

Transfer learning

Deep neural networks exhibit significant autonomous feature learning capabilities, efficiently expediting model training effects, and are well-suited for updating models at the edge. Several experiments [17–19] have demonstrated that data features evolve from general to specific as learning progresses. However, as domain dissimilarities increase, the ability to transfer features diminishes notably at higher levels. Deep networks have emerged as the most conducive architecture for facilitating transfer learning. Transitioning from fine-tuning methods [17], to techniques that maintain the fixed network feature extraction layer to enhance the learnable distance from the classification layer [18, 20], to the concept of implicit distributed distance learning through domain antagonism [21–23], deep transfer learning methods have swiftly gained prominence in active research areas. The practical application of migration tasks typically involves heterogeneous networks, each requiring a potent migration approach. While deep neural networks excel at acquiring general features, training a deep neural network model for a specific task demands substantial datasets and, in some cases, even retraining for novel tasks, incurring considerable expenses. Therefore, employing transfer learning is a viable approach for training against novel attacks.

Huang et al. [24] introduced an unsupervised domain adaptation method centered around category contrast, significantly enhancing image classification performance. Meanwhile, Wang et al. [25] explored the use of sparse coding or joint graph learning to establish domain correspondences. Later, with the advent of generative adversarial networks, Yu et al. [26] proposed a domain adaptive network model that harnessed generative adversarial principles. These principles were applied to tackle domain adaptation challenges by utilizing the source and target domains within generative adversarial networks. However, given the dynamic and ever-evolving nature of network intrusion

attack scenarios, expecting all attack types to remain constant is unrealistic. This dynamism leads to a fundamental inconsistency in the feature and category spaces of the data. As a result, domain adaptation may not be the most suitable approach for the current application scenario.

In the realm of transfer learning, there is also substantial research focused on exploring feature matching. For instance, Srinivas et al. [27] manually performed inter layer matching for heterogeneous networks. However, this approach has some inherent limitations. The AT model research team [28] introduced two key concepts: the teacher network (source domain network) and the student network (target domain network), along with an attention mechanism. This attention mechanism involves superimposing image color channels to generate high-representation features, forming an attention mechanism. It assists the student network in enhancing its performance by transferring the attention from the teacher network, as mentioned in the article's AT-loss. In the FitNet [29] model, some researchers employ a norm criterion to facilitate feature matching between the source domain and the target domain. Meanwhile, the L2t-ww [30] model employs a meta-network approach to enable the migration from the target domain network to the source domain network. Hence, employing feature matching to accelerate the training of intrusion detection models at the edge is a practical approach, effectively mitigating the impact of emerging attacks.

Our methodology

Adaptive network intrusion data scaling based on edge computing

Normalization is the process of transforming data into a specific range or standard distribution to facilitate better comparison and analysis across different datasets or algorithms. This typically involves scaling the data to ensure its values fall within a particular range or distribution, such as adjusting data to follow a normal distribution with a mean of 0 and a standard deviation of 1 or scaling data to a range of 0 to 1. Normalization helps eliminate dimensional differences between different features, thereby enhancing the effectiveness of model training and data analysis.

In the attack data preprocessing of each edge node, we often use the method of min-max scaling to normalize features with varying numerical values. However, there is a significant disparity in the numerical representations of different features, which can lead to suboptimal normalization results [31]. presents a method for feature transformation. Due to the varying magnitudes of feature values, logarithmic transformation is applied to each feature value f to rescale it, with $u \leftarrow log(u+1)$.

However, this method is inadequate for scenarios in which the minimum feature value is negative. Removing features with negative values would lead to the loss of crucial attack-related features. Hence, we propose an adaptive scaling method based on edge computing,when the edge node collects network intrusion traffic, we scale the captured characteristics of attack flow adaptively The intrusion detection data U undergoes minimum value selection as U_{min}. When applying logarithmic transformation to the values, we incorporate an additional step, adding the absolute value of the minimum feature value $|U_{min}|$. This guarantees that each feature u can undergo compression while maintaining the integrity of the features.

$$u \in U \tag{1}$$

$$u' = \log(u + |U_{min}| + 1) \tag{2}$$

Reconstruction error and compression feature model based on depth autoencoder

In the context of large-scale network intrusions, the attack landscape is both diverse and complex. The data imbalance, primarily driven by the prevalence of APT attacks, poses a significant challenge. Models deployed at the network edge must demonstrate a robust capability for multi-class classification. This approach customizes the classification process to suit real-world network scenarios, thereby elevating the precision of detection models deployed at the edge for handling multi-class tasks.

This model comprises three main components: a feature extraction network, a self-coding compression network, and a multi-classification network. The specific structural diagram is depicted in Fig. 2.

The parameters of the deep autoencoder network are tuned based on normal samples. Leveraging the imbalance in the dataset, this model capitalizes on the fact that the reconstruction error for a few abnormal samples tends to be significantly larger than that of normal samples. Simultaneously, intrusion detection data exhibits robust differentiation in low-dimensional space. Consequently, this model employs deep autoencoders to compress one-dimensional features and utilize reconstruction errors as data features for training multiple network intrusion classifiers, thereby enhancing the detection rate for abnormal samples. This model comprises three main components: a feature extraction network, a self-coding compression network, and a multi-classification network. The specific structural diagram is depicted in Fig. 2.

The critical information from the input sample is preserved in the low-dimensional space, encompassing dimensionality reduction features identified through dimensionality reduction techniques, as well as the induced reconstruction errors. These reconstruction errors play a vital role in enhancing the intrusion detection classifier's detection rate. They are derived from the

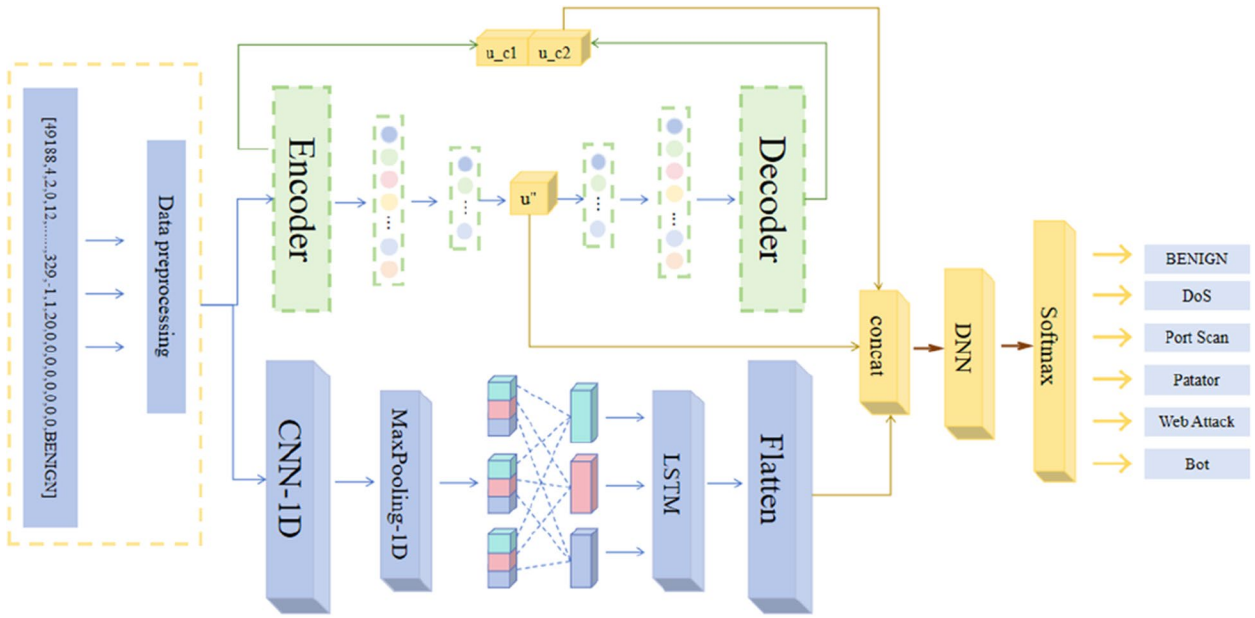


Fig. 2 Multi-classification model of network intrusion detection.u_{c_1}- Cosine similarity u_{c_2}-Euclidean distance. $u^{''}$-One-dimensional compression vector

compressed low-dimensional features and reconstruction errors generated by the deep autoencoder model. The reconstruction error is represented as a vector and can consist of multiple error measurement criteria.

This model uses the combined method of Euclidean distance u_{c_2} and cosine similarity u_{c_1} to calculate reconstruction error.

$$L_{reconstruct}(x, x') = ||x - x'||_2^2 \tag{3}$$

$$u^{''} = E(x; \theta_e) \tag{4}$$

$$x' = g\left(u^{''}; \theta_d\right) \tag{5}$$

Feature extraction and fusion network based on CNN-LSTM

The feature extraction network is mainly constructed for the purpose of extracting the features of high-dimensional network traffic intrusion data, which mainly uses one-dimensional convolutional network and LSTM.we can extract more efficient N-element local sequence feature abstracts. Based on the integration of CNN's local feature parallel extraction capability and LSTM's long-term temporal feature extraction, the time-space dimension features are completely extracted. CNN can extract local spatial or short-term structural relationships and capture local correlations of spatial or temporal structures. LSTM is specialized for time series modeling. For edge-based network intrusion detection, we have integrated CNN and LSTM to leverage their respective strengths.

Finally, we concatenate the abstract features extracted by the feature extractor with the error vector and pass them into the classifier. The role of classification network is to carry out multi-classification of network intrusion detection. The classification network uses the data of deep self-coding compression network and feature extraction network to complete multi-classification prediction, and uses SoftMax function to output multi-classification probability.

$$c = concat\left(u_{c_1}, u_{c_2}, u, H(x)\right) \tag{6}$$

$$y = Softmax(c) \tag{7}$$

Multi-classification detection model training

To circumvent local optimization issues, we employ an end-to-end training approach. The training objectives consist of the cross-entropy and reconstruction error functions of the classification network. In classification training, as the model's goal is to establish multiple classifications, we utilize a distinctive one-hot encoding method. This means that a K-category label is represented as a K-bit 01 one-hot vector. Such a label type represents the most desirable output result for the neural network. With N samples at hand, the training function of the model can be expressed as follows:

$$L_{total}(\theta|x,y) = \frac{1}{N}\sum_{i=1}^{N} L_{org}(\theta|x,y) + \frac{1}{N}\sum_{i=1}^{N} L_{reconstruct}(\theta|x,x') \tag{8}$$

The reconstruction error in this model primarily consists of the Euclidean distance criterion and cosine similarity criterion. Consequently, the final reconstruction error results from concatenating the values obtained from these two metrics to form a two-dimensional reconstruction error vector. In high-dimensional datasets with imbalanced data, the majority of anomalous samples exhibit distinct separation from normal samples in low-dimensional space. However, some abnormal samples may be concealed within the normal samples. Nevertheless, in high-dimensional space, this small number of abnormal samples differs significantly from normal samples. Therefore, this model combines reconstruction errors and compressed one-dimensional features within the deep autoencoder.

$$L_{cos}(x,x') = \frac{\sum_{i=1}^{n} x_i \times x_i'}{\sqrt{\sum_{i=1}^{n} x_i^2} \times \sqrt{\sum_{i=1}^{n} x_i'^2}} \tag{9}$$

$$L_d(x,x') = \sqrt{\sum_{i=1}^{n} (x_i - x_i')^2} \tag{10}$$

Input sample x_i (i = 1,2,3, ...) Data preprocessing is mainly to encode some discrete features and label the corresponding samples y_i (i =1,2, 3, ...). It's processed into a k-dimensional zero-one vector. Relu function was used for activation function and SoftMax was used for classification function. According to the description in Fig. 1, the corresponding end-to-end training scheme of the model was developed. First, the model propagates forward to get $\hat{y}$ (i =1,2,3, ...). Then according to the objective function set by the model, the backpropagation parameters are adjusted, ultimately completing the model's training.

Input: Datasets D_{train} = {(x, y)}, *Learning rate*α
Output Multiclass tag $\hat{y}$
Begin:
stage 1
1 Encoder$E(\cdot)$and Feature extractor$H(\cdot)$ enter a batch $B \subset D_{train}$
2 $H(\cdot)$extract local features and temporal features$H(x)$
3 $E(\cdot)$output compressed one-dimensional features u'
4 Suppress feature u input decoder$D(\cdot)$,output x'
5 Calculate Euclidean distanceu_{c_1} and cosine similarity u_{c_2},
stage 2
1 $concat(u_{c_1}, u_{c_2}, u'', H(x))$ as the input of $C(\cdot)$
2 $C(\cdot)$output y',use $L_{total}(\theta|x,y)$,back propagation updatesθ
Done

Algorithm 1 Network intrusion detection model based on edge computing

Intrusion detection multi-classification model based on feature matching

In the dynamic realm of evolving network environments, emerging attack types have become increasingly complex and diverse. Intrusion detection models deployed at the edge must swiftly adapt and learn from newly captured attack data. Traditional machine learning approaches, due to their extended training times, often fall short in providing a rapid response to these ever-changing attacks. To tackle this challenge, we introduce a feature matching algorithm tailored for attack traffic. This algorithm harnesses previously acquired knowledge from the source domain, expediting the learning efficiency for the target domain. As a result, it significantly accelerates the update speed of edge models, facilitating effective responses to emerging and novel attacks.

The multi-classification model for network intrusion detection, based on feature matching in the migration network, consists of two primary components: the source domain network, serving as an auxiliary component, and the target domain network, acting as the primary component. The classification network is responsible for categorizing network intrusion detection and classifying network traffic data into normal and abnormal types. The specific approach is depicted in the figure below.

As depicted in Formula 11 of the feature matching algorithm employed by this model, in a heterogeneous network where the source domain and target domain share precisely the same feature space, it becomes necessary to adopt suitable strategies for adaptation. In this regard, we have chosen to implement an objective function using the same approach as Fit Net [29].

$$L_{fm} = \left\| \varphi_\theta\left(T_{\theta_t}^n(I^t)\right) - S^m(I^s)\right\|_2^2 \tag{11}$$

Feature matching model training

We integrate the proposed feature matching algorithm with the original cross-entropy loss of the target domain network to facilitate the rapid convergence of the model. The classifier employs cross entropy, primarily utilized to assess the proximity between the actual output and the expected output. During the classification training, input samples x_i (i = 1,2,3, ...), data preprocessing is mainly to encode some discrete features and label the corresponding samples y_i (i =1,2,3, ...). It's processed into a k-dimensional zero-one vector. This type of label is the most desired output of the neural network. This model uses the label type to measure the difference between the output of the network and the multi-classification label, and uses the difference to update the network parameters through back propagation. θ_t refers to the model parameters of the target domain model, $S(\cdot)$ is the original source

domain model that needs to be loaded, and $T(\cdot)$ refers to the target domain model. When N samples are given, the training function of the model is as follows.

$$L_{target}\left(\theta_t|x_t,y_t\right)=\frac{1}{N}\sum_{i=1}^{N}L_{org}\left(\theta_t|x_t,y_t\right)+\frac{1}{N}\sum_{i=1}^{N}L_{reconstruct}\left(\theta_t|x_t,x_t{}'\right)+\frac{1}{N}\sum_{i=1}^{N}L_{fm} \tag{12}$$

The specific preprocessing and reconstruction algorithms of the core target domain model remain consistent with those detailed in the previous chapter. Input samples x_i (i = 1,2,3, ...), data preprocessing is mainly to encode some discrete features and label the corresponding samples y_i (i =1,2,3, ...). It's processed into a k-dimensional zero-one vector. Relu function was used for activation function and SoftMax was used for classification function. According to the description in Fig. 3, the corresponding end-to-end training scheme of the model was developed. The model propagates forward to get $\hat{y}_i$ (i =1,2,3, ...) Then according to the objective function set by the model, the back propagation parameters are adjusted.

Input: Datasets $D_{train}=\{(x_t,y_t)\}$, *Learning rate* α
Output: Multiclass tag$\hat{\boldsymbol{y}}$
Begin:
stage 1
1 Load source domain$S(\cdot)$
2 Target domain$T(\cdot)$ enter a batch $B \subset D_{train}$
3 $H(\cdot)$extract local features and temporal features$H(x)$
4 $E(\cdot)$output compressed one-dimensional features u
5 Suppress feature u input decoder$D(\cdot)$,output x'
6 Calculate Euclidean distanceu_{c_1} and cosine similarity u_{c_2}
7 Calculate source domain and target domain feature extractor L_{fm}
stage 2
1 $concat(u_{c_1},u_{c_2},u,H(x))$ as the input of $C(\cdot)$
2 $C(\cdot)$output y_t',use $L_{target}(\theta|x_t,y_t)$,back propagation updatesθ
Done

Algorithm 2 Multi-classification model of intrusion detection based on feature matching

Experimental results

Datasets description

To address the network environment challenges encountered when deploying intrusion detection models at the edge, this paper selected three datasets to validate the performance of the proposed algorithm and model. All chosen datasets include real-time data related to network traffic and exhibit the same imbalanced data characteristics as those captured in the real world.

In the multi-class intrusion detection task with imbalanced data, we choose the CIC-IDS-2017 dataset as the training and testing set for our model. This dataset contains several of the most common attack types encountered in real-life scenarios, and the distribution of attack data among different classes is imbalanced. In the feature matching algorithm, we have chosen the source domain dataset, KDD99, and utilized feature matching to transfer the multi-classification model to the target domain dataset, UNSW-NB15. These two datasets are entirely distinct. Regarding attack types, KDD99 lacks many modern attack types, while UNSW-NB15 encompasses some new types of attacks. This algorithm leverages knowledge from the source domain to enhance the detection capabilities of modern target domain network intrusion attack types.

CIC-IDS-2017

The CICIDS-2017 dataset was originally constructed by the Communication Security Establishment and the Canadian Institute for Cybersecurity [32] in 2017. This dataset comprises normal traffic as well as several of the most common types of attacks encountered in the real world, with each data entry containing 80 attributes. The dataset includes 2,273,097 samples of the normal class and 557,646 samples of attack classes, encompassing six attack categories: DoS, Port Scan, Infiltration, Web Attack, Patator, and Bot. However, due to the limited number of Infiltration attack samples, they are disregarded during the model's training and testing phases. Hence, this paper's model primarily focuses on detecting normal traffic and five categories of attack traffic.

KDD99

We employ the benchmark dataset KDDCUP99 [7] from the field of network intrusion detection as the source domain for model training and prediction. The experimental data is 7 weeks of network traffic data. The KDDCUP99 data set has a variety of attack types, which can be summarized as Normal, DoS, unauthorized access by remote host (R2L), and other types of attacks. Unauthorized local superuser privileged access (U2R) and port monitoring (Probing).

UNSW-NB15

As the target domain dataset for feature matching, the dataset [8] presents a diverse range of contemporary network traffic scenarios, encompassing numerous low-footprint intrusions and richly structured information. In the ever-evolving landscape of information technology, network attack methods have become increasingly diverse,

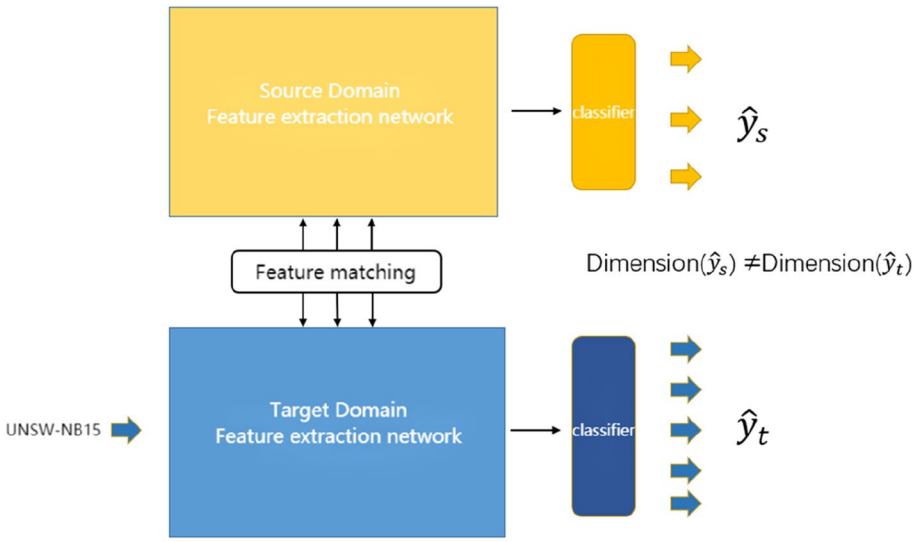


Fig. 3 Multi-classification model of intrusion detection based on feature matching

especially with the emergence of modern low-footprint attacks and novel attack techniques. Legacy datasets may not adequately capture these developments, and network intrusion detection continually requires the integration of new attack type detection methods. The dataset serves as a valuable experimental foundation for contemporary cyber-attack detection tasks. The dataset comprises a total of nine labels, represented by the "attack cat" field, which include categories such as Fussers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

Evaluation metrics

The four data used to evaluate the performance of a model are as follows: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The performance of the method is assessed using the following metrics, namely: 1) Accuracy, 2) Precision, 3) Recall, and 4) F1 Score.

$$Accuracy = \frac{TN + TP}{TN + FN + TP + FP} \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FP} \tag{15}$$

$$F1Score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}} \tag{16}$$

Experimental result

Intrusion detection multi-classification model

To validate the suitability of this approach for detecting imbalanced intrusion data collected at the edge, this method is compared to widely used deep learning algorithms for imbalanced sample detection such as DNN, CNN, and CNN-LSTM, as well as the latest multi-class intrusion detection algorithm, TLHA [33], in terms of precision, recall, and F1 score (Figs. 4, 5, 6, 7, 8 and Table 1).

The experimental results demonstrate that the method proposed in this paper exhibits better efficiency in the multi-class task of intrusion detection on imbalanced datasets. Specifically, the precision for all categories has significantly improved compared to other methods. In terms of F1 score and recall, except for the misclassification of some Bot traffic as normal traffic, the remaining metrics also outperform existing models (Table 2).

As shown in the table above, the method proposed in this paper consistently outperforms other methods in terms of the weighted average of the four metrics for the overall intrusion detection multi-class task. This demonstrates that the method presented in this paper is effective in successfully handling multi-class tasks.

Ablation study

For the multi-classification task of network intrusion detection at the edge, we propose three modules: Adaptive Scaling for data, Reconstruction Error-based Feature Compression using Deep Autoencoders, and Feature Extraction and Fusion model based on CNN-LSTM. Given that the

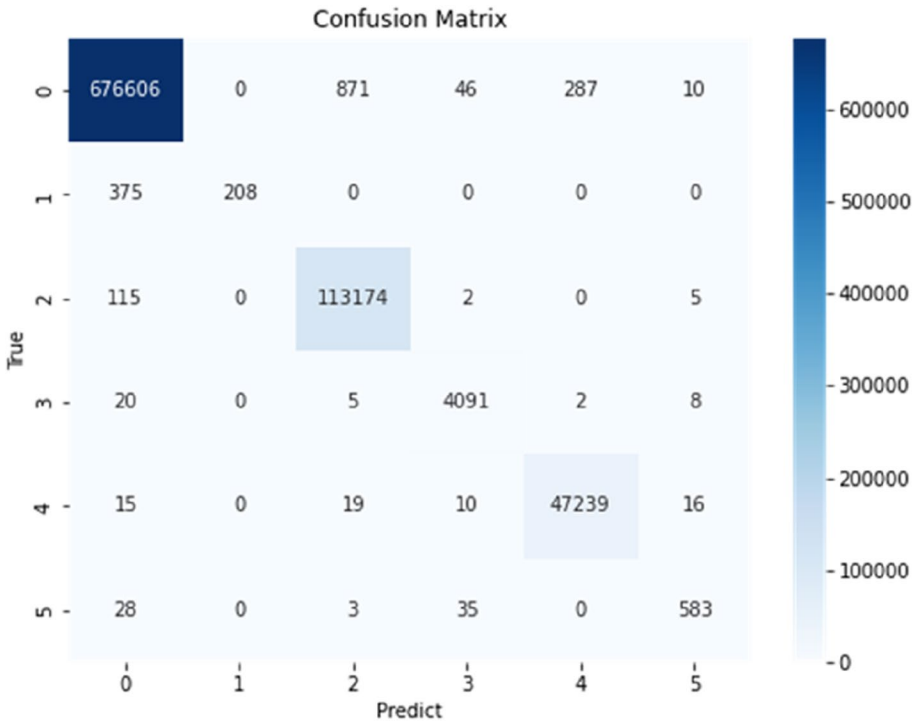


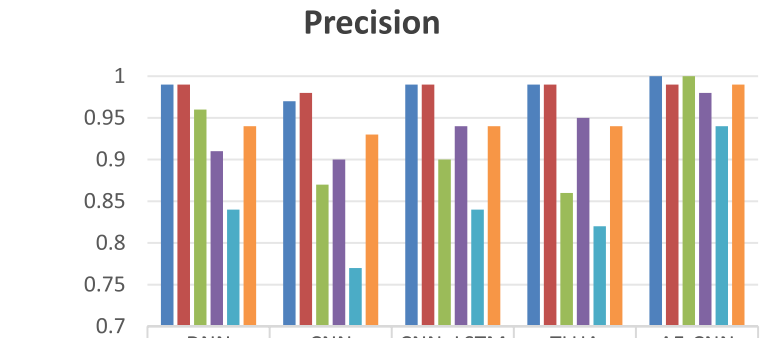
Fig. 4 Confusion matrix on CIC-IDS-2017

traffic collected at the edge is relatively smaller compared to the entire dataset, we compared the accuracy of these three modules using 10%, 20%, and 50% of the data (Table 3).

The experimental results indicate that the Adaptive Scaling module significantly enhances the data representation of the model on edge devices. Additionally, deep autoencoders exhibit notable discriminative capabilities, particularly for minority class samples, such as APT attacks.

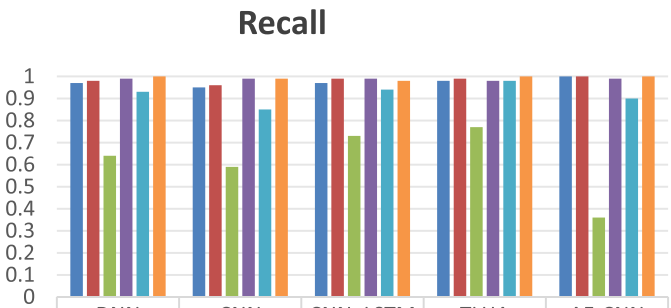
Multi-classification model of intrusion detection based on feature matching

In the experiment involving the migration of a multi-classification model for network intrusion detection based on feature matching, a five-classification model trained on the KDDCUP99 dataset is used as the source domain model, while the target domain model is based on the modern attack dataset UNSW-NB15 for migration. The



	DNN	CNN	CNN+LSTM	TLHA	AE-CNN
BENIGN	0.99	0.97	0.99	0.99	1
DoS	0.99	0.98	0.99	0.99	0.99
Bot	0.96	0.87	0.9	0.86	1
Patator	0.91	0.9	0.94	0.95	0.98
Web Attack	0.84	0.77	0.84	0.82	0.94
Port Scan	0.94	0.93	0.94	0.94	0.99

Fig. 5 The precision of different multi-class models



	DNN	CNN	CNN+LSTM	TLHA	AE-CNN
BENIGN	0.97	0.95	0.97	0.98	1
DoS	0.98	0.96	0.99	0.99	1
Bot	0.64	0.59	0.73	0.77	0.36
Patator	0.99	0.99	0.99	0.98	0.99
Web Attack	0.93	0.85	0.94	0.98	0.9
Port Scan	1	0.99	0.98	1	1

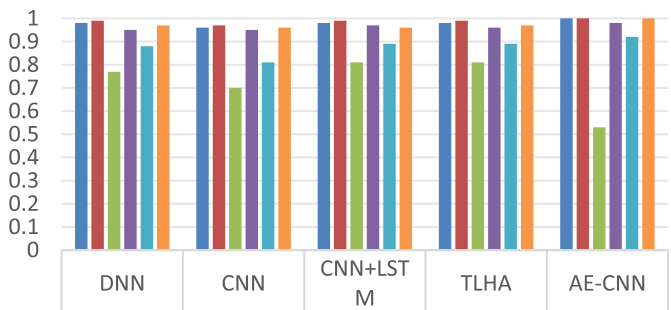
Fig. 6 The recall of different multi-class models

purpose of the study is to investigate whether the accuracy of the intrusion detection model is correlated with the proportion of abnormal data, the number of classifications, and potential relationships between certain categories. To explore these relationships, several experiments were designed.

Due to the limited data collected by edge devices, this experiment comprises four sets of distinct trials. Each set involves different quantities of intrusion detection data, simulating various proportions of traffic collected by edge devices. as outlined in the following table: The results correspond to the detection at data collection percentages of 5%, 10%, 20%, and 30%.

Concurrently, we assessed the learning outcomes of the model on the UNSW-NB15 dataset both before and after feature matching. After 5, 20, and 30 epochs, we evaluated the edge model's effectiveness in learning and training on the dataset.

Compared to the original model, the feature-matched model demonstrates a faster convergence rate and increased robustness. This further validates the effectiveness of the transfer algorithm. In networks with



	DNN	CNN	CNN+LSTM	TLHA	AE-CNN
BENIGN	0.98	0.96	0.98	0.98	1
DoS	0.99	0.97	0.99	0.99	1
Bot	0.77	0.7	0.81	0.81	0.53
Patator	0.95	0.95	0.97	0.96	0.98
Web Attack	0.88	0.81	0.89	0.89	0.92
Port Scan	0.97	0.96	0.96	0.97	1

Fig. 7 The F1-Score of different multi-class models

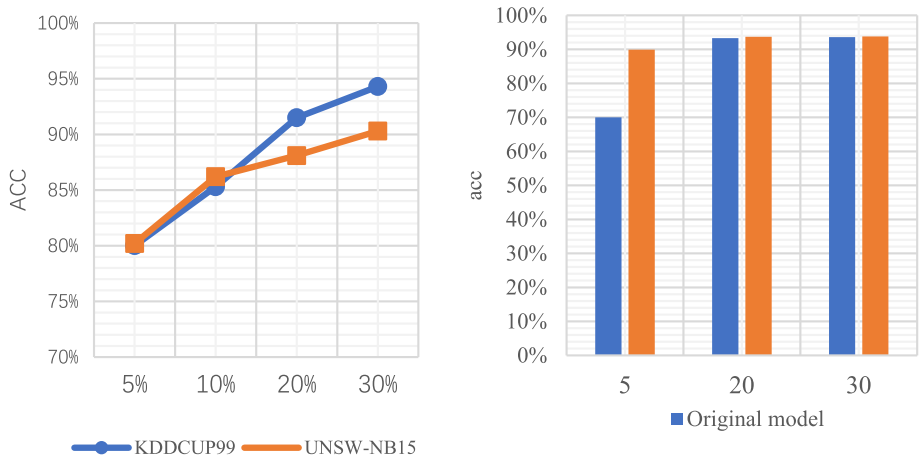


Fig. 8 The proportion of abnormal data and the influence of iterations on accuracy

Table 1 CIC-IDS-2017 data distribution

Class (Amount)	Attack Types in Data Set	Amount	Percentage
Training (1968801)	BENIGN	1,581,577	80.33%
	DoS	264,356	13.43%
	PortScan	110,364	5.61%
	Patator	9629	0.49%
	WebAttacks	1515	0.08%
	Bot	1360	0.07%
Testing (843773)	BENIGN	677,820	80.33%
	DoS	113,296	13.43%
	PortScan	47,299	5.61%
	Patator	4126	0.49%
	WebAttacks	649	0.08%
	Bot	583	0.07%

Table 2 Metrics comparison for CICIDS-2017 dataset

No.	Method	Accuracy	F1-Scrore	Precision	Recall
1	CSE-IDS [34]	92.00	–	–	98.00
2	Big Data Based DL [35]	97.8	–	–	97.8
3	Ensemble Based IDS [36]	88.96	–	–	96.25
4	TLHA [33]	98.46	98.1	86.42	96.25
5	Proposed Model	**99.81**	**99.79**	**99.80**	**99.81**

heterogeneous source and target domains, the lower-level fundamental features are common, while the higher-level features are domain-specific. Transferring these shared characteristics aids in training the target domain network, enhancing the iteration speed of

Table 3 Ablation study on the effectiveness of different components: CNN-LSTM.Adaptive Scaling (AS), Deep AE (DAE)

Components			Dataset Ratio		
CNN-LSTM	AS	DAE	10%	20%	50%
√			94.63	97.18	98.20
√		√	95.91	97.46	98.22
√	√		98.62	99.59	99.74
√	√	√	**99.52**	**99.69**	**99.79**

edge devices and reducing the cost and computational resources required for massive data collection.

Conclusion

This article focuses on developing a network intrusion detection model suitable for deployment at the edge, particularly addressing the challenges posed by imbalanced data. We propose a multi-classification approach for network intrusion detection based on reconstruction and feature matching. The method encompasses three key stages: Adaptive Scaling, Reconstruction and Feature Compression, and Feature Matching. Initially, data collected at the edge undergoes adaptive scaling to enhance feature distinctiveness. Subsequently, we introduce a reconstruction and compression method based on deep autoencoders, integrating features extracted by CNN-LSTM to derive multi-class classification results in the classifier. We compare this method with four existing approaches, and the results demonstrate its superior performance. Finally, we present a feature-matching-based model training approach tailored for edge devices to address scenarios involving novel attacks, reducing the training time required for the model to adapt to new threats.

To meet the demands for model size and speed in network intrusion detection at edge nodes, the deployed detection models on edge devices should be developed with a focus on lightweight characteristics. Consequently, our future emphasis will center on the application of lightweight devices in tasks related to detecting imbalanced intrusion network traffic. The goal is to improve model accuracy and shorten the time required for the model to learn and detect various types of attacks.

Authors' contributions

Yue Yang proposed the main ideas and principles of this research, designed and implemented parts of the algorithms and experimental schemes, and wrote the paper. Jieren Cheng guided the design of the algorithms and experiments, and oversaw the progress of the entire paper. Zhaowu Liu validated the dataset for the proposed model algorithm and visualized the results. Huimin Li assisted in drawing the charts and diagrams for the paper and revising the references. Ganglou Xu reviewed the entire manuscript and made revisions according to formatting requirements. All authors reviewed the manuscript.

Funding

This work was supported by National Natural Science Foundation of China (NSFC) (Grant No. 62162022,62162024), the Key Research and Development Program of Hainan Province (Grant No. ZDYF2020040,ZDYF2021GXJS003), the Major science and technology project of Hainan Province (Grant No. ZDKJ2020012), Hainan Provincial Natural Science Foundation of China (Grant No. 620MS021,621QN211), Science and Technology Development Center of the Ministry of Education Industry-university-Research Innovation Fund (2021JQR017), Innovative research project for Graduate students in Hainan Province (Grant No. Qhyb2022-93).

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 10 October 2023 Accepted: 28 December 2023
Published online: 03 February 2024

References

1. Wang F, Wang L, Li G et al (2022) Edge-cloud-enabled matrix factorization for diversified APIs recommendation in mashup creation. World Wide Web 25:1809–1829. https://doi.org/10.1007/s11280-021-00943-x
2. Yang Y, Yang X, Heidari M et al (2023) ASTREAM: data-stream-driven scalable anomaly detection with accuracy guarantee in IIoT environment. IEEE Trans Netw Sci Eng 10:3007–3016. https://doi.org/10.1109/TNSE.2022.3157730
3. Xu X, Tang S, Qi L et al (2023) CNN partitioning and offloading for vehicular edge networks in Web3. IEEE Commun Mag 61:36–42. https://doi.org/10.1109/MCOM.002.2200424
4. Wu S, Shen S, Xu X et al (2023) Popularity-aware and diverse web APIs recommendation based on correlation graph. IEEE Trans Comput Soc Syst 10:771–782. https://doi.org/10.1109/TCSS.2022.3168595
5. Wu Y, Nie L, Wang S et al (2023) Intelligent intrusion detection for internet of things security: a deep convolutional generative adversarial network-enabled approach. IEEE Internet Things J 10:3094–3106. https://doi.org/10.1109/JIOT.2021.3112159
6. Shiravani A, Sadreddini MH, Nahook HN (2023) Network intrusion detection using data dimensions reduction techniques. J Big Data 10:27. https://doi.org/10.1186/s40537-023-00697-5
7. Kong L, Tan J, Huang J et al (2023) Edge-computing-driven internet of things: a survey. ACM Comput Surv 55:1–41. https://doi.org/10.1145/3555308
8. Darzidehkalani E, Ghasemi-rad M, Van Ooijen PMA (2022) Federated learning in medical imaging: part II: methods, challenges, and considerations. J Am Coll Radiol 19:975–982. https://doi.org/10.1016/j.jacr.2022.03.016
9. Chalapathy R, Chawla S (2019) Deep learning for anomaly detection: a survey. https://doi.org/10.48550/arXiv.1901.03407
10. Yin J, Shi Y, Deng W et al (2023) Internet of things intrusion detection system based on convolutional neural network. Comput Mater Contin 75:2119–2135. https://doi.org/10.32604/cmc.2023.035077
11. Xie J, Wang H, Garibaldi JM, Wu D (2022) Network intrusion detection based on dynamic intuitionistic fuzzy sets. IEEE Trans Fuzzy Syst 30:3460–3472. https://doi.org/10.1109/TFUZZ.2021.3117441
12. Liu H, Lang B, Liu M, Yan H (2019) CNN and RNN based payload classification methods for attack detection. Knowl-Based Syst 163:332–341. https://doi.org/10.1016/j.knosys.2018.08.036
13. Yuan S, Ren K, Zhang C et al (2022) Canet: a hierarchical Cnn-attention model for network intrusion detection. SSRN J. https://doi.org/10.2139/ssrn.4243555
14. Huang X, Li Y, Ou L et al (2022) Research and implementation of industrial control network security intrusion detection classification based on deep learning. In: 2022 IEEE 10th joint international information technology and artificial intelligence conference (ITAIC). IEEE, Chongqing, China, pp 750–754
15. Thockchom N, Singh MM, Nandi U (2023) A novel ensemble learning-based model for network intrusion detection. Complex Intell Syst 9:5693–5714. https://doi.org/10.1007/s40747-023-01013-7
16. Ngo PC, Winarto AA, Kou CKL et al (2019) Fence GAN: towards better anomaly detection. In: 2019 IEEE 31st international conference on tools with artificial intelligence (ICTAI). IEEE, Portland, OR, USA, pp 141–148
17. Debicha I, Bauwens R, Debatty T et al (2023) TAD: transfer learning-based multi-adversarial detection of evasion attacks against network intrusion detection systems. Futur Gener Comput Syst 138:185–197. https://doi.org/10.1016/j.future.2022.08.011
18. Mahdavi E, Fanian A, Mirzaei A, Taghiyarrenani Z (2022) ITL-IDS: incremental transfer learning for intrusion detection systems. Knowl-Based Syst 253:109542. https://doi.org/10.1016/j.knosys.2022.109542
19. Guan H, Liu M (2022) Domain adaptation for medical image analysis: a survey. IEEE Trans Biomed Eng 69:1173–1185. https://doi.org/10.1109/TBME.2021.3117407
20. Ye Y, Lin Q, Ma L et al (2022) Multiple source transfer learning for dynamic multiobjective optimization. Inf Sci 607:739–757. https://doi.org/10.1016/j.ins.2022.05.114
21. Li Y, Peng X (2019) Learning domain adaptive features with unlabeled domain bridges. https://doi.org/10.48550/arXiv.1912.05004
22. Kang Z, Nielsen M, Yang B et al (2023) Online transfer learning with partial feedback. Expert Syst Appl 212:118738. https://doi.org/10.1016/j.eswa.2022.118738
23. Yang T, Yu X, Ma N et al (2022) Deep representation-based transfer learning for deep neural networks. Knowl-Based Syst 253:109526. https://doi.org/10.1016/j.knosys.2022.109526
24. Huang J, Guan D, Xiao A et al (2022) Category contrast for unsupervised domain adaptation in visual tasks. In: 2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, New Orleans, LA, USA, pp 1193–1204
25. Wang L, Wang M, Zhang D, Fu H (2022) Unsupervised domain adaptation via style-aware self-intermediate Domain. arXiv preprint arXiv:2209.01870, 2022
26. Yu C, Wang J, Chen Y, Huang M (2019) Transfer learning with dynamic adversarial adaptation network. In: 2019 IEEE international conference on data mining (ICDM). IEEE, Beijing, China, pp 778–786
27. Srinivas S, Fleuret F (2018) Knowledge transfer with jacobian matching. In: 2018 international conference on machine learning. PMLR, Sweden, pp 4723–4731
28. Zagoruyko S, Komodakis N (2017) Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. https://doi.org/10.48550/arXiv.1612.03928

29. Romero A, Ballas N, Kahou SE, et al (2015) FitNets: hints for thin deep nets. https://doi.org/10.48550/arXiv.1412.6550
30. Jang Y, Lee H, Hwang SJ et al (2019) Learning what and where to transfer. In: International conference on machine learning. PMLR, California, USA, pp 3030–3039
31. Chapaneri R, Shah S (2021) Multi-level Gaussian mixture modeling for detection of malicious network traffic. J Supercomput 77:4618–4638. https://doi.org/10.1007/s11227-020-03447-z
32. Sharafaldin I, Habibi Lashkari A, Ghorbani AA (2018) Toward generating a new intrusion detection dataset and intrusion traffic characterization: in: proceedings of the 4th international conference on information systems security and privacy. SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, pp 108–116
33. Harini R, Maheswari N, Ganapathy S, Sivagami M (2023) An effective technique for detecting minority attacks in NIDS using deep learning and sampling approach. Alex Eng J 78:469–482. https://doi.org/10.1016/j.aej.2023.07.063
34. Gupta N, Jindal V, Bedi P (2022) CSE-IDS: using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Comput Secur 112:102499. https://doi.org/10.1016/j.cose.2021.102499
35. Zhong W, Yu N, Ai C (2020) Applying big data based deep learning system to intrusion detection. Big Data Min Anal 3:181–195. https://doi.org/10.26599/BDMA.2020.9020003
36. Abbas A, Khan MA, Latif S et al (2022) A new ensemble-based intrusion detection system for internet of things. Arab J Sci Eng 47:1805–1819. https://doi.org/10.1007/s13369-021-06086-5

Publisher's Note