A Dynamic Intrusion Detection System Capable of Detecting Unknown Attacks

Na Xing, Shuai Zhao, Yuehai Wang, Keqing Ning, Xiufeng Liu
School of Information Science and Technology, North China University of Technology, Beijing, China

***Abstract*—In recent years, deep learning-based network intrusion detection systems (IDS) have shown impressive results in detecting attacks. However, most existing IDS can only recognize known attacks that were included in their training data. When faced with unknown attacks, these systems are often unable to take appropriate actions and incorrectly classify them into known categories, leading to reduced detection performance. Furthermore, as the number and types of network attacks continue to increase, it becomes challenging for these IDS to update their model parameters promptly and adapt to new attack scenarios. To address these issues, this paper introduces a dynamic intrusion detection system, Dynamic Unknown Attack Intrusion Detection System (DUA-IDS). This system aims to learn and detect unknown attacks effectively. DUA-IDS comprises three components: Feature Extractor: This component employs CNN and Transformer models to extract data features from various perspectives. Threshold-Based Classifier: The second part utilizes the nearest mean rule of samples to classify known and unknown attacks, enabling the distinction between them. Dynamic Learning Module: The third part incorporates data playback and knowledge distillation techniques to retain existing category knowledge while continuously learning new attack categories. To assess the effectiveness of DUA-IDS, this paper conducted experiments using the UNSW-NB15 public dataset. The experimental results show that DUA-IDS improves the classification accuracy of flow network data with unknown traffic attacks. Can accurately distinguish unknown traffic and correctly classify known traffic. When dynamically learning unknown traffic, the classification accuracy of previously learned known traffic is less affected. This indicates the advantages of DUA-IDS in detecting unknown attacks and learning new attack categories.**

Keywords*—Intrusion detection systems; transformer; DUA-IDS; data playback; variable perspectives features; Knowledge distillation

I. Introduction

With the increasing prevalence of interconnected devices, the issue of network security has gained significant prominence. Malicious network intrusions or attacks pose a threat to the fundamental elements of computer security policies, namely confidentiality, integrity, and availability. Hackers and cybercriminals continuously innovate their attack methods to steal data, gain control over host computers, and extort money. In order to combat these threats, Intrusion Detection Systems (IDS) have emerged as a major research focus in the field of network security. An IDS serves as a vital component of computer network security. Its primary function is to monitor network traffic for malicious activities, such as distributed denial of service attacks and injection attacks, and respond appropriately to intrusions. When an IDS detects a network attack on a computer, it promptly alerts the security administrator, enabling them to take necessary measures to mitigate the threat.

One of the main challenges in network intrusion detection systems is the ability to detect and recognize unknown events, including risks like zero-day attacks and low-frequency attacks such as worm attacks. Additionally, relying on administrators to manually detect, identify, and address network security issues is both inefficient and time-consuming. The limitations of the signature-based detection method further exacerbate the problem, as it cannot handle unknown threats and requires frequent updates to the signature database. Consequently, the optimal solution lies in equipping machines with the capability to analyze network data and identify suspicious or abnormal behavior.

This paper aims at the two problems mentioned above: detecting unknown events and frequently updating feature database manually. A dynamic intrusion detection system which can detect unknown attacks is proposed.

The rest of this paper is organized as follows: In the Section II, the related work is briefly reviewed. Section III introduces the network model and related algorithms. Section IV introduces the data set used in the laboratory. In the Section V, the experimental setup and results are clarified, and the simulation results are discussed. Finally, conclusions are drawn in Section VI.

II. Related Work

In the early days, IDS algorithms primarily relied on traditional machine learning methods like decision trees, support vector machines, genetic algorithms, and logistic regression. These approaches classified or clustered based on known attack characteristics. For instance, Senthilnayaki et al. [1] introduced an IDS method using the CART decision tree, which enhanced detection accuracy. Wang et al. [2] proposed a feature selection model based on sparse logistic regression, effectively identifying attack data. In recent years, researchers have combined deep learning algorithms with IDS due to their advancements. For example, Xiao et al. [3] developed a CNN-IDS using a convolutional neural network (CNN) that reduced input data dimensions through principal component analysis. The data was then transformed into a grayscale image and fed into the CNN for classification learning. Yan et al. [4] constructed a CNN-based IDS and employed it to generate synthetic attack trajectories against the network. Wang Y et al. [5] proposed a CNN-based network threat detection model that

avoided feature selection and manual extraction. They directly converted original traffic data into two-dimensional image data and employed CNN for detection and classification. Yang et al. [6] utilized long-term and short-term memory networks (LSTM) to extract time series characteristics from network traffic, employing an attention mechanism to capture data dependencies. Hassan et al. [7] designed a CNN and LSTM-based model (WDLSTM) with reduced weight. CNN was used for extracting local features, while WDLSTM preserved the time series features and prevented overfitting. HO et al. [8] transformed data streams into RGB images and classified them using a Vision-Transformer model. Yang et al. [9] proposed an improved ViT model, enhancing the local feature modeling capability by incorporating a sliding window mechanism. Wang W et al. [10] presented a robust unsupervised IDS (RUIDS) by introducing a mask context reconstruction module into the self-supervised learning based on Transformers. This approach maintained detection accuracy and improved the model's robustness.

Most IDS algorithms are trained using static datasets, meaning they can only recognize known attacks that were present in the training set. However, when faced with new and unknown attack methods, these models struggle to effectively classify and detect them. To address this issue, researchers have proposed various methods to identify unknown attacks. For instance, Cruz S et al. [11] utilized W-SVM to detect unknown attacks and evaluated their approach using the KDDCUP'99 dataset. Henrydoss et al. [12] introduced an open-set IDS method based on EVM, which achieved higher accuracy compared to W-SVM. Chen et al. [13] proposed a hierarchical detection model based on conditional variational self-coding. Their model learned the classification function for known attacks and the identification function for unknown attacks separately, resulting in good performance. Li et al. [14] developed a network intrusion generation countermeasure network called IDS-GAN. This approach involved scoring the output of normal network data using a discriminator to establish a normal interval. Any data falling outside of this interval was considered an unknown attack. These advancements aim to enhance the ability of IDS algorithms to detect and classify unknown attacks, thus improving their overall performance.

In practical applications, acquiring comprehensive datasets that encompass all types of intrusion detection poses a significant challenge. The collection of network attack types cannot be accomplished at once, as new attack methods continually emerge over time. Confronted with an increasing number of new and unknown attacks, simply categorizing them as unknown attacks can negatively impact the model's detection accuracy. However, retraining a model after merging new data with the original dataset is often impractical. To address this issue, class incremental learning has been proposed [15, 16]. This approach enables dynamic learning and allows for the updating of model parameters based on input from new classes. Li ZZ et al. [15] introduced the LwF model, which mitigates forgetting of previous knowledge by adding a constrained regularization term to the loss function of new tasks. This knowledge distillation technique freezes network layers and aids in retaining old knowledge. Rebeffisa et al. [16] proposed the iCaRL model, which addresses catastrophic forgetting by preserving representative old data and training it alongside new data. Zhang et al. [17] combined the K nearest neighbor clustering algorithm with the nearest class mean classifier to design an intrusion detection system called OCN, which supports class incremental learning. Wu et al. [18] proposed a method for intrusion detection based on dynamic integrated incremental learning (DEIL-RVM), which facilitates a dynamically adjusted integrated intrusion detection model. Zhang et al. [19] developed an incremental intrusion detection algorithm that utilizes an asymmetric multi-feature fusion self-encoder (AMAE) and a classification depth neural network (C-DNN). The confidence of the AMAE result is used to select the final result from the C-DNN output. This approach not only retains the ability to classify old data but also improves the detection accuracy for new categories.

However, the existing methods face limitations in simultaneously detecting unknown attacks and dynamically adding new attack categories. To address these challenges, this paper introduces a novel algorithm called DUA-IDS, which aims to enhance the classification of both old and new attack categories while improving the detection of unknown attacks. The main contributions of this paper are as follows:

1) This paper proposes a feature extraction module that combines a Transformer encoder [20] and a convolutional neural network (CNN). By leveraging the Transformer encoder, this module extracts time series and global features from the training data. The CNN network is employed to extract local features. The integration of these multi-angle features using a self-attention mechanism improves the model's classification capabilities for both new and old data categories.

2) A threshold-based classifier is introduced in this paper. During the training stage, the classifier calculates the threshold for the nearest mean classifier. In the testing stage, if the distance between the test data and the nearest class exceeds the threshold range, it is classified as unknown data. This approach enhances the model's ability to detect unknown data more effectively.

3) This paper presents a dynamic learning method to mitigate the problem of forgetting old data categories. The proposed method utilizes data playback and knowledge distillation techniques, along with the introduction of a spatial distillation loss function. This approach enhances the model's scalability and ability to retain knowledge about previously encountered attack categories.

III. METHOD

A. System Overview

To enhance the detection accuracy of new attacks, particularly unknown attacks encountered in real scenarios, and to dynamically update model parameters as the number of unknown attacks increases, this paper introduces a novel algorithm called DUA-IDS. The algorithm transforms unknown classes into known classes, thus improving the

model's overall performance. The following modules have been designed:

1) Feature extraction module: This module utilizes both a Transformer encoder and a CNN to extract time series information, global information, and local information from traffic data. The extracted multi-angle features are then fused using a self-attention mechanism, thereby improving the model's classification accuracy.

2) Unknown data detection module: This module employs a threshold-based classifier. By calculating the threshold of the nearest mean classifier, the module can determine the distance between the test data and the nearest class. If the distance falls within the threshold range, the data is classified as a known class and further categorized accordingly. Conversely, if the distance exceeds the threshold range, the data is classified as an unknown class.

3) Model dynamic updating module: For the unknown categories identified in the previous module, these categories are marked using other data cleaning algorithms. Based on the principles of data playback and knowledge distillation, the model continuously learns the unknown categories while retaining the detection accuracy for old categories.

These modules collectively contribute to the DUA-IDS algorithm, enabling improved detection accuracy for emerging new attacks and facilitating the dynamic update of model parameters in response to the increasing number of unknown attacks.

The overall framework of DUA-IDS intrusion detection system is shown in Fig. 1.

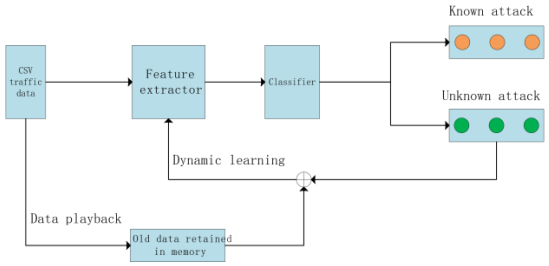


Fig. 1. Overall framework of model.

B. Feature Extraction Module

As shown in Fig. 2, this paper proposes a neural network structure with multi-angle feature fusion. It includes Transformer encoder, CNN and Attention Fusion.

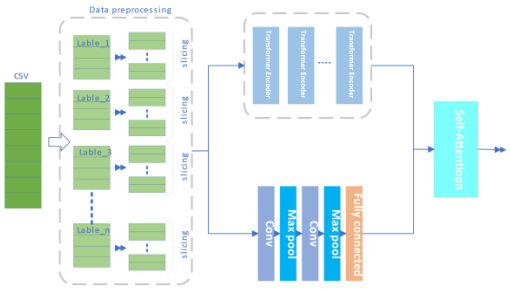


Fig. 2. Feature extractor.

The original Transformer model consists of an encoder and a decoder, but since intrusion detection primarily involves classification, this module only utilizes the encoder component. The encoder is composed of two main sublayers: the Multi-Head Attention mechanism and a fully connected Feed-Forward Network. By incorporating position coding and the self-attention mechanism of the Transformer encoder, it becomes possible to effectively extract global features and time sequence features from each subsequence.

The Multi-Head Attention mechanism serves as the core of the Transformer encoder, enabling the capture of dependencies between data within stream data blocks. This mechanism operates by first linearly transforming the input stream's data block into three vectors: Query, Key, and Value. Next, the attention distribution is computed by comparing the query vector with all the key vectors. Each value vector is then multiplied by its corresponding attention distribution to obtain a weighted value vector. Ultimately, these weighted value vectors are concatenated to form a new feature vector. By utilizing different attention mechanisms, diverse attention representations can be obtained, thereby providing a comprehensive reflection of the information shared among the data within the input data stream block. The calculation formula (1) for the attention mechanism is as follows:

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},V)=\text{softmax}(\frac{\boldsymbol{QK}^T}{\sqrt{d_k}})V \tag{1}$$

Where Q, K and V respectively represent three matrices of Query, Key and Value, and dk is the dimension of Key. The calculation formula (2) of multi-head attention is as follows:

$$\begin{cases} Q_i = XW_i^Q, \mathbf{K}_i = XW_i^K, V_i = XW_i^V & i=1,....,n \\ head_i = \text{Attention}(Q_i, \mathbf{K}_i, V_i) & i=1,....n \\ \text{MultiHead}(Q,\mathbf{K},V) = \text{Concat}(head_1,...,head_n)W^o & \end{cases} \tag{2}$$

Where W^Q represents a parameter when Q is calculated for input X, W^K represents a parameter when K is calculated for input X, and W^V represents a parameter when V is calculated for input X.

The CNN component comprises two convolution layers, two maximum pooling layers, and a fully connected layer. The two-dimensional convolution neuron (Conv2D) is primarily employed in the convolution layers for processing. In this study, CNN is predominantly utilized to extract local spatial features from network traffic data.

Subsequently, the attention mechanism is applied to integrate global features, time series features, and local features from multiple perspectives. The feature extraction module enables the consideration of correlations and complementarity among features, thereby enhancing the model's expressive and generalization capabilities.

C. Unknown Data Detection Module

The nearest class mean classifier is a simple and effective classification algorithm that falls under the category of centroid-based classifiers. In this classifier, each class is represented by the mean vector of its training samples, which serves as a prototype for that particular class. During the

testing phase, the distance between the test sample and each class prototype is calculated, and the test sample is assigned to the class whose prototype is closest to it.

To be more specific, suppose we have a training set consisting of K classes. Let $\boldsymbol{\mu}_k$ represent the mean vector of class k. The mean vector is obtained by taking the average of all the training samples belonging to that specific class. In order to classify a new test sample $\mathbf{x}$, we compute the distance between $\mathbf{x}$ and each class prototype.

$$d_k = \| \mathbf{x} - \mu_k \| \tag{3}$$

where $|\cdot|$ denotes the Euclidean distance. Finally, the test sample is assigned to the class whose prototype has the smallest distance.

The traditional nearest class mean classifier is limited to classifying known classes only. It determines the class of a test data by calculating the distance between the test data and the centroid of each known class, and assigning it to the class with the closest centroid. Regardless of the actual distance, the test data is always classified into that specific category. This approach poses a problem when encountering unknown categories, as there will always be a known class whose centroid is the closest, even if the actual distance is significantly large. To address this issue, we can enhance the nearest distance calculation. By introducing a threshold on the nearest distance, we can determine whether a test data belongs to an unknown class. If the distance between the test data and the centroid of the nearest class exceeds this threshold, it is classified as an unknown class. The classification calculation method is as follows:

$$\hat{y}_i = \begin{cases} \arg\min\limits_{k\in\{1,\cdots,m\}} d(f(x_i), c_k), & if\ \min d(f(x_i), c_k) < threshold_k \\ m+1, & otherwise. \end{cases} \tag{4}$$

Where $d()$ an example of test data and class centroid is, m is the number of classes, $threshold_k$ is the classifier threshold, c_k is the class centroid of class k and $f(x_i)$ is the result of model output.

In this paper, we utilize a threshold-based classifier, as illustrated in Fig. 3, to identify unknown categories. The threshold for the nearest mean classifier is computed based on the centroids of both known and unknown classes, enabling us to distinguish between them.

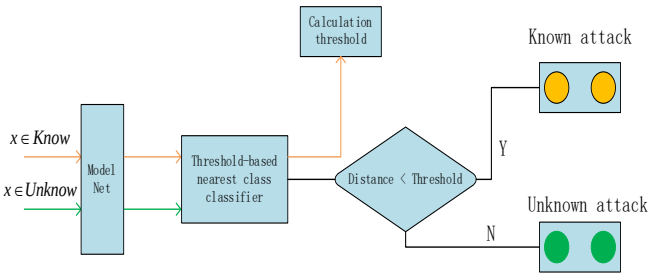


Fig. 3. Process of classifying known categories and unknown categories.

Algorithm1: Calculation threshold

Input: Training data x $x \in know$
Require: Number of each category n $n = (n_1, n_2, \cdots, n_m)$
Require: Feature extractor $\varphi(x) = model(x)$
1: FOR $k = \{1, 2, \cdots, m\}$ DO
2: $\mu_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \varphi(x_i)$ // Calculate class mean
3: END FOR
4: Calculate the minimum distance
from the center of mass in each category
5: $Threshold_k = \min\limits_{k=1\cdots m} \| \varphi(x) - \mu_k \|$
6: $Threshold = \max\limits_{k=1\cdots m} Threshold_k$ // Calculate the classifier //threshold
7: END

The threshold calculation algorithm and classification algorithm are shown in algorithm 1 and algorithm 2.

Algorithm 2: Classification algorithm

Input: Training data x $x \in know \,\&\, unkonw$
Require: Number of each category n $n = (n_1, n_2, \cdots, n_m)$
Require: Feature extractor $\varphi(x) = model(x)$
1: FOR $k = \{1, 2, \cdots, m\}$ DO
2: $\mu_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \varphi(x_i)$ // Calculate class mean
3: END FOR
4: IF $\min\limits_{k=1\cdots m} \| \varphi(x) - \mu_k \| < Threshold$
5: $\hat{y} = \arg\min\limits_{k=1\cdots m} \| \varphi(x) - \mu_k \|$ //Classified as the //category closest to the center of mass.
6: ELSE
7: $\hat{y} = m+1$ //unknow attach
8: END IF
9: END

D. Dynamic Learning Module

When dynamically learning new attack categories, the model often experiences catastrophic forgetting of the known attack categories. The feature extraction module's training objective is to accurately classify the training flow data into a predefined set of categories. However, when a new attack category emerges, the model needs to update its parameters to incorporate the new information. Unfortunately, as the model updates its parameters, it may "over-fit" the new data and forget the previously learned information, resulting in a significant decrease in the detection accuracy of known attack categories. This decline in accuracy occurs because the new data may differ in distribution or feature space from the old data. Consequently, the model might adjust its parameters in a

way that sacrifices the performance of old data while improving the performance on the new data. To address this challenge, this paper employs a method based on data playback and knowledge distillation to reinforce the learning of known attack categories.

1) Data playback: Once each task is learned, the data playback method saves a subset of samples from each category for future model training, as depicted in Fig. 4. These saved samples, comprising known attack categories, are combined with the data of new attack categories to update the parameters of the current feature extraction module. In this paper, a sample saving strategy is employed. The total number of saved samples is fixed at K, and the number of samples saved in each category is n=K/m, where m represents the number of currently known attack categories. This approach ensures that the available storage of K samples is utilized efficiently.

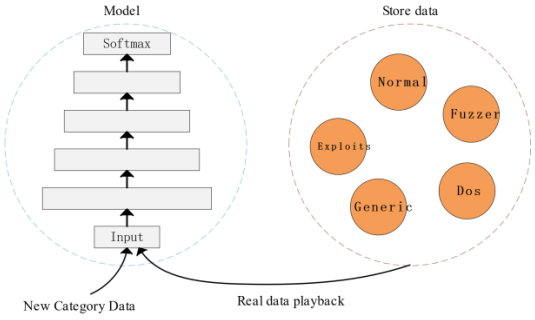


Fig. 4. Class incremental dynamic learning.

Furthermore, this paper employs the "Herding" strategy to select n representative samples from each category. The strategy operates in iterations to choose N samples. In each iteration, a sample is selected from the current training set and included in the saved sample set. This approach aims to ensure that the average feature vector of the saved feature space for known samples closely aligns with the average feature vectors of all training samples. Consequently, the saved set of known samples functions as a priority queue, with the sample order indicating their significance. The specific process of saving the sample set using the "Herding" strategy is outlined in Algorithm 3.

Algorithm 3: Sample preservation based on Herding strategy

Input: Training sample set $X=\{x_1,\cdots,x_m\}$ belonging to category y

Require: Number of samples stored in each category n

Require: Feature extractor $\varphi(x)=model(x)$

1: Initialize: $\mu=\frac{1}{m}\sum_{x\in X}\varphi(x)$

2: FOR $k=\{1,2,\cdots,n\}$ DO

3: $p_k=\underset{x\in X}{\operatorname{argmin}}\,\|\mu-\frac{1}{k}[\varphi(x)+\sum_{j=1}^{k-1}\varphi(p_j)]\|$

4: END FOR

5: $M^y\leftarrow(p_1,p_2,\cdots,p_n)$

2) Knowledge distillation: When learning a new model, the knowledge distillation method requires that the output of the new model aligns with the given data in a consistent manner with the old model. The old model refers to the model that learned the previous task (known attack category) and remains unchanged. The new model inherits the parameters of the old model and can be updated for new tasks. During the training process, the new model is optimized to minimize two loss functions: (i) The standard cross entropy loss between the predictions of the new model and the actual labels, and (ii) Distillation loss, which measures the disparity between the predictions of the new model and the soft targets generated by the old model.

The traditional distillation loss primarily focuses on the distillation output of the final feature layer of both the old and new models. However, this paper delves into the distillation calculation of the output from the middle layer of the feature extraction model. By combining the output of the final feature layer with each layer's output, the new model can effectively absorb the knowledge of the old model and mitigate the problem of catastrophic forgetting for known attack categories.

Representation of distillation loss of final characteristic layer:

$$L_{\text{end}}(\mathbf{h}^{t-1},\mathbf{h}^{t})=\left\|\mathbf{h}^{t-1}-\mathbf{h}^{t}\right\|^2 \tag{5}$$

Where h^{t-1} represents the final feature layer output of the old model and h^t represents the final feature layer output of the new model being trained.

Aiming at the feature extraction module in this paper, the data shape after preprocessing the traffic data is as follows: (bitch_size, series, feature_size), abbreviation (B，S，F), For the output of the middle layer of the feature extraction module, the distillation loss is calculated from two dimensions respectively.

Distillation loss in two dimensions.

$$L_{\text{S}}(\mathbf{h}_{\ell}^{t-1},\mathbf{h}_{\ell}^{t})=\sum_{f=1}^{F}\left\|\sum_{s=1}^{S}\mathbf{h}_{l,s,f}^{t-1}-\sum_{s=1}^{S}\mathbf{h}_{l,s,f}^{t}\right\|^2 \tag{6}$$

$$L_{\text{F}}(\mathbf{h}_{\ell}^{t-1},\mathbf{h}_{\ell}^{t})=\sum_{s=1}^{S}\left\|\sum_{f=1}^{F}\mathbf{h}_{l,s,f}^{t-1}-\sum_{f=1}^{F}\mathbf{h}_{l,s,f}^{t}\right\|^2 \tag{7}$$

$$L_{\text{spatial}}(\mathbf{h}_{\ell}^{t-1},\mathbf{h}_{\ell}^{t})=L_{\text{S}}(\mathbf{h}_{\ell}^{t-1},\mathbf{h}_{\ell}^{t})+L_{\text{F}}(\mathbf{h}_{\ell}^{t-1},\mathbf{h}_{\ell}^{t}) \tag{8}$$

Where h_l^{t-1} represents the characteristic output of the first layer of the old model, h_l^t represents the characteristic output of the first layer of the new model, and S and F represent the intermediate layer loss in the S and F dimensions respectively.

Combining the final characteristic layer output and the intermediate characteristic layer output as the distillation loss of the model.

$$L_{\text{final}}(\mathbf{x}) = \frac{\lambda_c}{L-1}\sum_{\ell=1}^{L-1} L_{\text{spatial}}\left(f_\ell^{t-1}(\mathbf{x}), f_\ell^{t}(\mathbf{x})\right) + \lambda_f L_{\text{end}}\left(f^{t-1}(\mathbf{x}), f^{t}(\mathbf{x})\right) \quad (9)$$

Where the hyperparameter λ_c and λ_f are used to balance the output of the intermediate layer and the output of the final layer, L represents the total number of layers of the feature module, and L-1 is the other intermediate layers except the final layer.

3) Model parameter updating algorithm: Whenever the model acquires new attack data ($X_{m+1},...,X_{m+k}$) for the new categories (m+1,...,m+k), these data are utilized to update the parameters of the feature extraction module and the cached sample set. Algorithm 4 outlines the steps involved in updating the feature extraction module parameters.

Algorithm 4: Class incremental learning algorithm

Input: New category data $X = \{x_{m+1}, \cdots, x_{m+k}\}$

Require: Old data cached in memory $P = \{P_1, \cdots, P_m\}$

1: Initialize: Training data $X_{train} = X \bigcup P$

2: FOR $index, x, label$ in X_{train} DO

3: Save the output of the middle layer of the feature extractor network. $q[index] = \varphi(x)$

4: END FOR

5: Train model one epoch:

6: FOR x, y in X_{train} DO

7: $\hat{y} = \varphi(x)$

8: $Loss = Loss_{ce} + Loss_{final}$

9: Update parameters

10: END FOR

11: END

First, the cached data of known attack categories and the input data of new attack categories are combined into a new training dataset. Next, the output results of each layer of the old feature extraction module for the known attack categories are stored. Finally, the parameters of the feature extraction module are updated using the loss function, which is a weighted sum of the classification cross entropy loss and the hierarchical distillation loss.

IV. Dataset Description

A. UNSW-NB15

The dataset used in this paper is the UNSW-NB15[21-26], which is a comprehensive dataset designed by the Australian Cyber Security Center Laboratory in 2015 for network intrusion detection systems. The dataset aims to simulate real attack environments using the IXIA traffic generator, based on vulnerability information technology published on the CVE website. It consists of normal traffic and various types of attack traffic, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms.

The dataset comprises 49 features and 10 labels. These features include basic information such as source IP address, destination IP address, and protocol type, content-related features like HTTP method and URI length, time-related features such as connection duration and average byte rate, as well as other features such as service type and status code. The dataset is available in three versions: full version, training version, and test version. For this experiment, we utilize the full version of the CSV data, which contains 2,540,044 data samples.

Considering the significant imbalance in the number of normal samples compared to other samples, random sampling is applied to reduce the impact of this data imbalance on the classification results. Known classes include common attack types like DOS, Generic, Exploits, and Fuzzers, while the remaining classes are treated as unknown classes. Detailed data can be found in Table I.

TABLE I. Number of Categories in the UNSW-NB15 Dataset

Categories	Data Size
Normal	200000
Fuzzers	24240
Dos	16350
Exploits	44520
Generic	215480
Backdoor	2320
Analysis	2670
Reconnaissance	13980
ShellCode	1510
Worms	170

V. Experiment

A. Experimental Environment

Experimental configuration: Ubuntu as the operating system, Intel Xeon Gold 5118 as the processor, 32GB of memory, NVIDIA GeForce RTX 3090 as the GPU, Python3.8 as the programming language and Pytorch 1.11.0 as the learning framework.

B. Evaluation Indexes

This experiment is a classification model, using a confusion matrix to evaluate the classification structure. The confusion matrix is shown in Table II.

TABLE II. The Confusion Matrix

The real situation	Prediction result	
	Positive	***Negative***
True	TP	FN
False	FP	TN

The evaluation indexes used in the experiment mainly include Accuracy, Precision, Recall and F1-score.

Accuracy refers to the proportion of the number of samples that can be correctly classified by the model in the total samples.

$$Acuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (10)$$

Precision represents the ratio of the number of correctly classified samples to the number of samples retrieved.

$$Precision = \frac{TP}{TP + FP} \quad (11)$$

Recall is the ratio of the number of correctly classified samples to the number of correctly classified samples.

$$Recall = \frac{TP}{TP + FN} \quad (12)$$

F1-score refers to the harmonic mean of Precision and Recall.

$$F1 - score = \frac{2 * P * R}{P + R} \quad (13)$$

C. Results and Discussion

1) Hyperparameter setting: Through many experiments, the values of parameters are adjusted to achieve the best experimental results. In the process of adjustment, the super parameters of the best experimental results are obtained, as shown in Table III.

TABLE III. MODEL HYPERPARAMETER

Hyperparameter	Value
Epochs	50
Learning rate	1e-5
Time series	10
Bitch size	10
Number of headers	8
Number of layers	3
Embding size	768
K	2000

2) Experimental analysis of known attack detection: To evaluate the detection performance of DUA-IDS on both known categories and unknown attacks, we have selected several advanced intrusion detection methods as baselines.

a) Methods for detecting unknown attacks:

i) EVM [12]: Extreme Value Machine (EVM) is a classifier that detects unknown attack categories using non-nuclear, nonlinear, and variable bandwidth outlier detection.

ii) IDS-GAN [14]: IDS-GAN defines the normal interval based on the scoring of normal network data by the discriminator. Any data outside this interval is considered an unknown attack.

b) Methods unable to detect unknown attacks:

i) CNN-BiLSTM [27]: This model combines CNN, BiLSTM, and self-encoder to extract high-dimensional traffic features and self-monitoring features, aiming to improve the classification accuracy.

ii) CNN-WDLSTM [26]: CNN-WDLSTM adopts a combination of CNN and LSTM (WDLSTM) with reduced weight. CNN is used to extract local features, while WDLSTM preserves time series features and prevents overfitting, thus enhancing the classification accuracy.

iii) RUIDS [10]: RUIDS is a robust unsupervised intrusion detection system and one of the most advanced closed-set intrusion detection algorithms. It incorporates a shielded context reconstruction module into the self-supervised learning scheme based on the transformer. The self-supervised learning scheme captures internal relationships within the learning context.

Fig. 5 illustrates the loss function of these six groups of models during the training process on known classes using the UNSW-NB15 dataset. The figure demonstrates that all six models exhibit good convergence on the training dataset, yielding optimal parameters. Subsequent experiments will evaluate the model's performance on the test set using these optimal parameters.

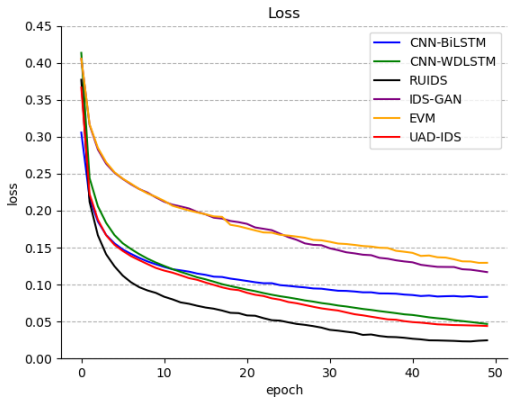


Fig. 5. The loss vs. epoch under different model.

Table IV presents the intrusion detection results of the model proposed in this paper on the test set of the UNSW-NB15 dataset. It compares the classification accuracy of our model with other models of different types on known attack categories. From the data in the table, it is evident that the classification accuracy of intrusion detection algorithms capable of detecting unknown categories in the past is not as high as those based on closed sets (where unknown categories cannot be detected). To enable the detection of unknown categories, our model adopts a non-optimal classification strategy. Consequently, its classification accuracy on known categories is slightly lower than that of closed-set detection models.

TABLE IV. EXPERIMENTAL RESULTS OF THE MODEL ON KNOWN CATEGORIES IN THE UNSW-NB15 DATASET

	UNSW-NB15		
	Acc	F1	Pre
EVM	0.841	0.8379	0.8345
IDS-GAN	0.8582	0.8627	0.8442
CNN-BiLSTM	0.8689	0.865	0.8582
CNN-WDLSTM	0.8786	0.8725	0.8688
RUIDS	0.891	0.8908	0.8871
DUA-IDS	0.8847	0.8854	0.8794

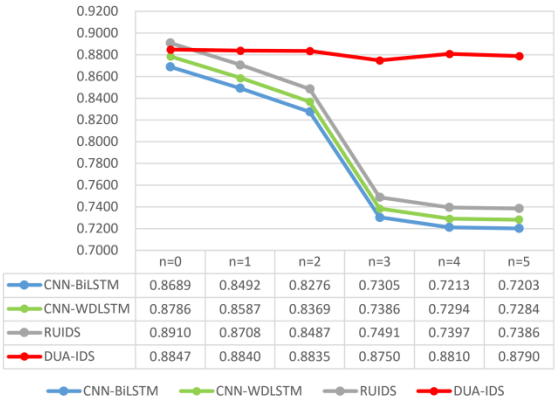
Compared with other models, EVM adopts the traditional method to classify network flow data based on outliers. In binary classification, it may have better classification accuracy. For multi-classification tasks, the accuracy of multi-classification is weaker than other models.

DUA-IDS is superior to IDS-GAN in the utilization of time series features. DUA-IDS obtains more comprehensive characteristics of network stream data.

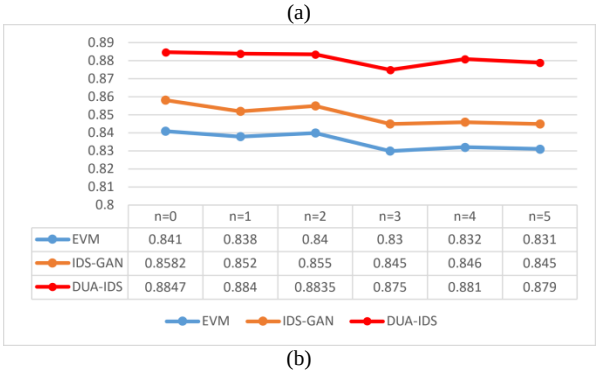
Compared with CNN-BiLSTM and CNN-WDLSTM, the Transformer proposed in this paper combines CNN's feature extractor to analyze the time, global and local features of data.

However, the classification accuracy of the model proposed in this paper surpasses that of most closed-set-based intrusion detection algorithms. It is only 0.63% lower than the most advanced closed-set algorithm, RUIDS. This demonstrates the model's strong advantages in feature extraction and classification.

3) Experimental analysis of unknown attack detection: In real-world scenarios, there may be attack categories that are unseen and do not belong to our known training set. To address this, we employ a nearest class mean classifier with a threshold to identify unknown attack categories in this paper. To evaluate the performance of the proposed model in detecting unknown attack categories, we compare DUA-IDS with both closed-set-based intrusion detection models and models capable of detecting unknown categories. The experimental results are depicted in Fig. 6.



(a)



(b)

Fig. 6. (a). Comparison with the algorithm based on closed-set, (b). Comparison of models for detecting the accuracy of unknown categories.

In this paper, we systematically introduce unknown categories into the test dataset one by one, following the order specified in Table II. The number of added unknown categories is represented as "n=1" in Fig. 6. As depicted in Fig. 6(a), it is evident that our proposed DUA-IDS outperforms the closed-set-based intrusion detection model as the number of unknown attack categories increases. The closed-set model can only recognize data from known categories, and when confronted with new categories, it mistakenly classifies them into known categories, resulting in a continuous decline in classification accuracy. This demonstrates the necessity for intrusion detection models to possess the ability to detect unknown classes.

Fig. 6(b) illustrates the classification accuracy of DUA-IDS and other comparison models, which tends to stabilize as the number of unknown attack categories increases. These models effectively identify most unknown attacks as unknown classes, thereby maintaining high classification accuracy. Notably, the proposed DUA-IDS consistently outperforms the comparison model in classification accuracy, highlighting its superior capability in detecting unknown attack categories.

4) Experimental analysis of dynamic class increment: To evaluate the detection performance of DUA-IDS on both original and new categories after updating the model parameters with the addition of new categories, we compare it with the OCN [17] model in this paper.

OCN: The OCN model combines the concepts of nearest class classifier and K nearest neighbor clustering. It directly incorporates the clustering results into the classifier's category list, enabling incremental learning of categories.

Fig. 7 illustrates the changes in classification accuracy for both DUA-IDS and OCN models when new categories are added. We incrementally introduce one new category at a time, denoted as "n=1" in the figure. As depicted in the figure, as the number of new categories increases, the classification accuracy of both models decreases. However, the proposed DUA-IDS in this paper exhibits a slight decrease in accuracy when new categories are added, while the OCN model experiences a noticeable decline. This indicates that the DUA-IDS model proposed in this paper possesses significant advantages in dynamically updating the model.

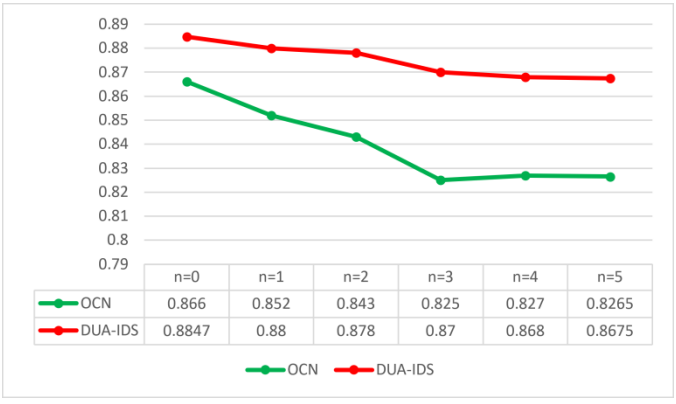
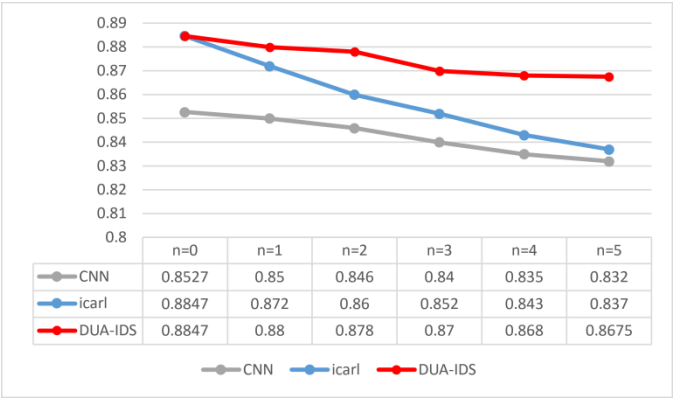


Fig. 7. Comparison of dynamic learning models.

5) Ablation experiment: In the DUA-IDS proposed in this paper, we have incorporated two key elements: a feature extraction module based on transformer and spatial distillation. To further investigate their effectiveness, we conducted ablation experiments focusing on these two aspects.

a) CNN-DUA-IDS: To evaluate the impact of the feature extraction module based on transformer designed in this paper, we conducted a control experiment by replacing it with a simple CNN module. As illustrated in Fig. 8, the utilization of the CNN module resulted in a decrease in overall model accuracy. This clearly demonstrates the significant role played by the transformer-based feature extraction module in enhancing the classification accuracy of the model.



	n=0	n=1	n=2	n=3	n=4	n=5
CNN	0.8527	0.85	0.846	0.84	0.835	0.832
icarl	0.8847	0.872	0.86	0.852	0.843	0.837
DUA-IDS	0.8847	0.88	0.878	0.87	0.868	0.8675

Fig. 8. Comparison of ablation experimental results.

b) icarl-DUA-IDS: To investigate the impact of the introduced spatial distillation loss in this paper, we conducted a control experiment by removing it from the DUA-IDS model. As depicted in Fig. 8, when a new category is added, the classification accuracy of icarl-DUA-IDS noticeably decreases. This observation highlights the substantial contribution of the spatial distillation loss in mitigating the effects of knowledge forgetting.

VI. Conclusion

In the actual network environment, network data is dynamic and will produce new types of unknown network data in real time. The traditional intrusion detection system based on static data can't adapt well to new types of data, which leads to the decrease of classification accuracy. In addition, the increasing number of unknown data also poses a potential threat to the stability of the model.

This paper addresses the challenges of detecting unknown network attacks and dynamically updating models in network intrusion detection. To tackle these issues, a dynamic intrusion detection system capable of detecting unknown attacks is proposed. The model is mainly composed of three parts. Firstly, in the feature extraction module, this paper proposes to combine the multi-angle features of network data by combining Transformer with CNN. Secondly, the nearest class mean classifier based on threshold is used to find the potential unknown attack categories. Thirdly, the unknown class data are updated by class increment method based on data playback and distillation learning. The experimental results show that the method proposed in this paper is effective in detecting unknown data and stable after dynamically learning new types of data.

However, the DUA-IDS model has certain limitations when it comes to handling detected unknown attack categories and incorporating them back into the model. Therefore, additional measures are required to cleanse and label the identified unknown attack data before further learning. Future research efforts will focus on refining the process of handling and relearning unknown data, aiming to achieve automated self-learning in network intrusion detection models.

Acknowledgment

This work was supported in part by the R&D Program of Beijing Municipal Education Commission (KM202310009001), in part by the Scientific Research Foundation of North China University of Technology (110051360002), and in part by the National Key Research and Development Program of China (2020YFC0811004).

References

[1] C. M. K. Ho, K.-C. Yow, Z. Zhu, S. Aravamuthan," Network Intrusion Detection via Flow-to-Image Conversion and Vision Transformer Classification." IEEE Access 10, 97780-97793 2022.

[2] B. Senthilnayaki, K. Venkatalakshmi, A. Kannan," Intrusion detection system using fuzzy rough set feature selection and modified KNN classifier." Int. Arab J. Inf. Technol. 16, 746-753 2019.

[3] Y. Wang," A multinomial logistic regression modeling approach for anomaly intrusion detection." Computers & Security 24, 662-674 2005.

[4] Y. Xiao, C. Xing, T. Zhang, Z. Zhao," An intrusion detection model based on feature reduction and convolutional neural networks." IEEE Access 7, 42210-42219 2019.

[5] Q. Yan, M. Wang, W. Huang, X. Luo, F. R. Yu," Automatically synthesizing DoS attack traces using generative adversarial networks." International journal of machine learning and cybernetics 10, 3387-3396 2019.

[6] Y. Wang, J. An, W. Huang, in 2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS). (IEEE, 2018), pp. 400-404.

[7] S. Yang, M. Tan, S. Xia, F. Liu, in Proceedings of the 2020 5th International Conference on Machine Learning Technologies. (2020), pp. 46-50.

[8] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, G. Fortino," A hybrid deep learning model for efficient intrusion detection in big data environment." Information Sciences 513, 386-396 2020.

[9] Y. G. Yang, H. M. Fu, S. Gao, Y. H. Zhou, W. M. Shi," Intrusion detection: A model based on the improved vision transformer." Transactions on Emerging Telecommunications Technologies 33, e4522 2022.

[10] W. Wang, S. Jian, Y. Tan, Q. Wu, C. Huang," Robust unsupervised network intrusion detection with self-supervised masked context reconstruction." Computers & Security 128, 103131 2023.

[11] S. Cruz, C. Coleman, E. M. Rudd, T. E. Boult, in 2017 IEEE International Symposium on Technologies for Homeland Security (HST). (IEEE, 2017), pp. 1-6.

[12] J. Henrydoss, S. Cruz, E. M. Rudd, M. Gunther, T. E. Boult, in 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). (IEEE, 2017), pp. 1089-1093.

[13] X. Chen, doctor, University of Science and Technology of China (2021).

[14] X. Li, master, University of Electronic Science and Technology of China (2022).

[15] Z. Li, D. Hoiem," Learning without forgetting." IEEE transactions on pattern analysis and machine intelligence 40, 2935-2947 2017.

[16] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C. H. Lampert, in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. (2017), pp. 2001-2010.

[17] Z. Zhang, Y. Zhang, D. Guo, M. Song," A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks." International Journal of Machine Learning and Cybernetics 12, 1649-1665 2021.

[18] Z. Wu, P. Gao, L. Cui, J. Chen," An incremental learning method based on dynamic ensemble RVM for intrusion detection." IEEE Transactions on Network and Service Management 19, 671-685 2021.

[19] B. Zhang, H. Xia, Y. Zhang, Z. Gao," Incremental intrusion detection based on multi-feature fusion automatic encoder." Computer systems & applications 32, 42-50 2023.

[20] A. Vaswani et al.," Attention is all you need." Advances in neural information processing systems 30, 2017.

[21] Y. Zhang et al.," PCCN: parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows." IEEE Access 7, 119904-119916 2019.

[22] N. Moustafa, J. Slay, in 2015 military communications and information systems conference (MilCIS). (IEEE, 2015), pp. 1-6.

[23] N. Moustafa, J. Slay," The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." Information Security Journal: A Global Perspective 25, 18-31 2016.

[24] N. Moustafa, J. Slay, G. Creech," Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks." IEEE Transactions on Big Data 5, 481-494 2017.

[25] N. Moustafa, G. Creech, J. Slay," Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models." Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications, 127-156 2017.

[26] M. Sarhan, S. Layeghy, N. Moustafa, M. Portmann, in Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings 10. (Springer, 2021), pp. 117-135.

[27] Liang, Xing, Hou," CNN-BiLSTM network access detection method based on self-supervised feature enhancement." Journal of Electronic Measurement and Instrument 36, 65-73 2022.