www.nature.com/scientificreports

scientific reports

OPEN

Multimodal fusion based few-shot network intrusion detection system

Congyuan Xu[1,2]✉, Yong Zhan[3], Zhiqiang Wang[1] & Jun Yang[1]

As network environments become increasingly complex and new attack methods emerge more frequently, the diversity of network attacks continues to grow. Particularly with new or rare attacks, gathering a large number of labeled samples is extremely difficult, resulting in limited training data. Existing few-shot learning methods, while reducing reliance on large datasets, mostly handle single-modality data and fail to fully exploit complementary information across different modalities, limiting detection performance. To address this challenge, we introduce a multimodal fusion based few-shot network intrusion detection method that merges traffic feature graphs and network feature sets. Tailored to these modal characteristics, we develop two models: the G-Model and the S-Model. The G-Model employs convolutional neural networks to capture spatial connections in traffic feature graphs, while the S-Model uses the Transformer architecture to process and fuse network feature sets with long-range dependencies. Furthermore, we extensively study the fusion effects of these two modalities at various interaction depths to enhance detection performance. Experimental validation on the CICIDS2017 and CICIDS2018 datasets demonstrates that our method achieves multi-class accuracy rates of 93.40% and 98.50%, respectively, surpassing existing few-shot network intrusion detection methods.

Keywords Intrusion detection, Network security, Multimodal, Few-shot, Deep learning

Amid rapid technological advancements, network security has emerged as a pivotal component of contemporary security infrastructures[1]. Traditional Network Intrusion Detection Systems (NIDS) are adept at managing attacks that conform to well-defined rules, known patterns, or signatures. By exploiting predefined rule sets and signature databases, these systems effectively thwart such threats, excelling in stable and predictable environments. Nonetheless, as network environments evolve swiftly and attack methodologies continue to advance, particularly with respect to high-dimensional data complexity and diverse attack techniques, traditional NIDS are increasingly challenged. The rapid development of deep learning offers novel solutions to these challenges. Deep Neural Networks (DNNs), celebrated for their superior feature extraction capabilities, are becoming a dominant technology for enhancing NIDS performance[2]. Deep learning's capacity to derive insights from vast and complex datasets enables NIDS to more effectively detect intricate network attacks. This signifies a progressive transformation of traditional intrusion detection methods through deep learning to address contemporary network security threats. However, present deep learning models generally depend on substantial volumes of training data. In high-security network environments, it is impractical to delay responses until frequent attacks occur. Thus, there is an imperative need for systems that can perform rapid and effective detection with limited samples. Proactive defense is crucial in addressing novel and rapidly evolving attack patterns, particularly when these attacks often lack adequate training samples. Consequently, researchers are investigating techniques that remain efficient and accurate in sample-constrained environments. The application of few-shot techniques to NIDS has garnered significant interest. A notable achievement of 78.26% accuracy in a 5-way 1-shot scenario was reached by transforming principal static features into a two-dimensional matrix format and using only static features, marking substantial progress in few-shot environments[3]. However, this method did not fully exploit the internal traffic information and heavily relied on expert knowledge for feature selection. Another approach transformed raw network traffic into traffic images and processed the internal structure, making significant strides[4]. However, the sample cropping and selection during processing might lead to the omission of some critical features, thereby limiting the full utilization of the data. To address these shortcomings, we propose a few-shot NIDS that leverages multimodal fusion technology. This system integrates the dynamic structure of traffic, represented by feature graphs, with static features from network feature sets, through advanced multimodal feature extraction and fusion techniques. This integration of dynamic traffic feature maps and static network feature sets allows the system to fully exploit the interrelationships among data,

[1]College of Information Science and Engineering, Jiaxing University, Jiaxing, Zhejiang, China. [2]School of Electrical and Information Engineering, Tianjin University, Tianjin, China. [3]School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, Zhejiang, China. ✉email: cyxu@zjxu.edu.cn

enhancing the network security system's ability to tackle complex and variable threats. The combined approach not only preserves the temporal dynamics of network traffic but also leverages contextual information from static features, providing a comprehensive view that is crucial for identifying and mitigating sophisticated cyber threats.

Traditional network intrusion detection models typically rely on voluminous data and adhere strictly to known attack patterns or signatures. Yet, as data becomes more complex and attacks more diverse, DNNs have gradually supplanted traditional methods. However, they still depend heavily on large datasets. In security-sensitive environments, securing sufficient samples is neither safe nor feasible. Contemporary few-shot NIDS solutions are broadly divided into two categories: those that utilize raw traffic and those that employ structured files post-feature extraction. For raw traffic, whether dealing with flows or packets[5], despite its format regularity and not requiring expert input for feature extraction decisions, the scalability limitations in data processing frequently lead to cropping and selection issues, causing the loss or diminishment of crucial information, such as packet counts and lengths. Conversely, structured files post-feature extraction, although more readable and smaller in scale, are extensively utilized[6], but often lack the detailed content of raw traffic, such as the positional relationships among packets and the temporal relations within traffic flows. In our study, we amalgamate the benefits of raw traffic and structured files, and through sophisticated feature fusion techniques, we remedy their deficiencies. We proposed an integrated system employing multimodal fusion technology, generating two modal data types through heterogeneous data: traffic feature graphs and network feature sets. Traffic feature graphs depict the intricate features of traffic content, with an emphasis on the spatial relationships among internal data packets, processed using convolutional neural networks. For the network feature set, we classify features into discrete and continuous categories and process them separately. Employing the multi-head attention mechanism of Transformer, we adeptly integrate and process both sets of features. Ultimately, we fuse the outputs from both modalities, conducting experiments across various interaction depths, and achieving optimal outcomes under deep self-attention interaction fusion. Additionally, we implement transfer enhancement strategies to further improve the performance of few-shot network intrusion detection. The principal contributions of this work include the following points:

(1) A multimodal fusion based detection system is proposed that combines the spatial feature extraction capabilities of convolutional neural networks (CNNs) with the global information capture of the Transformer's multi-head attention mechanism.
(2) Three multimodal fusion strategies are presented that enhance data representation by extracting and merging features and employing data heterogeneity techniques to generate multimodal data.
(3) The results of a performance evaluation on the benchmark CICIDS2017 and CICIDS2018 datasets reveal accuracies of 93.40% and 98.50%, surpassing existing methods.

The remainder of this paper is organized as follows: "Related work" introduces the recent related work. Section "Multimodal feature fusion detection method" describes the multimodal feature fusion detection method. Section "Experiments" describes the experiments, and "Discussion" presents a comparison and discussion. Section "Future work" outlines our future work. Finally, "Conclusion" concludes the paper.

Related work

This section briefly reviews recent studies published on deep learning-based network intrusion detection models, malicious traffic detection methods for few-shot learning, and the application of multimodal feature fusion techniques. We will explore in detail the contributions of these studies in enhancing efficiency and innovation within the field of cybersecurity. Additionally, we will examine the advantages these methods display in practical applications and their potential in addressing complex issues within cybersecurity.

Network intrusion detection

In recent years, deep learning-based methods have been widely applied in network intrusion detection, significantly improving the detection effectiveness. As these algorithms continue to evolve and become more sophisticated, increasingly efficient NIDSs are being developed to combat the growing complexity of cybersecurity threats effectively. Souradip et al.[7] optimized several aspects, including removing multicollinearity, sampling, and dimension reduction, thereby enabling the effective detection of network attacks and anomalies in resource-constrained environments. Ankit et al.[8] introduced techniques that combine autoencoding and principal component analysis (PCA) to capture both the linear and nonlinear relationships between features, reduce data dimensionality, and offer new solutions for managing complex network traffic data. Jafar et al.[9] developed a hybrid deep-learning framework that integrates the strengths of CNNs and long short-term memory (LSTM) networks to enhance the detection rates. Maya et al.[10] combined bootstrapping aggregation with gradient boosting decision trees using a dual ensemble model. Zakieh et al.[11] enhanced the African vultures optimization algorithm with the sine cosine algorithm to avoid local optima and enhance the global search capabilities. Liu et al.[12] highlighted the limitations of popular deep-learning algorithms in terms of accuracy and dependency on manually selected features and proposed an enhanced empirical component analysis approach that combines empirical mode decomposition with PCA, retaining the most relevant features and classifying attack nodes using LSTM. Karima et al.[13] designed an end-to-end one-dimensional (1D) CNN model tailored to detect complex threats in Industrial IoT (IIoT) environments, which exhibited excellent performance on the Edge-IIoT set dataset through k-fold cross-validation. Miel et al.[14] introduced a new multistage hierarchical intrusion detection method that can detect unknown zero-day attacks and is easy to deploy. MD et al.[15] developed a two-stage intrusion detection system using the generalized mean grey wolf optimizer and ElasticNet shrinkage autoencoders for feature selection, significantly improving the attack classification accuracy. Vladimir et al.[16] discussed how most current methods are based on datasets spanning at least five years, leading to unclear security performance insights. They proposed a modular network intrusion detection architecture that can simulate real-world network attacks and assess their defense capabilities. Ratul et al.[17] focused on real-time traffic detection and proposed a feature selection method that combines particle swarm optimization with genetic algorithms to establish a two-stage NIDS. Nguyen et al.[18] introduced a traffic-aware self-supervised learning method known as TS-IDS for IoT NIDSs aimed at capturing traffic relationships between network entities. Their experiments on the real-world NF-ToN-IoT and NF-BoT-IoT datasets demonstrated the model's potential to enhance the detection performance and work effectively even without labeled data, outperforming state-of-the-art baseline models. Amir et al.[19] proposed a novel lightweight structure based on parallel deep autoencoders that leverages local and surrounding information in feature vectors. This type of feature separation enables the model to improve the accuracy while significantly reducing the number of parameters, memory usage, and processing power requirements.

Few-shot detection

Few-shot learning has gained widespread application in the field of network intrusion detection to mimic rapid and flexible human-like learning capabilities and adapt to modern network environments. The objective of few-shot learning is to develop a model that operates effectively with minimal labeled data. Marija et al.[20] demonstrated that coarse-grained processing of fewer feature types is crucial, with their method accurately detecting attacks with only three instances. Radhika et al.[21] addressed the imbalance between different attack categories, which diminishes the learning performance of machine-learning models for malicious traffic, by introducing a regularized Wasserstein generative adversarial network (WGAN) to balance the dataset by augmenting minority attack samples. Their enhanced WGAN-IDR performed better than other augmentation techniques. Danish et al.[22] emphasized prioritizing key elements in datasets and allocating more computational resources to segments that are likely to contain patterns or anomalies indicative of security threats. This approach, combined with Bi-LSTM, improved the ability of the detection system to learn effectively from limited datasets by integrating the Shapley additive explanation (SHAP)mechanism to enhance the transparency, credibility, and interpretability of the system. Xiao et al.[23] argued that existing network intrusion detection methods rely heavily on traditional machine-learning or deep-learning techniques based on the statistical characteristics of network flows that can only be extracted after flow termination, thereby delaying intrusion detection. To address this issue, they proposed a detection method based on graph embedding techniques. Their approach classifies graph vectors using random forests, automatically extracts flow graph features using subgraph structures, and relies on only a few initial packets of each bidirectional network flow. Mohamed et al.[24] developed an intelligent hybrid model using machine learning and artificial intelligence with feature reduction techniques, including singular value decomposition, PCA, and a k-means clustering model, to enhance the information gain, thereby ensuring high accuracy and reliability of the extracted features. Nan et al.[25] proposed a malicious traffic detection model based on feature enhancement for small unbalanced datasets, grouping the original traffic features using Gaussian eigenvalues and generating clustering features using the k-means algorithm. This dual classification model, which was built on shallow neural networks and random forests, was used for network traffic detection. Ankit et al.[26] used a bagging classifier to address class imbalance issues, employing deep neural networks (DNNs) as base estimators to achieve generalization while handling the class imbalance in intrusion detection datasets with dual benefits and advantages. Rajkumar et al.[27] focused on data generation, starting with data preprocessing to enhance the quality of the training data. They then used adaptive synthetic oversampling techniques to generate minority class samples to overcome the class imbalance. Finally, they incorporated SHAP feature importance into the recursive feature elimination across five base classifiers for feature selection, and input these features into a dynamic ensemble selection technique that classifies by varying the k-value.

Multimodal feature fusion

Multimodal feature fusion techniques aim to combine data or features from different sources to enhance the decision-making accuracy in systems. The integration of information from multiple data sources can effectively identify and respond to potential network threats during network intrusion detection. However, practical applications often encounter challenges owing to data processing complexities and insufficient fusion efficiency. Ren-Hung et al.[28] evaluated three host-based data sources, namely network traffic, system logs, and host metrics, to assess their combined detection capabilities across various attack stages and types. Network traffic data were processed using CNNs for improved automatic feature selection, system log data were handled with LSTMs and attention models to enhance the temporal relationship exploration, and host metrics were processed through DNNs, thereby enhancing the model performance through the detailed handling of diverse data types. Ankit et al.[29] designed a DNN-based IDS that used a statistical significance fusion of the standard deviation and the difference between the mean and median for feature selection, with the aim of filtering out highly discriminative and biased relevant features for more effective learning. Liu et al.[30] proposed a multitask deep-learning intrusion detection approach that combines anomaly detection, clustering, and classification, effectively addressing attack detection and class distribution imbalances in network traffic. Juan et al.[31] introduced a feature fusion technique that is enhanced by gradient importance, combining feature fusion and enhancement to make the models focus more on classification-relevant sample features. Hong et al.[32] initially designed an adversarial sample generation algorithm to assess the performance of IoT network intrusion detectors and proposed a new framework that defends against adversarial attacks through feature grouping and multimodel fusion, thereby contributing significantly to IoT network intrusion detection. Xiao et al.[33] focused on fusing heterogeneous threat intelligence from security information and event management systems to reconstruct multistep attack scenarios and identify key attack paths. They formatted structured threat information to express heterogeneous threat intelligence cohesively, using semantic association weights and community detection algorithms to mine attack scenarios.

Multimodal feature fusion detection method

This section provides a detailed introduction to our proposed multimodal feature fusion framework. We begin by discussing the data acquisition and preprocessing steps to ensure data quality and consistency, which are crucial for model training and system detection performance. Next, we demonstrate how heterogeneous data is generated to form multimodal datasets and explain how our framework implements comprehensive data analysis and feature fusion.

Data processing

To ensure the effectiveness of model training and the accuracy of system detection, this paper utilizes the raw pcap data from CICIDS2017 and CICIDS2018 provided by the Canadian Institute for Cybersecurity (CIC) as experimental datasets. These two datasets include benign network traffic and various common attack traffic, adhering to real-world standards to ensure the broad applicability and reproducibility of the datasets. Additionally, during the preprocessing phase, we rigorously align the network traffic data to generate corresponding multimodal data, thereby providing a solid foundation for subsequent analysis and detection.

Network traffic sources

Obtaining high-quality and representative network traffic data is crucial for network intrusion detection because it directly influences the model training and system detection performance. Common techniques such as packet capturing, port mirroring, traffic redirection, and network probes are utilized to capture raw traffic data from various network environments. These methods provide a foundation for model training and validation, ensuring the broad adaptability of data processing and reproducibility of experiments, and enhancing the universality and transferability of research findings.

Preprocessing and labeling

Raw traffic is first scrutinized to remove incomplete, incorrect, or anomalous records, thereby ensuring data quality and consistency. The cleaned traffic is then finely segmented by quintuple information (source IP address, source port, destination IP address, destination port, and transport-layer protocol), producing individual flow pcaps. Each flow pcap is matched against the official CSV file by comparing the same quintuple fields, and labels are assigned based on the corresponding columns in the CSV. In cases where the official CSV does not include complete quintuple information and direct matching is not possible, we employ CICFlowMeter to extract flow features and segment the original pcap into flows; these flows are then aligned to CSV entries using protocol identifiers and timestamps to assign labels, and the remaining processing proceeds as in the first approach.

Multimodal data generation

For labeled data, multimodal data are generated through data heterogeneity while ensuring the anonymization of sensitive information. Based on the research by Xu et al.[34] and Wang et al.[35], the model's performance is excellent when the number of packets ranges from 6 to 30 and the number of bytes per packet ranges from 100 to 200. Therefore, this study selects the first 16 packets and the first 256 bytes as input data, which is considered sufficient and adequate. Initially, the first 256 bytes from the first 16 packets of each network flow are extracted, and the four-tuple information (source IP, destination IP, source port, destination port) is anonymized to mitigate biases stemming from IP address variations. This anonymization process preserves the structural integrity of the data while safeguarding privacy and reducing skew in the analysis. Subsequently, these bytes are transformed into a traffic feature graph (Modality 1). For ease of loading and processing, the selected 16 packets of 256 bytes each are sequentially reconstructed into a 64 × 64 8-bit grayscale image. Although this may lead to some loss of original data, it enhances the detection timeliness and reduces the sample size. In addition, detailed traffic feature sets (Modality 2), including the total packet numbers and flow duration, are extracted from the complete flow for in-depth analysis and model training[36]. These features, which are shown in Table 1, include continuous features such as total traffic and discrete features such as TCP flag combinations to explore specific distribution combinations and TCP transmission states. For convenience in downstream processing, discrete combination features are one-hot encoded, and all features are saved in CSV format. To maintain one-to-one correspondence with the grayscale images, we add an additional id column to the CSV, with each id matching its corresponding image filename, thereby ensuring that every row in the CSV aligns with the correct image and preventing any loss or mismatch during data cleaning.

Network feature set composition		
Number of Packets	Packet Size Distribution_3	TCP Flags Count_SA
Total Traffic	Packet Size Distribution_4	TCP Flags Count_S
Duration	Packet Size Distribution_5	TCP Flags Count_R
Average Packet Size	Packet Size Distribution_6	TCP Flags Count_PA
Max Packet Size	Inter Arrival Time Mean	Packet Size Distribution Combination
Min Packet Size	Inter Arrival Time Variance	TCP Flags Combination
Packet Size Distribution_1	TCP Flags Count_A	
Packet Size Distribution_2	TCP Flags Count_FA	

Table 1. Network feature set composition.

Multimodal fusion framework

The proposed few-shot network intrusion detection system leverages multimodal feature fusion, where original traffic data is transformed into two types of data through data heterogeneity. Different data types reveal distinct characteristics of network traffic. Traffic feature graphs primarily contain spatial distribution and temporal sequence information, suitable for capturing the spatial correlations of traffic patterns; whereas the network feature set includes more abstract network information such as packet counts and traffic length, suitable for analyzing high-level features of network behavior. To effectively process these two modalities of data, we have designed two feature extraction models: the Traffic Feature Graph Feature Extraction Model (G-Model) and the Network Feature Set Feature Extraction Model (S-Model). The G-Model employs a CNN architecture to analyze and extract local patterns and spatial dependencies in traffic features. The S-Model uses a Transformer-based architecture, which excels in handling long sequence data and can capture long-range dependencies in the network feature set, followed by feature fusion. Furthermore, transfer learning is introduced to adapt to few-shot scenarios. The overall framework is illustrated in Fig. 1.

The multimodal fusion framework, as shown in Algorithm 1, starts by loading a batch of traffic feature graphs $\mathscr{G}_T$ and network feature sets $\mathscr{S}_F$. For $\mathscr{G}_T$, it undergoes convolution processing in the Upper Layers before being flattened and passed to the Lower Layers for further processing, as detailed in lines 8 to 9. $\mathscr{S}_F$ is processed through steps, as detailed in lines 11 to 13. Finally, feature vectors $\mathscr{y}_s$ and $\mathscr{y}_g$ are obtained from the S-Model and G-Model, respectively, and feature fusion is carried out in line 15 as $\Phi(\cdot)$. For the Transfer-Enhanced Model, a transition from $\mathscr{D}_1$ to $\mathscr{D}_2$ is required, along with fine-tuning of the model, as depicted in lines 20 to 24, culminating in the output of the classification results.

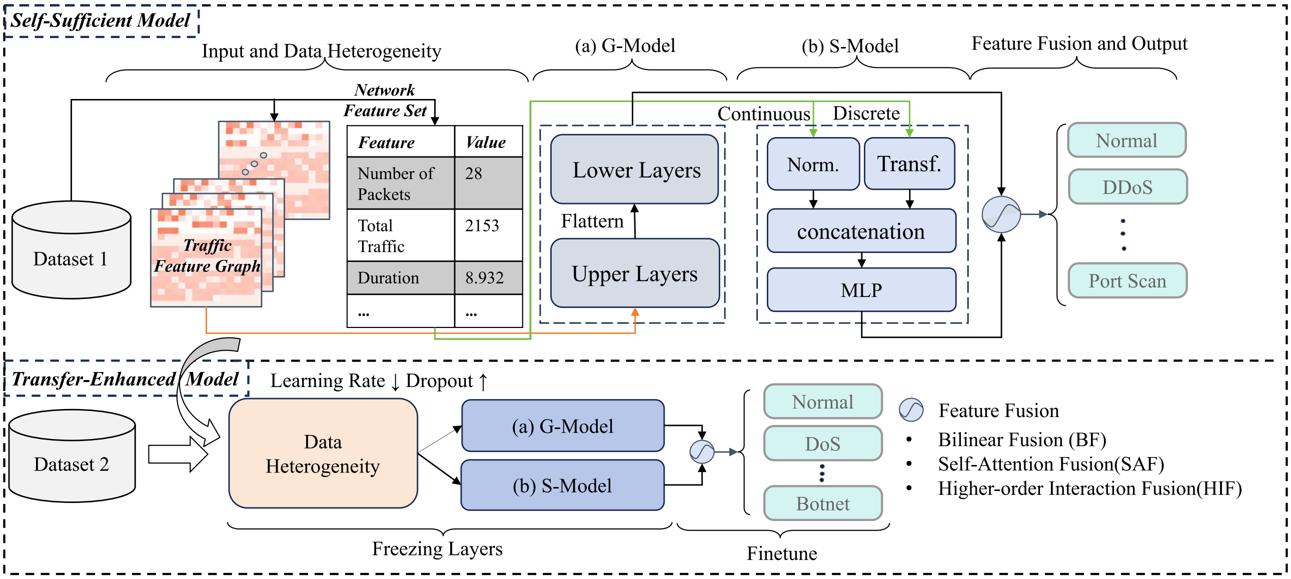


Figure 1. Overview of the multimodal fusion framework. The Self-Sufficient Model generates features from heterogeneous data sources through the G-Model and S-Model for feature extraction, followed by multimodal fusion and classification. The Transfer-Enhanced Model inherits and freezes these feature extractors, focusing on fine-tuning the multimodal fusion component.

```
Input      :Datasets $\mathscr{D}_1$, $\mathscr{D}_2$
Output     :Classification $\mathscr{C}$
Initialize :models $\mathscr{G}$-Model and $\mathscr{S}$-Model with parameters $\theta_g$ and $\theta_s$; number of epochs $Epochs$
Set learning rate $\eta$, dropout rate $\delta$;
for $epoch$=1,2,... $Epochs$ do
    foreach $batch$ in $\mathscr{D}_1$ do
        // Stage 1:  Self-Sufficient Model
        $\mathscr{G}_T \leftarrow$ Traffic Feature Graph($batch$);
        $\mathscr{S}_F \leftarrow$ Network Feature Set($batch$);
        // Feature Extraction with $\mathscr{G}$-Model
        $y_{upper} \leftarrow$ UpperLayers($\mathscr{G}_T, \theta_g$);
        $y_g \leftarrow$ LowerLayers(Flatten($y_{upper}$), $\theta_g$);
        // Feature Extraction with $\mathscr{S}$-Model
        $\mathscr{F}_c, \mathscr{F}_d \leftarrow$ SeparateFeatures($\mathscr{S}_F$);
        $x \leftarrow \text{Concat}(\text{Norm.}(\mathscr{F}_c, \theta_s), \text{Transf.}(\mathscr{F}_d, \theta_s))$;
        $y_s \leftarrow \text{MLP}(x, \theta_s)$;
        // Feature Fusion and Classification
        $z \leftarrow \Phi(y_g, y_s; \text{BF}, \text{SAF}, \text{HIF})$;
        Classify $z$ into $\{\text{Normal}, \text{DDoS}, \text{Port Scan}, \ldots\}$;
    end for
end for
// Stage 2:  Transfer-Enhanced Model
Transfer features $z$ from $\mathscr{D}_1$ to $\mathscr{D}_2$, considering data heterogeneity $\mathscr{H}$;
Freeze layers in $\mathscr{G}$-Model and $\mathscr{S}$-Model as per transfer learning strategy $\mathscr{L}$;
Fine-tune feature fusion using modified learning rate $\eta'$ and dropout $\delta'$;
Apply the same advanced feature fusion techniques $\Phi(\cdot)$;
Classify refined features into categories $\mathscr{C}$ : $\{\text{Normal}, \text{DoS}, \text{Botnet}, \ldots\}$;
```

Algorithm 1. Multimodal fusion framework

Data heterogeneity

To capture and analyze the complexity and diversity of network traffic thoroughly, this study introduces two distinct data representation methods: the traffic feature graph and network feature set. These methods aim to delve into the network activities from different dimensions.

Traffic Feature Graph: This representation focuses on the intrinsic connections among network packets by sequentially selecting the first 16 packets of each flow and stacking their contents. This micro-level view helps our G-Model to identify potential anomalies hidden within single or consecutive packets.

Network Feature Set: From a macro perspective, this method synthesizes key continuous features such as the packet count, total traffic, and session duration, along with key discrete features such as the packet size distribution and TCP flag distribution combinations. These data serve as inputs for the S-Model, enabling it to detect potential threats from broader network behaviors.

Each network flow corresponds to a unique traffic feature graph and network feature set. During data processing, we randomly shuffle the dataset to ensure a balanced data distribution. We then iterate through the shuffled dataset and load the corresponding two modalities of data: the network feature set, which comprises 21 continuous features and 2 discrete features as structured data; and the traffic feature graph, which is a $16 \times 16 \times 16$ tensor reconstructed from the first 256 bytes of 16 pcap files. By employing this multimodal approach, the aim is to analyze network traffic at multiple levels, leveraging the complementary strengths of both models to enhance the detection comprehensiveness and accuracy.

G-Model

The G-Model is designed to process data from Modality 1 and the specific schematic of the framework, comprising an arrangement of the upper and lower layers, is shown in Fig. 2. The Upper Layers, specifically the first and second layers, as detailed in Table 2, utilize a three-dimensional (3D) convolution (Conv3d) with parameters indicating the input channels, output channels, number of kernels, stride, and padding. BatchNorm3d

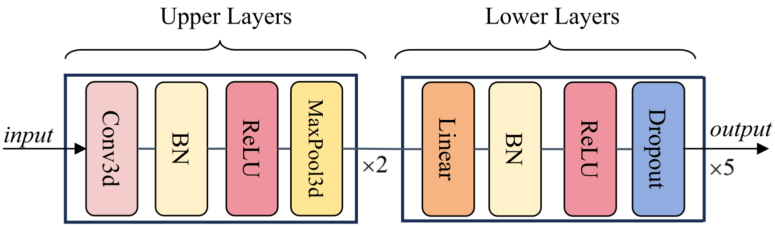


Figure 2. Schematic of the G-Model framework.

Layers	Parameters	Output dimensions
Conv3d-1	(1,32,3,1,1)	[− 1,32,16,16,16]
Conv3d-2	(32,64,3,1,1)	[− 1,64,8,8,8]
BatchNorm3d-1	(32)	[− 1,32,16,16,16]
BatchNorm3d-2	(64)	[− 1,64,8,8,8]
ReLU-1	(N/A)	[− 1,32,16,16,16]
ReLU-2	(N/A)	[− 1,64,8,8,8]
MaxPool3d-1	(2,2)	[− 1,32,8,8,8]
MaxPool3d-2	(2,2)	[− 1,64,4,4,4]
Flatten	(N/A)	[− 1,4096]

Table 2. Parameters and output dimensions of the Upper Layers in the G-Model.

Layers	Parameters	Output dimensions
Linear-3	(4096,2048)	[− 1,2048]
Linear-4	(2048,1024)	[− 1,1024]
Linear-5	(1024,512)	[− 1,512]
Linear-6	(512,256)	[− 1,256]
Linear-7	(256,128)	[− 1,128]
BatchNorm1d-3	(2048)	[− 1,2048]
BatchNorm1d-4	(1024)	[− 1,1024]
BatchNorm1d-5	(512)	[− 1,512]
BatchNorm1d-6	(256)	[− 1,256]
BatchNorm1d-7	(128)	[− 1,128]
ReLU	(N/A)	Same as above
Dropout	(0.1)	Same as above

Table 3. Parameters and output dimensions of the lower layers in the G-Model.

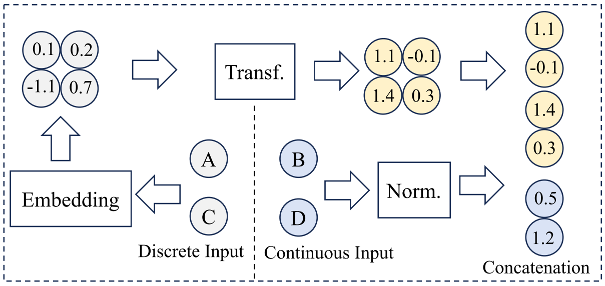


Figure 3. Schematic of the S-Model framework.

normalizes the specified number of feature channels. The MaxPool3d parameters define the number of kernels and stride. The flattened layer without learnable parameters converts the input dimensions into a 1D vector.

This 1D vector is then fed into the Lower Layers (layers 3 to 7), as detailed in Table 3, where the linear layers (dense layers) transform the input with specific dimensions, such as 4096 to 2048. The dropout layer, set to a rate of 0.1, randomly nullifies 10% of its inputs during training to prevent overfitting.

S-Model

The S-Model processes data from Modality 2 and handles continuous and discrete data separately, as shown in Fig. 3. First, discrete inputs are processed through an embedding layer that graphs discrete values to a continuous, high-dimensional space. Each categorical feature has a separate embedding matrix sized as the product of the number of categories and output dimension. The tensors from the embedding layer then enter a Transformer[37] block (Transf), as shown in Fig. 4, which includes self-attention, residual connections, normalization, fully connected feedforward layers, and multiple stackable layers based on the task requirements.

This model focuses on the relationships between different parts of the input features. Each encoder block has multiple self-attention heads that allow the model to learn the feature relationships in parallel subspaces. For continuous inputs, the initial processing step involves normalization (Norm), where the mean of each feature

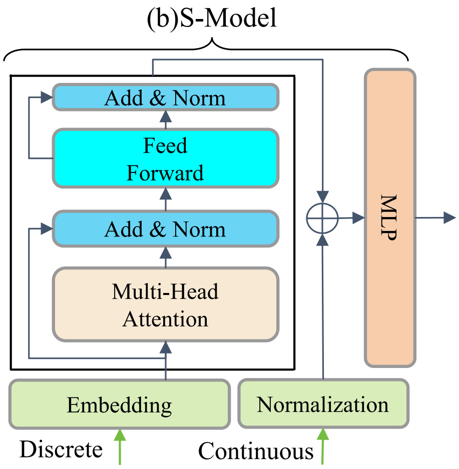


Figure 4. Processing flowchart for discrete and continuous features in the S-Model.

Layer	Input shape	Output shape	Description
Embedding	[− 1, 2]	[− 1, 2, 128]	2 discrete features are embedded into 128 dimensions.
Shared	[− 1, 2]	[− 1, 2, 112]	Captures common relationships among categorical values.
Independent	[− 1, 2]	[− 1, 2, 16]	Captures unique characteristics of each categorical value.
Transformer	[− 1, 2, 128]	[− 1, 2, 128]	128 represents 112 independent and 16 shared embeddings.
Multi– head Attention	[− 1, 2, 128]	[− 1, 2, 128]	Performs multi-head attention processing.
Feed Forward	[− 1, 2, 128]	[− 1, 2, 128]	Performs feed-forward neural network processing.
Layer Norm	[− 1, 21]	[− 1, 21]	21 continuous contents pass through Layer Norm.
MLP	[− 1, 277]	[− 1, 128]	277 represents all input embeddings, 2*128 + 21 inputs.

Table 4. Layer structure with input and output shapes and descriptions for the S-model.

is subtracted and then divided by its standard deviation, as illustrated in Equation (1), where X represents the original data,μ is the mean of the original data,σ is the standard deviation, and X' is the data post-standardization. This step helps to accelerate the training process and improves the performance of the model. Standardized continuous features are then concatenated with discrete features processed using Transf, forming a hybrid input from both modalities.

$$X' = \frac{X - \mu}{\sigma} \tag{1}$$

Specifically, we first categorize the input data into discrete and continuous features based on their attribute types. For continuous features, we compute their mean and variance for standardization; for discrete features, we embed them into predetermined feature dimensions, as shown in Table 4. These feature dimensions comprise shared embeddings and independent embeddings. Considering that a single column may contain multiple categorical values, which often share common or structural relationships, shared embeddings of the same type are maintained consistently. These shared embedding vectors are randomly initialized and serve as trainable parameters of the model. For the Transformer layer input with a dimension of 128, it is formed by concatenating 16 shared embeddings with 112 independent embeddings. The concatenated feature sequence is then fed into the Multi-Attention and Feed Forward modules, both of which utilize the standard Transformer architecture and are stackable.

For continuous features, taking into account the continuity of their distribution, we apply Layer Normalization solely to the standardized data and then concatenate the normalized output with the discrete feature vectors. Here, the 277-dimensional vector represents the concatenation of two expanded 128-dimensional continuous features and 21 continuous features, resulting in the final 277-dimensional feature representation. Finally, dimensionality reduction is performed through a Multi-Layer Perceptron (MLP). The MLP structure is also stackable, with each continuous hidden layer's size being four and two times that of the previous layer, respectively. The input dimensions sequentially change as [277, 1108, 2216, 128], ultimately reducing to 128 dimensions, consistent with the output feature dimension of the G-Model.

Feature fusion methods

In this study, we delve into the fusion effects of different modal data at various interaction depths to maximize the information quantity and quality extracted from limited samples. Specifically, we have designed three feature fusion methods, each corresponding to different interaction depths to enhance the model's capability to recognize complex network attacks: bilinear fusion (BF), self-attention fusion (SAF), and higher-order interaction fusion (HIF). Let $\alpha \in \mathbb{R}^{d_1}$ represent the vector processed by the G-Model and $\beta \in \mathbb{R}^{d_1}$ represent the vector processed by the S-Model, where d_1 and d_2 are the output dimensions, with the default setting $d_1 = d_2 = d$.

(a) BF This method adopts a bilinear form to integrate features from two modalities. For each output dimension *i*, a unique weight matrix $W^{(i)} \in \mathbb{R}^{d_1 \times d_2}$ is specifically designed to handle the pairwise product of corresponding elements in the two input feature vectors. Since it only considers pairwise interactions and does not involve higher-order or deeper relationships, it can be viewed as a single-layer deep interaction. In our framework, BF serves as a fundamental approach by directly using a bilinear mapping to combine features from the two modalities. Let $\alpha \in \mathbb{R}^{d_1}$ and $\beta \in \mathbb{R}^{d_2}$ be the feature vectors of the two modalities, respectively. For each output dimension *i*, the bilinear interaction is computed via the distinct weight matrix $W^{(i)}$. An activation function $\phi(\cdot)$(e.g., ReLU) can be applied to enhance nonlinearity. The calculation is given by

$$output_i = \phi\Big(\alpha^T\, W^{(i)}\, \beta + b_i\Big), \tag{2}$$

where b_i is the bias term, and the final output vector has dimension *o*. By merging the weight matrices of all output dimensions, we obtain a three-dimensional tensor $W \in \mathbb{R}^{o \times d_1 \times d_2}$.

(b) SAF SAF first concatenates the two modality vectors into $X \in \mathbb{R}^{d_1 + d_2}$, and then employs learnable matrices $W^Q, W^K, W^V \in \mathbb{R}^{(d_1+d_2)\times(d_1+d_2)}$ to project *X* into the query *Q*, key *K*, and value *V*. This self-attention mechanism can dynamically compute interaction weights among different features, thus enabling more flexible information integration. Its core formula is

$$output = softmax\Big(\frac{QK^T}{\sqrt{d_k}}\Big)\, V, \tag{3}$$

where $\sqrt{d_k}$ serves as a scaling factor to balance the magnitude of the dot product and stabilize gradients. To further enhance representation capability, multi-head attention is introduced. It concatenates multiple parallel self-attention heads and then applies a linear transformation, as defined by

$$MultiHead(Q, K, V) = Concat\Big(head_1, \ldots, head_h\Big)\, W_O, \tag{4}$$

where each attention head is computed as

$$head_i = softmax\Big(\frac{Q\, W_i^Q\, \big(K\, W_i^K\big)^T}{\sqrt{d_k}}\Big)\, \big(V\, W_i^V\big), \quad i = 1, \ldots, h. \tag{5}$$

Here W_i^Q, W_i^K, $W_i^V \in \mathbb{R}^{(d_1+d_2)\times d_k}$ are learnable matrices, and $W_O \in \mathbb{R}^{(h\, d_k)\times(d_1+d_2)}$ is used to map the concatenated heads to the output. This design allows the model to capture a variety of feature interactions from multiple subspaces.

(c) HIF Unlike first-order approaches that only consider original features and their linear combinations, high-order interactions explicitly introduce richer feature transformations such as elementwise squares and elementwise products (also known as Hadamard products). This strategy can yield more powerful representational capacity. In this method, let $\alpha \in \mathbb{R}^{d_1}$ and $\beta \in \mathbb{R}^{d_2}$ be the feature vectors from two modalities. HIF further includes the elementwise squares α^2, β^2 as well as the elementwise product $\alpha \odot \beta$. The elementwise Hadamard product can be defined as

$$(\alpha \odot \beta)_i = \alpha_i\, \beta_i, \tag{6}$$

which enriches the feature representation space. Therefore, the fused feature vector is

$$output = [\,\alpha,\ \beta,\ \alpha^2,\ \beta^2,\ \alpha \odot \beta\,]. \tag{7}$$

Flattening this concatenated result and feeding it into a learnable linear layer can balance the modeling of high-order dependencies and computational efficiency.

Type	Identifier	Description
Brute force	C1	Attempts to crack by trying multiple username and password combinations until successful
Port scan	C2	Scans various ports on a server to discover open ports and services
Web attack	C3	Attack targeting web applications, including SQL injection, command injection, etc
DDoS	C4	Distributed Denial of Service attacks making the target server or network resources unreachable
Normal	C5	Represents normal network traffic without malicious activities

Table 5. Types, identifiers, and descriptions of selected CICIDS2017 dataset.

Type	Identifier	Description
Botnet	D1	Steals personal information by capturing browser user records and forms
DDoS	D2	Distributed denial of service attacks making the target server or network resources unreachable
DoS	D3	Denial of service attacks targeting web applications, including SQL injection, command injection, etc
Brute Force	D4	Attempts to crack by trying multiple username and password combinations until successful
Normal	D5	Represents normal network traffic without malicious activities

Table 6. Types, identifiers, and descriptions of selected CICIDS2018 dataset.

Experiments

This section validates the practical effectiveness and efficiency of the proposed method through experiments conducted on two widely recognized network traffic datasets, CICIDS2017 and CICIDS2018. We utilized the pcap files from the aforementioned dataset as the raw data. For each network flow, we selected the first 16 pcap packets and the first 256 bytes of each packet, reconstructing them into a $16 \times 16 \times 16$ tensor to represent the traffic feature graph. Concurrently, we performed feature extraction on the pcap files as described in Table 1, resulting in the network feature set. To facilitate storage and usage, we saved the extracted network feature set in CSV format. We designed a series of experiments aimed at comprehensively evaluating the performance of the method using different evaluation metrics and hyperparameter settings. The experimental setup is divided into three main stages: first, the sample sensitivity experiments explore the model's performance in a few-shot environment to assess its accuracy and adaptability with limited data support; second, the feature fusion strategy experiments examine the effects of different levels of modality fusion strategies, aiming to identify the optimal feature fusion method to enhance the overall performance of the model; and finally, the model performance analysis and comparison investigate the advantages and disadvantages of the proposed model in terms of practicality and complexity by comparing it with traditional DNN methods.

In few-shot scenarios, the scarcity of data makes traditional training and testing ratio splits (such as 8:2) unsuitable. To address this, we introduce four predefined parameters: k_{train}, k_{test}, k_{source}, and n_{way}, to clearly define the absolute sizes of the training and testing datasets. Specifically, for the Self-Sufficient Model, the setup is as follows: from each class in the dataset, k_{train} samples are selected for training, covering n_{way} classes, resulting in a total of $k_{\text{train}} \times n_{\text{way}}$ training samples. Simultaneously, k_{test} samples are selected from each class for testing, also covering n_{way} classes, resulting in a total of $k_{\text{test}} \times n_{\text{way}}$ testing samples. For the Transfer-Enhanced Model, the training and testing data originate from different datasets (source domain and target domain). In this case, the source domain uses $k_{\text{source}} \times n_{\text{way}}$ samples for pre-training, while the target domain uses $k_{\text{train}} \times n_{\text{way}}$ samples for fine-tuning and $k_{\text{test}} \times n_{\text{way}}$ samples for testing. It is important to note that both k_{train} and k_{test} are defined per class rather than as overall sample counts. This definition ensures that the number of attack samples during testing and evaluation is both clear and limited, thereby effectively validating the correctness of the experiments and the robustness of the model under few-shot conditions.

This study used two widely recognized network traffic datasets, CICIDS2017 and CICIDS2018, which are provided by CIC. These datasets include both normal and malicious traffic, simulating real-world network environments to evaluate the performance of NIDSs effectively. We excluded attack types with extremely small sample sizes that could not ensure statistical stability and meet model training requirements. Specifically, in CICIDS2017, we excluded Infiltration and Botnet types. In CICIDS2018, we excluded the Web Attack type. The attack types utilized in subsequent experiments and their descriptions are shown in Tables 5 and 6.

Evaluation metrics

In this study, we used the following evaluation metrics to assess the model performance comprehensively: accuracy rate (ACC), precision rate (PR), detection rate (DR), and F1-score. Macro-averaging was introduced to provide a more precise evaluation of multiclass problems. The specific formula is shown in Equation (5), where N represents the total number of categories,C_{ii} represents the number of correct predictions,C_{ij} represents the total number of predictions, and TP_i, TN_i, FP_i, FN_i with the subscript *i* is used to distinguish between the different types.

$$\begin{cases} ACC = \frac{\sum_{i=1}^{N} C_{ii}}{\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}} \\ Precision_i = \frac{TP_i}{TP_i+FP_i} \\ Detection_i = \frac{TP_i}{TP_i+FN_i} \\ Macro\text{-}PR = \frac{1}{N}\sum_{i=1}^{N} Precision_i \\ Macro\text{-}DR = \frac{1}{N}\sum_{i=1}^{N} Detection_i \\ Macro\text{-}F1 = 2 \cdot \frac{Macro\text{-}PR \times Macro\text{-}DR}{Macro\text{-}PR + Macro\text{-}DR} \end{cases} \tag{8}$$

Hyperparameter settings

Table 7 details the baseline model architecture and its hyperparameter configurations. The parameter mlp_mults represents the multiplier factors used to determine the sizes of the hidden layers in the MLP, where (4, 2) indicates that each successive hidden layer is four and two times the size of the preceding one, respectively. Linear Fusion (LF) refers to the final feature fusion part that uses only a linear combination of vectors from the two modalities.

In the S-Model, heads indicate the number of heads in the multi-head attention mechanism, while depth refers to the number of layers to be stacked. fusion dim represents the dimension of the data after processing by the G-Model and S-Model, and epochs denote the total number of training rounds, noting that the mentioned epochs are default values. batch refers to the number of samples processed per batch. k_{train} and k_{test} represent the number of samples per type in the training and testing phases, respectively. Considering the concepts of source and target domains in transfer learning, the number of training samples used in the source domain is denoted as k_{source}. n_{way} specifies the number of categories in the classification problem.

Experimental setup

We designed three-phased experiments to evaluate and optimize our model thoroughly. Each phase examined the performance of different models within the detection system from various perspectives, identifying key strategies for improvement, and ultimately determining the best model combination.

Sample sensitivity

This experiment explored the sensitivity of the Self-Sufficient and Transfer-Enhanced models to different sample sizes. Using the CICIDS2017 and CICIDS2018 datasets, we aimed to evaluate the generalization and

Model architecture (stacking times/fusion method)	Hyperparameter	Value
Upper Layers G-Model (2)	Padding	1
Lower Layers G-Model (5)	Dropout	0.1
S-Model-Transf. (6)	Heads	4
	Dim	32
	Depth	6
	attn_dropout	0.1
	ff_dropout	0.1
S-Model-MLP	mlp_mults	(4, 2)
Self-sufficient model (LF)	lr	0.001
	Dropout	0.1
	Optimizer	adam
	Loss function	Cross entropy
Transfer-enhanced model (LF)	lr	0.0001
	Dropout	0.3
Common settings	Batch size	32
	Epochs	200
	Fusion dim	128
	k_{train}	[5, 10, 15]
	k_{test}	30
	k_{source}	100
	n_{way}	5

Table 7. Baseline architecture and hyperparameter configurations.

adaptability of the models using data collected at different times and in different network environments. This aids in understanding the performance of the models under various conditions, providing a basis for further optimization and application.

Feature fusion

In this section, we investigate the impact of three different feature fusion strategies on the model performance by comparing them with the baseline model. Comprehensive evaluations were conducted using the CICIDS2017 and CICIDS2018 datasets to determine the most suitable fusion strategy for the model framework.

Model performance analysis and comparison

We compared our model with common DNN models using multiple metrics including the accuracy, floating-point operations (FLOPs), and total parameters. These comparisons aimed to analyze the advantages of our model in terms of resource efficiency and performance relative to other methods, highlighting its potential application value.

Detection results

To investigate the impact of selecting an appropriate number of training epochs on model performance using the CICIDS2017 and CICIDS2018 datasets, we employed a Self-Sufficient Model under limited sample conditions and conducted experiments using the baseline model LF. We set K = 3 and $n_{\text{way}} = 5$, and used ACC from Equation (5) as the evaluation metric to demonstrate the model's performance convergence throughout the training process. As shown in Fig. 5, the model's ACC was approximately 20% within the first 25 training epochs, indicating that the model had not yet learned effective features for the five-class classification problem. Between 25 and 75 training epochs, the model's accuracy rapidly increased, and it stabilized with minimal fluctuations between 75 and 200 training epochs. Considering both model performance and training time, we set 200 training epochs as the initial configuration for subsequent experiments.

Sample sensitivity experiment

To investigate the sensitivity of the self-sufficient and Transfer-Enhanced models to the sample size, the Transfer-Enhanced Model was pre-trained with 100 samples on the baseline model. Specifically, if fine-tuning is required on CICIDS2017, the model is first pre-trained on CICIDS2018, and vice versa. We then determined the model accuracy for different sample sizes (K). The experimental results are shown in Figs. 6 and 7.

Figure 6 shows that the accuracy of the Self-Sufficient Model rapidly increased within 25 epochs, with the model quickly converging and exhibiting minor accuracy fluctuations. As the sample size increased, the accuracy on both the CICIDS2017 and CICIDS2018 datasets continued to improve. Similarly, the accuracy curve in Fig. 7 indicates that as the sample size increased, the overall accuracy of the Transfer-Enhanced Model also increased significantly. Additional metrics were used to provide a comprehensive evaluation, the results of which are presented in Tables 8 and 9.

The experimental results shown in Tables 8 and 9 indicate that Δ ACC represents the performance improvement of the Transfer-Enhanced model over the Self-Sufficient model for the corresponding K values. For the CICIDS2017 dataset, the performance of the Self-Sufficient model gradually improved as the value of K increased, with ACC rising from 0.912 to 0.947. Simultaneously, the values of Macro-DR, Macro-F1, and Macro-PR also increased, demonstrating the model's adaptability to the target dataset. The performance of the Transfer-Enhanced model was more complex. At K = 5, the ACC was 0.892, which is a 2.0% decrease compared to the Self-Sufficient model's 0.912. However, as K increased, the Transfer-Enhanced model's performance improved, with ACC rising to 0.951 at K = 10 and 0.960 at K = 15. Additionally, Macro-DR, Macro-F1, and Macro-PR

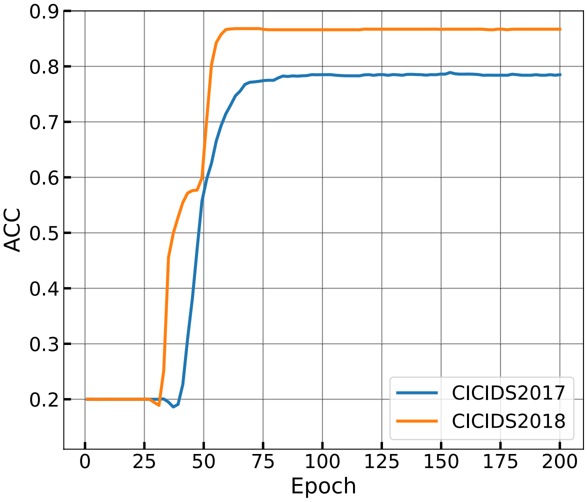
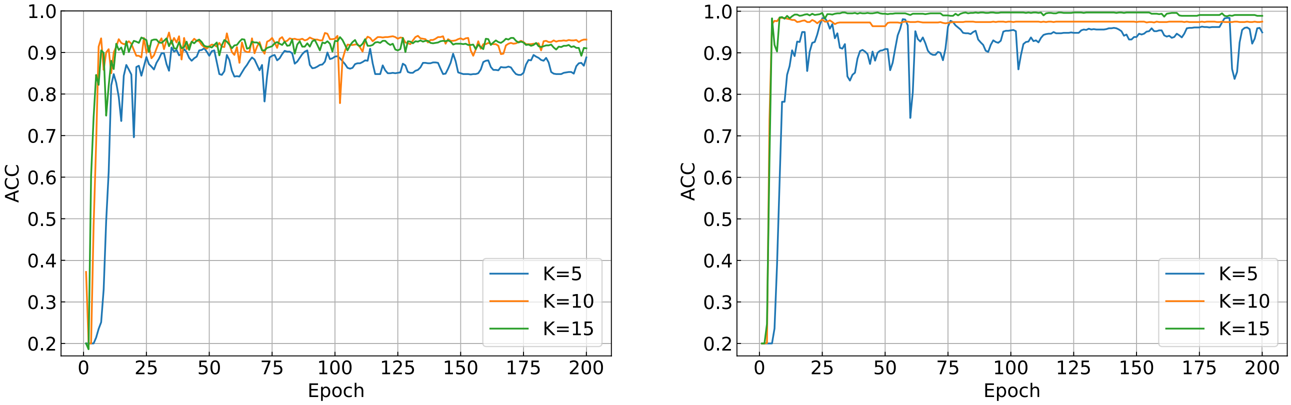


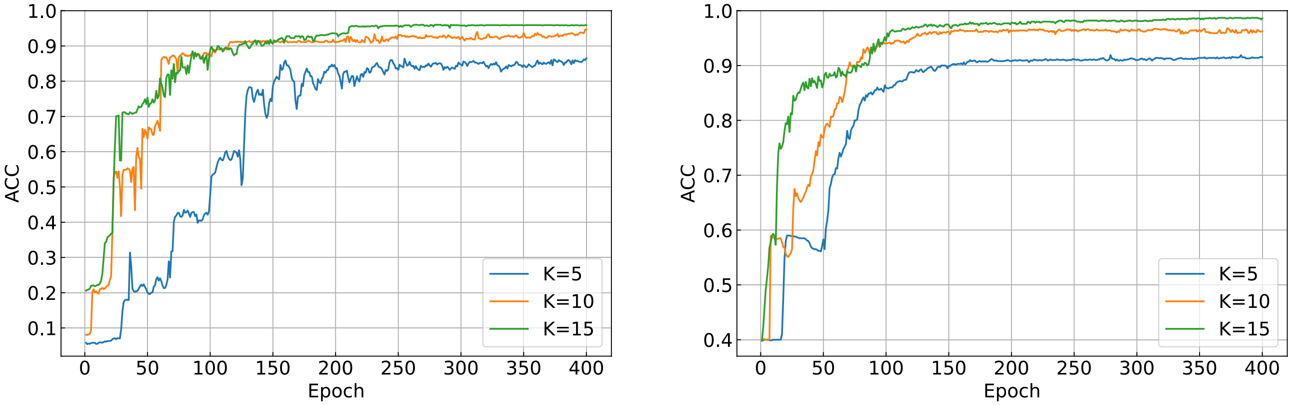
Figure 5. Performance of the Self-Sufficient Model on the CICIDS2017 and CICIDS2018 datasets.



(a) Performance on CICIDS2017.

(b) Performance on CICIDS2018.

Figure 6. Performance of the self-sufficient model on two datasets.



(a) Performance on CICIDS2017.

(b) Performance on CICIDS2018.

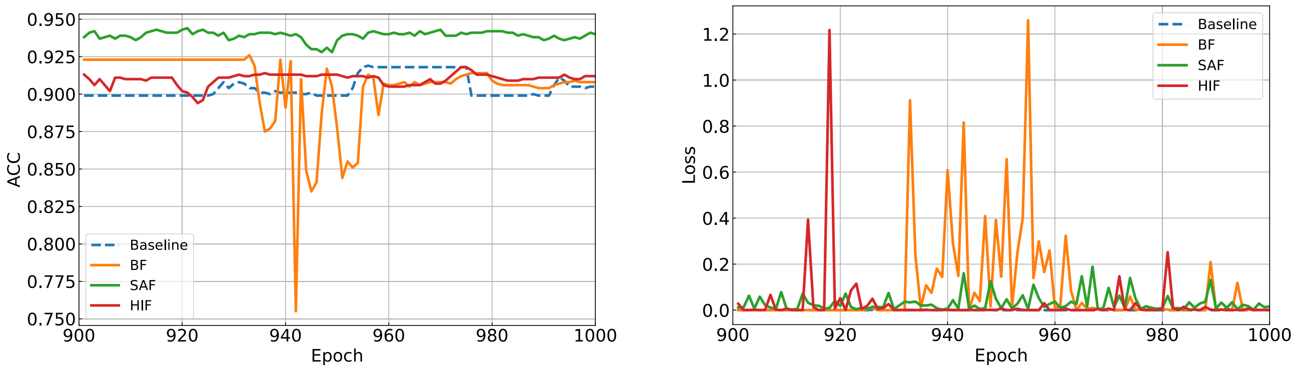
Figure 7. Performance of the transfer-enhanced model on two datasets.

Model	K	ACC	Δ ACC (%)	Macro-DR	Macro-F1	Macro-PR
Self-sufficient	5	0.912	–	0.9143	0.9138	0.9132
	10	0.938	–	0.9392	0.9399	0.9387
	15	0.947	–	0.9495	0.9491	0.9486
Transfer-enhanced	5	0.892	– 2.0% ↓	0.8840	0.8863	0.8865
	10	0.951	+1.3% ↑	0.9512	0.9506	0.9499
	15	0.960	+1.3% ↑	0.9631	0.9626	0.9621

Table 8. Performance of two models on the CICIDS2017 dataset.

Model	K	ACC	Δ ACC (%)	Macro-DR	Macro-F1	Macro-PR
Self-sufficient	5	0.984	–	0.9844	0.9843	0.9842
	10	0.984	–	0.9845	0.9844	0.9843
	15	0.997	–	0.9970	0.9970	0.9970
Transfer-enhanced	5	0.920	– 6.4% ↓	0.9242	0.9231	0.9221
	10	0.971	– 1.3% ↓	0.9712	0.9712	0.9711
	15	0.988	– 0.9% ↓	0.9881	0.9881	0.9880

Table 9. Performance of two models on the CICIDS2018 dataset.



(a) Accuracy variation curves of the three fusion methods.

(b) Loss variation curves of the three fusion methods.

Figure 8. Performance of three feature fusion methods on the CICIDS2017 dataset.

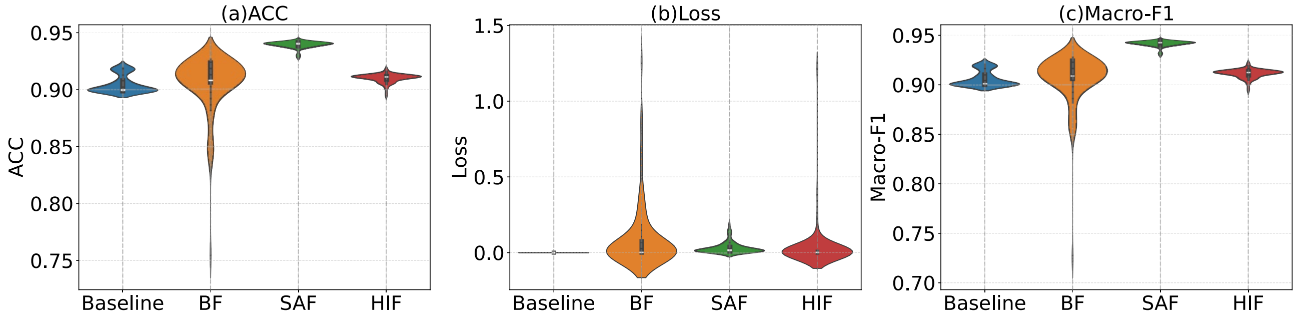


Figure 9. Performance of three feature fusion methods on various evaluation metrics.

also improved, suggesting that although the Transfer-Enhanced model performed worse in small sample sizes, its advantages began to show as the value of K increased, benefiting from pretraining. For the CICIDS2018 dataset, the Self-Sufficient model performed exceptionally well, especially at K = 15, where ACC reached 0.997, and Macro-DR, Macro-F1, and Macro-PR were all 0.9970. The Transfer-Enhanced model performed slightly worse, but still showed certain advantages. At K = 5, the ACC was 0.920, which was a 6.4% decrease compared to the Self-Sufficient model, and at K = 10, the ACC decreased by 1.3%. However, as K increased, the Transfer-Enhanced model's performance steadily improved, particularly at K = 15, where the ACC reached 0.988, closely approaching the performance of the Self-Sufficient model.

Overall, the Transfer-Enhanced model performed poorly at small sample sizes (K = 5), likely due to the discrepancy between source and target domain data, which hindered effective feature transfer during fine-tuning. However, as the sample size increased, the Transfer-Enhanced model gradually exhibited the advantages of pretraining, especially when the target domain dataset was larger, offering improved generalization. In contrast, the Self-Sufficient model showed more stable performance on the target dataset, particularly with fewer samples. However, its drawback lies in its potential inability to handle complex patterns and variations in the target dataset. The Self-Sufficient model may be more suitable when the data is limited or when the source and target domains are similar, while the Transfer-Enhanced model is better suited for scenarios with larger target domain data and significant differences between source and target domain distributions, leveraging the benefits of pretraining.

Feature fusion experiment

In this experiment, the accuracy and loss values of the final 100 epochs were collected to reflect the performance of the three feature fusion strategies compared with the baseline model. The results are shown in Fig. 8.The baseline model exhibited a low loss and an accuracy of approximately 90%. In contrast, BF and HIF exhibited greater loss fluctuations. The accuracy of SAF consistently remained above the baseline, SAF, and HIF, with loss values below 0.2.

In addition, we collected the performance results of these three methods over 100 epochs on the test set to explore the effectiveness of the various feature fusion methods, as shown in Fig. 9.

To reflect the performance of the models for the different types further, we repeated the experiments for 100 epochs for the three methods and accumulated the corresponding confusion matrices, as shown in Fig. 10. The figure reveals that the baseline model performed poorly on the DDoS and port scanning types, which is an issue that extended to BF. However, all fusion methods showed an improvement in ACC compared with the baseline model, with increases of 0.11%, 2.86%, and 0.54%, respectively, and SAF showed the largest improvement.

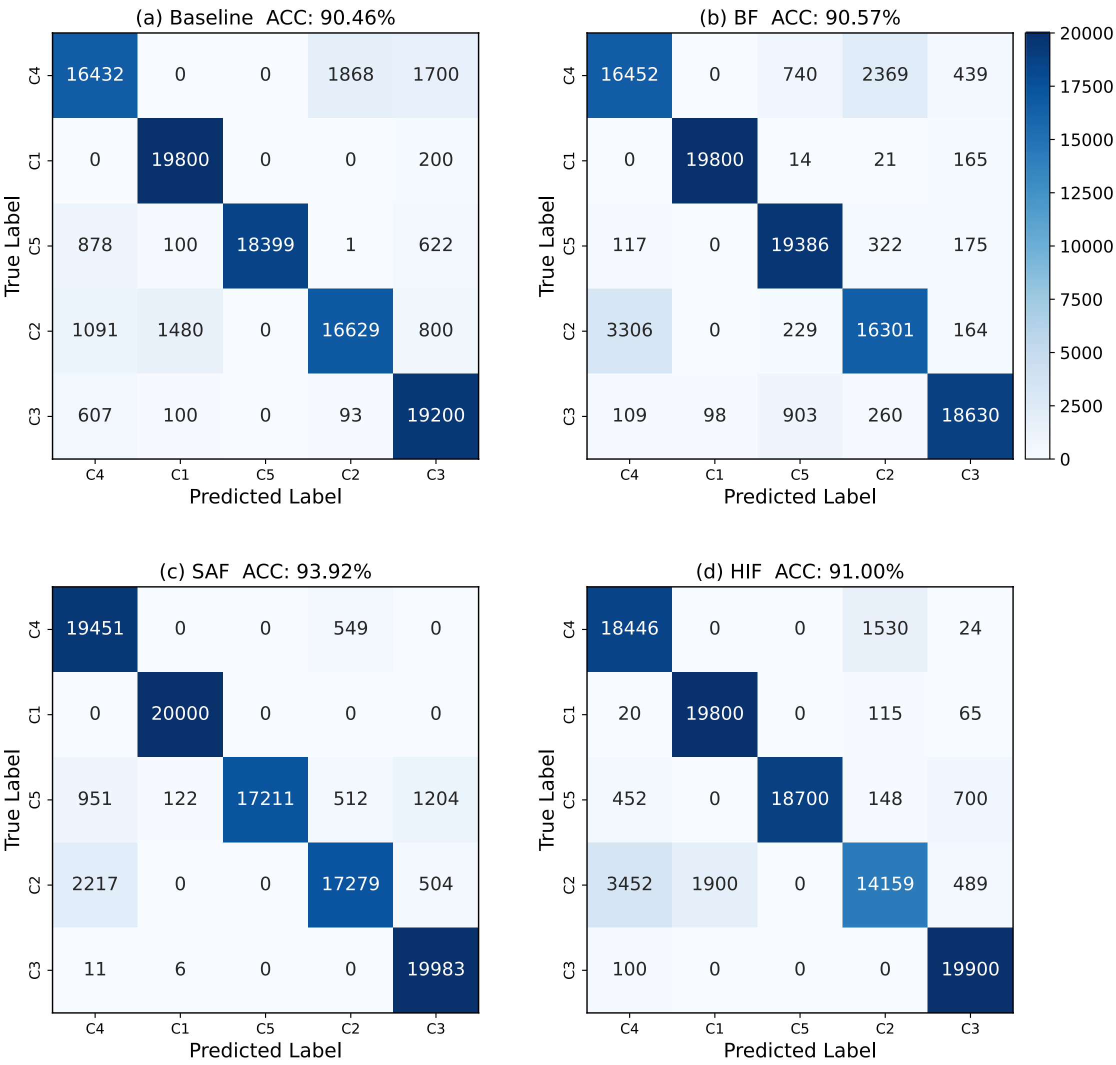


Figure 10. Confusion matrix of four feature fusion methods.

According to the detection results across various types, SAF effectively improved the detection of the DDoS and port scanning types, which were poorly handled by the baseline model. The experimental results indicate that SAF outperforms BF and HIF not only in overall accuracy but also in loss stability and its ability to detect DDoS and port scanning attacks. From these observations, we can infer that SAF is more robust in fusing different feature information, effectively mitigating redundancy and noise among features, thereby capturing target patterns more accurately. Overall, we selected SAF to replace the default LF for the following reasons: the accuracy improvement of SAF over the baseline model, as shown in Fig. 10, indicates that the model can extract and learn more complex patterns, supporting its suitability for few-shot environments. Moreover, this stability is not only evident in the numerical metrics but also reflects the model's enhanced adaptability to variations in data distribution, which offers a stronger safeguard against the diverse attacks encountered in practical applications. We further speculate that the design philosophy of SAF enables the model to maintain a high level of generalization even when facing few-shot or abnormal data, thereby improving the overall practicality and reliability of the system. Figures 8 and 9 show that the SAF results were stable and enhanced the detection across different types, providing a robust basis for transfer-enhanced learning.

Comparison of evaluation metrics for different models

In this section, we compare the performance of various deep-learning models in network intrusion detection tasks by assessing their performance with different sample sizes and providing the FLOPs and total parameters. The FLOPs are represented in millions (M) and Total params denote the total number of model parameters.

Table 10 shows that the proposed models outperformed other DNN models with K = 5, achieving accuracies of 93.1% and 98.8% with K = 15. While AlexNet and GoogleNet achieved accuracies of 93.3% and 93.7%, their FLOPs were significantly higher, at 90.59M and 116.61M, respectively. This indicates that our framework maintains high accuracy while reducing the computational resource requirements, effectively improving the model efficiency.

Feature extraction module	ACC (%)			FLOPs (M)	Params (M)
	K = 5	K = 10	K = 15		
CNN2D	83.6	89.3	91.4	38.97	0.13
CNN3D	84.4	92.3	89.7	43.69	11.24
ResNet18	88.4	90.0	92.6	142.43	11.17
VGG16	20.0	20.0	20.0	1367.65	134.27
AlexNet	82.2	92.5	93.3	90.59	57.01
GoogleNet	71.1	94.5	93.7	116.61	5.60
DenseNet	77.3	83.5	91.4	229.98	6.95
MobileNet	71.4	83.4	92.0	26.05	2.23
SqueezeNet	64.0	70.2	75.3	22.29	0.72
Self-Sufficient	92.8	92.9	93.1	43.99	11.45
Transfer-Enhanced	93.4	95.2	98.8	43.99	11.45

Table 10. Comparison of metrics for different feature extraction modules.

Summary of results

The proposed Self-Sufficient Model effectively addressed few-shot multiclass network intrusion detection, achieving 91.2% accuracy with only five samples in the baseline model. Selecting an appropriate multimodal fusion method increased the accuracy to 92.8%, which was an improvement of 1.6%. Utilizing prior data for pretraining with the Transfer-Enhanced Model further boosted the accuracy to 93.4%, which was a 2.2% increase. Although the Transfer-Enhanced Model improves the accuracy of the Self-Sufficient Model, it may lead to performance degradation when the sample size is small (K = 5) due to the inability to effectively transfer features from the source domain to the target domain. The introduction of the SAF (Self-Attention Fusion) method effectively addresses this issue by improving the performance of the Transfer-Enhanced Model with multimodal feature fusion, increasing the accuracy by 1.6% at K = 5. This combined approach of pretraining and multimodal fusion not only shows advantages with larger sample sizes but also effectively enhances the model's performance in few-shot settings, further demonstrating the potential of the Transfer-Enhanced Model. The following three conclusions can be drawn:

(1) The proposed multimodal feature extraction technique significantly improved the accuracy in few-shot environments. Even with a linear feature combination, it achieved 91.2% accuracy, outperforming other methods and demonstrating effective feature extraction.
(2) The multimodal fusion method reduced model fluctuations while improving accuracy and requiring fewer computational resources. SAF achieved 92.8% accuracy, which was superior to the baseline accuracy of 91.2%, with only 43.99M FLOPs, which was much lower than those of AlexNet (90.59M) and GoogleNet (116.61M).
(3) Leveraging prior data with the Transfer-Enhanced Model further enhanced the accuracy from 92.8% to 93.4% and from 92.9% to 95.2%, increasing it by 0.6% and 2.3%, respectively.

Discussion

In this section, we will explore the comparative results between the proposed model and other few-shot methods, as well as analyze the outcomes of the one-shot cases and ablation experiments. First, by comparing with the latest few-shot approaches, we will demonstrate the effectiveness and advantages of the multimodal method in network intrusion detection. Next, in the one-shot cases, we will test the model's adaptability under extremely limited sample conditions, examining the impact of transfer learning and feature fusion in such extreme scenarios. Finally, in the ablation study, we will assess the influence of different modality interaction dependencies to gain a deeper understanding of how feature extraction modules work together to enhance overall system performance, revealing the key role each feature extraction module plays in boosting system efficiency.

Comparison with related work

To investigate the performance of the proposed model, we compared it with recent few-shot methods. Unlike many existing methods that use only one type of file, the datasets used include both CSV and pcap files. It is important to note that the CSV files used in this study were created by us through feature extraction from the original pcap files. These feature files were saved in CSV format for easier storage and accessibility, and are not the same as the CSV files provided by CIC. This multimodal approach is a key innovation in the field. In Table 11, we summarize each method's sample size, type, dataset, accuracy, and three additional design dimensions: Feat. Ext., Data Type, and Learning Paradigm. Feat. Ext. indicates the backbone network used for feature extraction; Data Type indicates the form of input data; Learning Paradigm indicates the overall training strategy.

Table 11 shows selected similar works, where FE-MTDM used 1% of the CICIDS2017 dataset and GDE used a total of 140 samples. The sample sizes of the other models are shown per type, and not the total.The FSIDS-IOT[3] is a hybrid dataset that includes a mixture of five types of data from CICIDS2017 and CICIDS2018 datasets. The methods in the table mainly focus on different sample sizes (K values) and datasets. Our Self-Sufficient Model and Transfer-Enhanced Model show excellent performance. For example, on the CICIDS2017 dataset, the Self-Sufficient Model (K = 5) achieved an accuracy of 92.80%, while the Transfer-Enhanced Model (K = 5) reached

Model/method	K	Type	Dataset	ACC	Feat. Ext.	Data type	Learning paradigm
Meta-learning (2022)[38]	10	Multi	CICIDS2017	97.56%	CNN	Raw network traffic	Prototypical network
FS-IDS (2022)[39]	5	Binary	CICIDS2017	97.51%	CNN	Raw network traffic	Metric learning
SPN (2023)[40]	5	Binary	CICIDS2017	94.37%	CNN + Attention	Raw network traffic	Supervised learning
GDE (2023)[41]	140	Multi	CICIDS2018	99.13%	CNN	GAF image encoding	Diffusion model
Siamese Network (2023)[42]	1	Multi	CICIDS2017	80.81%	ANN	Statistical feature vectors	Siamese network
MetaMRE (2023)[43]	10	Binary	CICIDS2017	93.30%	Dilated causal conv	Raw network traffic	MAML
MetaMRE (2023)[43]	10	Multi	CICIDS2017	91.80%	Dilated causal conv	Raw network traffic	MAML
FE-MTDM (2023)[25]	65341 (1%)	Multi	CICIDS2017	99.70%	Shallow NN + RF	Raw + statistical features	Prototypical network
MAML+CNN (2023)[3]	5	Multi	FSIDS IoT	89.64%	CNN	Statistical feature vectors	MAML
FML (2024)[44]	10	Multi	CICIDS2017	87.27%	ResNet	Statistical feature vectors	Federated meta learning
Res-Natural GAN (2024)[45]	15	Binary	CICIDS2018	95.75%	GAN based CNN	Raw network traffic	Prototypical network
Self-Sufficient Model	5	Multi	CICIDS2017	92.80%	CNN + Transformer	Raw + statistical features	Supervised learning
Transfer-Enhanced Model	5	Multi	CICIDS2017	93.40%	CNN + Transformer	Raw + statistical features	Transfer learning
Self-Sufficient Model	10	Multi	CICIDS2017	92.90%	CNN + Transformer	Raw + statistical features	Supervised learning
Transfer-Enhanced Model	10	Multi	CICIDS2017	95.20%	CNN + Transformer	Raw + statistical features	Transfer learning
Self-Sufficient Model	5	Multi	CICIDS2018	98.40%	CNN + Transformer	Raw + statistical features	Supervised learning
Transfer-Enhanced Model	5	Multi	CICIDS2018	98.50%	CNN + Transformer	Raw + statistical features	Transfer learning
Self-Sufficient Model	10	Multi	CICIDS2018	98.70%	CNN + Transformer	Raw + statistical features	Supervised learning
Transfer-Enhanced Model	10	Multi	CICIDS2018	99.50%	CNN + Transformer	Raw + statistical features	Transfer learning

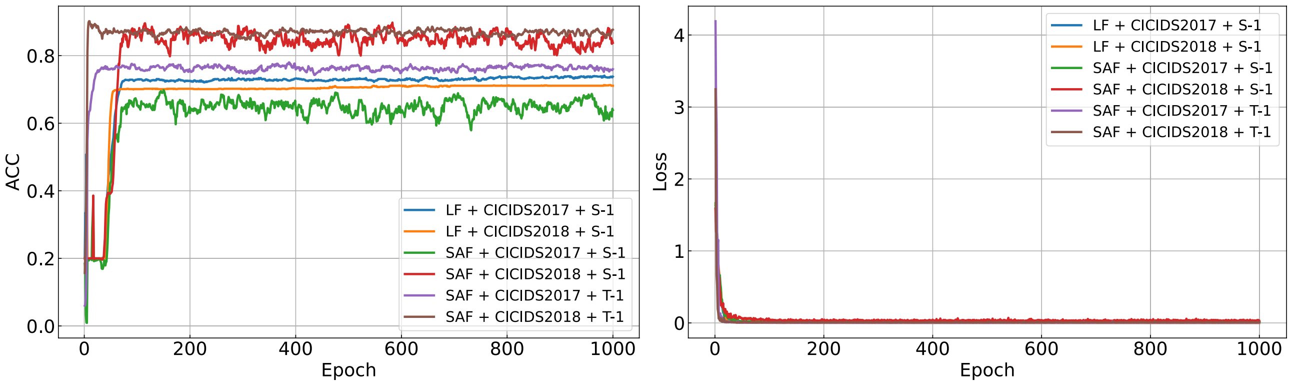
Table 11. Comparison of sample size, classification type, dataset, accuracy, feature extractor, data type, and learning paradigm in similar works.

93.40%. On the CICIDS2018 dataset, the Transfer-Enhanced Model (K = 5) achieved an accuracy of 98.50%, slightly higher than the Self-Sufficient Model (98.40%). In comparison with other methods, although our models are slightly lower than FE-MTDM (99.70%), the Transfer-Enhanced Model at K = 10 achieved 95.20% accuracy on the CICIDS2017 dataset, outperforming most similar models. Although FE-MTDM achieved a very high accuracy of 99.70% using only 1% of the CICIDS2017 dataset, its sample size is much larger than that in our experimental setup. Additionally, the performance of MAML+CNN on the FSIDS-IoT dataset indicates that while a smaller sample size can enhance generalization ability, there are still challenges in adaptability. To further explore the generalization ability of our model, we conducted a generalization analysis.

In terms of data types, existing methods can be grouped into four categories: the first uses only raw network traffic, including Meta-learning[38], FS-IDS[39], SPN[40], MetaMRE[43] and Res Natural GAN[45], which preserve packet-level timing and payload details to capture fine-grained attack patterns but lack flow-level global statistical information; the second relies solely on static statistical features, such as Siamese Network[42], MAML+CNN[3] and FML[44], using low-dimensional vectors such as packet counts, flow durations and byte distributions that are easy to interpret and computationally efficient but unable to model packet-level dynamics and therefore perform poorly in scenarios with brief spikes or similar statistical profiles; the third, exemplified by GDE[41], maps traffic into Gramian Angular Field images with diffusion-based augmentation and achieves high accuracy on CICIDS2018 but its preprocessing pipeline is complex and time-consuming and thus unsuitable for online deployment; and the fourth fuses raw traffic with static features, as in FE-MTDM[25] and our method, where FE-MTDM concatenates modalities at a high level and our approach employs multi-head self-attention for cross-modal interaction, enabling deeper dynamic-static integration and superior generalization in few-shot and cross-dataset transfer scenarios. In terms of backbone feature extraction, FE-MTDM uses a shallow neural network and random forest, which is lightweight and fast but limited in expressiveness; MAML+CNN[3] and Siamese Network[42] adopt convolutional architectures that excel at capturing local temporal patterns but struggle with long-range dependencies; SPN[40] enhances convolutional networks by adding an attention mechanism to emphasize critical packets; and our approach combines a CNN for temporal feature extraction with a Transformer encoding of statistical vectors, thereby retaining packet-level dynamics and capturing flow-level distributions to deliver stronger representational power at the cost of increased complexity and computation. Regarding convergence speed and rapid adaptation, prototype networks such as those in Meta-learning[38] require only prototype and classification-head learning to converge quickly and support incremental new classes; MAML-based methods like MetaMRE[43] leverage meta-training and fine-tuning to adapt swiftly in extremely low-data regimes albeit with higher training overhead; and transfer learning, when source and target domains are similar and class distributions differ markedly, achieves fast convergence and strong adaptation through pretraining plus lightweight fine-tuning but under extreme few-shot conditions or significant distribution shifts it cannot match MAML's rapid few-gradient-step adaptation.

One shot cases

We explored the performance of the models using only one sample and analyzed the impact of transfer learning and feature fusion in these extreme few-shot scenarios. The results are shown in Fig. 11 and Table 12, where S represents the Self-Sufficient Model and T represents the Transfer-Enhanced Model. The Self-Sufficient Model



(a) Accuracy variation curves for different models.

(b) Loss variation curves for different models.

Figure 11. Performance under different model, sample, and dataset combinations.

Model and sample	CICIDS2017	CICIDS2018
SAF + S-1	69.8%	89.8%
SAF + T-1	78.0%	90.2%
LF + S-1 (baseline)	74.1%	71.3%

Table 12. Effects of different models, sample sizes, and datasets on the test accuracy.

achieved accuracies of 74.1% and 71.3%, respectively. Incorporating different fusion models yielded varied results: a 4.3% decrease on CICIDS2017 but an 18.5% increase on CICIDS2018, indicating the sensitivity of the model to datasets, particularly in extreme few-shot scenarios. The Transfer-Enhanced Model showed increases of 8.2% and 0.4%, respectively.

In the CICIDS2017 dataset, we replaced LF with SAF, which performed best in the Self-Sufficient Model, resulting in a 4.3% decrease in model performance. In the Transfer-Enhanced Model, using SAF instead of LF improved model performance by 3.9%. This indicates that SAF has a certain dependency on sample size and that the Attention mechanism requires more data to leverage its advantages. In contrast, the Transfer-Enhanced Model utilized more data during training, thereby achieving performance gains. Under extremely small sample conditions, the Self-Sufficient Model combined with SAF failed to fully realize its performance and was even inferior to the LF-based model. For the CICIDS2018 dataset, SAF demonstrated significant performance improvements of 18.5% and 18.9% in both the Self-Sufficient Model and the Transfer-Enhanced Model, respectively. This suggests that under extremely small sample conditions, internal differences within the dataset are amplified, and the performance of the Attention mechanism varies significantly across different datasets. Additionally, compared to the Siamese Network in Table 11, the method proposed in this paper still has a 2.81% performance gap. This is also the direction for our future optimization efforts.

Ablation study

Finally, we designed specific feature extraction modules (G-Model and S-Model) to handle multimodal data. To investigate the impact of these modules on the model performance, we sequentially removed different feature extraction modules. This process helped us to understand the contribution and role of each module and how their interactions affected the overall experimental results. Specifically, we sequentially set the output dimensions of the G-Model and S-Model to zero to nullify the effects of each model. The experimental results are shown in Fig. 12, where S-Model + G-Model indicates the use of both models.

To evaluate the model thoroughly, we used additional evaluation metrics for the test set. The results are shown in Fig. 13.

Figures 12 and 13 show that removing the G-Model or S-Model resulted in significant declines in the ACC, Macro-F1, and Macro-PR metrics. Specifically, using only the S-Model resulted in decreases of 2.2%, 2.5%, and 2.8%, respectively, whereas using only the G-Model resulted in decreases of 2.6%, 3.3%, and 4.0%, respectively. These experimental results demonstrate that the standalone use of either the G-Model or S-Model significantly diminishes key performance metrics, emphasizing the critical role of both modules in enhancing overall model performance. Our analysis suggests that the G-Model primarily captures global feature information, providing a macroscopic foundation for discrimination, while the S-Model extracts fine-grained local features, addressing any details that may be overlooked by the global representation. This complementary dynamic enables the model to maintain robust discrimination overall, while still ensuring precise recognition of fine details, even when confronted with complex and diverse attack types. This design not only validates the effectiveness of multimodal feature extraction but also offers a theoretical framework for tackling diverse and emergent attacks in real-

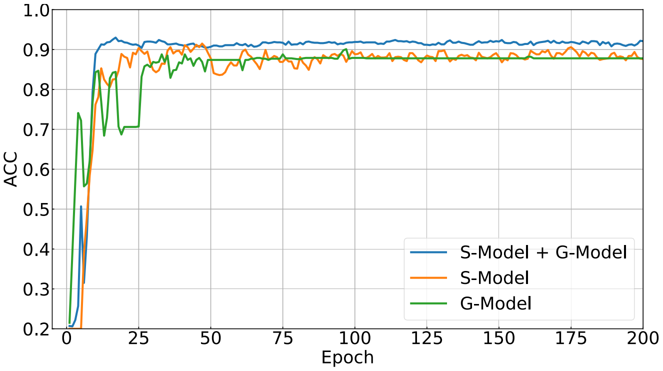


Figure 12. Accuracy across training epochs for different feature extractors.

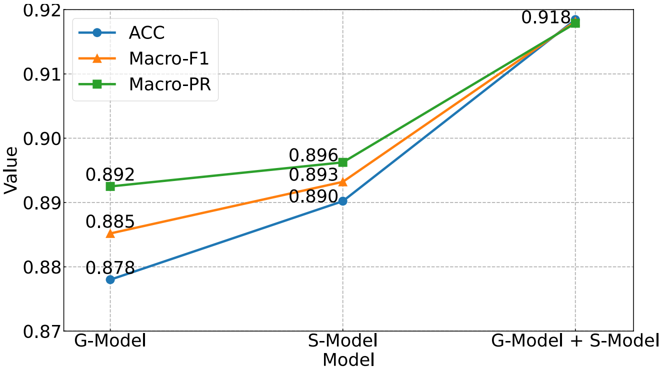


Figure 13. Performance comparison of different models on various evaluation metrics.

world settings. Furthermore, the synergy between the two modules boosts the model's adaptability to novel or unknown attack types, enhancing its competitiveness in few-shot scenarios.

In addition to the primary contributions of our work, we observe that the G-Model and S-Model, through their complementary relationship, further enhance the model's robustness and generalization ability in few-shot tasks. While this observation is based on experimental results and intuitive reasoning, it provides an additional insight into the potential advantages of the multimodal approach. These results demonstrate that both the G-Model and S-Model are crucial components of the proposed method. Their combination effectively handled multimodal data, showing clear advantages over single-feature extraction modules across various evaluation metrics. The ablation study confirmed the importance of each component and its significant contribution to improving the overall system performance.

Generalization analysis

To further investigate the model's generalization ability, specifically its adaptability to different attack types, we note that the CICIDS2017 and CICIDS2018 datasets contain the same attack types. This could cause the model to implicitly learn corresponding feature representations during training, which may affect its generalization performance. To address this issue, we merged these two datasets. Specifically, for attack types common to both datasets, we randomly selected half of the data from each dataset, resulting in a final dataset containing seven distinct attack types. To evaluate the model's generalization to unseen attack types, we employed a source-target domain splitting strategy. In the source domain, we pre-trained using n_{way} attack types, while in the target domain, we tested using the remaining $7 - n_{\text{way}}$ attack types. In the experiments, the number of samples in the source domain was fixed at 100, while the number of samples in the target domain was varied with K = 5, K = 10, and K = 15. Additionally, we tested with $n_{\text{way}} = 2, 3, 4$ to reflect the model's generalization performance across different classification scenarios and sample sizes. The results are shown in Table 13.

From the results in Table 13, it can be observed that in the 2-class case, the accuracy continuously increases as K increases, reaching 95.9% at K = 15. In the case of $n_{\text{way}} = 3$, since the source domain pre-training contains only 4 attack types, the target domain classification becomes a 3-class problem. This results in reduced pre-training data and increased classification difficulty, leading to a slight decrease in accuracy at different K values. As n_{way} further increases, this trend becomes more pronounced. Nevertheless, the model still achieves high accuracy in this scenario, with values of 88.9% and 92.6%, demonstrating that the proposed model retains strong generalization capabilities when faced with different classification tasks and attack types.

n_{way}	ACC (%)		
	K = 5	K = 10	K = 15
2	91.3	93.7	95.9
3	85.2	89.3	92.6
4	81.8	84.3	88.9

Table 13. Generalization performance of the model under different attack types.

Potential threats and assumptions

To more comprehensively evaluate the security and applicability of our proposed method in real-world scenarios, we present a set of assumptions regarding potential threat scenarios and possible defense strategies. We hypothesize that adversaries may alter network traffic features or execute data poisoning. In such cases, defense strategies could include anomaly detection[46] to identify feature alterations and robust data validation methods, such as outlier detection, to mitigate the impact of data poisoning. We also consider different levels of adversarial information access. In black-box attacks, adversaries rely on system outputs, while in white-box attacks[47], they can access internal model details. Potential countermeasures could involve model obfuscation techniques, like ensembling or gradient masking[48], and robust training to improve generalization. Finally, we assume that datasets in network intrusion detection research typically have high integrity, but in real-world deployments, data may be subject to corruption or interference[49]. In response, we suggest that real-time anomaly detection and continuous learning[50] could help the model adapt to evolving attack patterns and maintain performance in the presence of noise.

Regarding the detection of unseen attacks, we mainly focus on the few-shot scenario, in which the system encounters a novel attack type with only a very limited number of samples, whereas the completely zero-shot case belongs to a different research domain. In fact, in real-world environments, zero-day and real-time attack scenarios are often accompanied by an extremely limited number of attack samples, sometimes even only a single sample; in such one-shot cases, the detection challenge is particularly severe. Traditional methods that rely on large amounts of training data often struggle to adapt quickly to such extreme cases, whereas our algorithm is specifically designed for few-shot (and even one-shot) scenarios. Existing methods leverage pre-trained models with domain adaptation strategies[51] to rapidly capture novel attack behaviors, enabling timely responses and real-time detection of zero-day attacks. While our approach does not currently incorporate this, it could benefit from this strategy in future work, thus extending its applicability to a broader range of practical scenarios.

As for the detection of encrypted traffic, although direct detection without decryption poses significant challenges, previous research has demonstrated that effective identification of malicious traffic can be achieved by extracting statistical features that remain invariant during the encryption process[52,53]. In summary, despite the valuable insights provided by our theoretical analysis, we currently lack experimental validation for these potential threats and defense strategies; in future work, we plan to design and implement systematic experiments to evaluate and enhance the robustness of our method against unseen attacks, interference from adversarial samples, encrypted traffic, and other complex scenarios.

Future work

In our future research, we plan to take the following measures to address the current challenges faced by our system: First, to tackle computational complexity, we will develop more streamlined model architectures and experiment with model compression techniques such as parameter pruning and quantization to reduce resource consumption. Simultaneously, we plan to leverage the latest hardware acceleration[54] technologies and optimized distributed computing strategies to enhance our capability to process large-scale data. Regarding fusion strategies, we will explore methods that reduce computational burden without sacrificing prediction accuracy and investigate new lightweight fusion frameworks suitable for multimodal data. In terms of explainability, we will delve deeper into integrating advanced explainable artificial intelligence technologies[55], such as feature attribution, visualization of attention mechanisms[56], and surrogate models. This will enhance the transparency and trustworthiness of our models[57], increasing security analysts' reliance on our judgments. Through these measures, we aim not only to solve efficiency and performance issues but also to significantly enhance user acceptance and the practical value of our models. Furthermore, we will continue our research and technological innovations to further optimize our multimodal detection systems, making them more adaptable to the rapidly changing landscape of network security.

Conclusion

This study introduces a multimodal fusion NIDS to address the limitations of single-modality methods in detecting diverse and complex multiclass attacks. The system employs heterogeneous data generation techniques to produce multimodal data from different perspectives and integrates CNNs with the Transformer's multi-head attention for effective local and global information processing. Additionally, we implement three feature fusion strategies to integrate information from various modalities, enhancing detection performance and generalization. Experimental results on CICIDS2017 and CICIDS2018 datasets show that even with only five samples, our system achieves accuracies of 93.40% and 98.50%, surpassing existing technologies. In future work, further refinement of fusion strategies and system expansion will be required to counter evolving network threats more effectively.

Data availability

The two datasets and code used in this study can be accessed through the following links: the CICIDS2017 dataset https://www.unb.ca/ cic/datasets/ids-2017.html and the CICIDS2018 dataset https://www.unb.ca/cic/datasets/ids-2018.html. The code for this study can be found at https://github.com/cyxuzju/multimodal-IDS.

Received: 7 November 2024; Accepted: 2 June 2025
Published online: 01 July 2025

References

1. Kaur, R., Gabrijelčič, D. & Klobučar, T. Artificial intelligence for cybersecurity: Literature review and future research directions. *Inf. Fusion* **97**, 101804 (2023).
2. Thakkar, A. & Lohiya, R. Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system. *Inf. Fusion* **90**, 353–363 (2023).
3. Lu, C., Wang, X., Yang, A., Liu, Y. & Dong, Z. A few-shot-based model-agnostic meta-learning for intrusion detection in security of internet of things. *IEEE Internet Things J.* **10**, 21309–21321 (2023).
4. Zhang, X. et al. Enhanced few-shot malware traffic classification via integrating knowledge transfer with neural architecture search. *IEEE Trans. Inf. Forensics Secur.* **19**, 5245–5256 (2024).
5. Hore, S., Ghadermazi, J., Shah, A. & Bastian, N. D. A sequential deep learning framework for a robust and resilient network intrusion detection system. *Comput. Secur.* **144**, 103928 (2024).
6. Wei, N. et al. An autoencoder-based hybrid detection model for intrusion detection with small-sample problem. *IEEE Trans. Netw. Serv. Manage.* **21**, 2402–2412 (2024).
7. Roy, S., Li, J., Choi, B.-J. & Bai, Y. A lightweight supervised intrusion detection mechanism for iot networks. *Futur. Gener. Comput. Syst.* **127**, 276–285 (2022).
8. Thakkar, A., Kikani, N. & Geddam, R. Fusion of linear and non-linear dimensionality reduction techniques for feature reduction in lstm-based intrusion detection system. *Appl. Soft Comput.* **154**, 111378 (2024).
9. Alzubi, J. A., Alzubi, O. A., Qiqieh, I. & Singh, A. A blended deep learning intrusion detection framework for consumable edge-centric iomt industry. *IEEE Trans. Consum. Electron.* **70**, 2049–2057 (2024).
10. Louk, M. & Tama, B. A. Dual-ids: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system. *Expert Syst. Appl.* **213**, 119030 (2023).
11. Sharifian, Z., Barekatain, B., Quintana, A. A., Beheshti, Z. & Safi-Esfahani, F. Sin-cos-biavoa: A new feature selection method based on improved african vulture optimization algorithm and a novel transfer function to ddos attack detection. *Expert Syst. Appl.* **228**, 120404 (2023).
12. Zhiqiang, L., Mohiuddin, G., Jiangbin, Z., Asim, M. & Sifei, W. Intrusion detection in wireless sensor network using enhanced empirical based component analysis. *Futur. Gener. Comput. Syst.* **135**, 181–193 (2022).
13. Hassini, K., Khalis, S., Habibi, O., Chemmakha, M. & Lazaar, M. An end-to-end learning approach for enhancing intrusion detection in industrial-internet of things. *Knowl.-Based Syst.* **294**, 111785 (2024).
14. Verkerken, M. et al. A novel multi-stage approach for hierarchical intrusion detection. *IEEE Trans. Netw. Serv. Manage.* **20**, 3915–3929 (2023).
15. Moizuddin, M. & Jose, M. V. A bio-inspired hybrid deep learning model for network intrusion detection. *Knowl.-Based Syst.* **238**, 107894 (2022).
16. Ciric, V., Milosevic, M., Sokolovic, D. & Milentijevic, I. Modular deep learning-based network intrusion detection architecture for real-world cyber-attack simulation. *Simul. Model. Pract. Theory* **133**, 102916 (2024).
17. Chowdhury, R., Sen, S., Goswami, A., Purkait, S. & Saha, B. An implementation of bi-phase network intrusion detection system by using real-time traffic analysis. *Expert Syst. Appl.* **224**, 119831 (2023).
18. Nguyen, H. & Kashef, R. Ts-ids: Traffic-aware self-supervised learning for iot network intrusion detection. *Knowl.-Based Syst.* **279**, 110966 (2023).
19. Basati, A. & Faghih, M. M. Pdae: Efficient network intrusion detection in iot using parallel deep auto-encoders. *Inf. Sci.* **598**, 57–74 (2022).
20. Milosevic, M. S. & Ciric, V. M. Extreme minority class detection in imbalanced data for network intrusion. *Comput. Secur.* **123**, 102940 (2022).
21. Chapaneri, R. & Shah, S. Enhanced detection of imbalanced malicious network traffic with regularized generative adversarial networks. *J. Netw. Comput. Appl.* **202**, 103368 (2022).
22. Attique, D., Hao, W., Ping, W., Javeed, D. & Kumar, P. Explainable and data-efficient deep learning for enhanced attack detection in iiot ecosystem. *IEEE Internet Things J.* (2024).
23. Hu, X. et al. Towards early and accurate network intrusion detection using graph embedding. *IEEE Trans. Inf. Forensics Secur.* **18**, 5817–5831 (2023).
24. Behiry, M. H. & Aly, M. Cyberattack detection in wireless sensor networks using a hybrid feature reduction technique with ai and machine learning methods. *J. Big Data* **11**, 16 (2024).
25. Wei, N. et al. A feature enhancement-based model for the malicious traffic detection with small-scale imbalanced dataset. *Inf. Sci.* **647**, 119512 (2023).
26. Thakkar, A. & Lohiya, R. Attack classification of imbalanced intrusion data for iot network using ensemble learning-based deep neural network. *IEEE Internet Things J.* **10**, 11888–11895 (2023).
27. Batchu, R. K. & Seetha, H. An integrated approach explaining the detection of distributed denial of service attacks. *Comput. Netw.* **216**, 109269 (2022).
28. Hwang, R.-H. et al. Host-based intrusion detection with multi-datasource and deep learning. *J. Inf. Secur. Appl.* **78**, 103625 (2023).
29. Thakkar, A. & Lohiya, R. Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system. *Inf. Fusion* **90**, 353–363 (2023).
30. Liu, Q., Wang, D., Jia, Y., Luo, S. & Wang, C. A multi-task based deep learning approach for intrusion detection. *Knowl.-Based Syst.* **238**, 107852 (2022).
31. Fu, J.-J. & Zhang, X.-L. Gradient importance enhancement based feature fusion intrusion detection technique. *Comput. Netw.* **214**, 109180 (2022).
32. Jiang, H., Lin, J. & Kang, H. Fgmd: A robust detector against adversarial attacks in the iot network. *Futur. Gener. Comput. Syst.* **132**, 194–210 (2022).
33. Zang, X., Gong, J., Zhang, X. & Li, G. Attack scenario reconstruction via fusing heterogeneous threat intelligence. *Comput. Secur.* **133**, 103420 (2023).
34. Xu, C., Shen, J. & Du, X. A method of few-shot network intrusion detection based on meta-learning framework. *IEEE Trans. Inf. Forensics Secur.* **15**, 3540–3552 (2020).
35. Wang, W. et al. Hast-ids: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **6**, 1792–1806 (2018).

36. Leevy, J. L. & Khoshgoftaar, T. M. A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data. *J. Big Data* **7**, 1–19 (2020).
37. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008 (2017).
38. Xu, H. & Wang, Y. A continual few-shot learning method via meta-learning for intrusion detection. In *2022 IEEE 4th International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 1188–1194 (2022).
39. Yang, J., Li, H., Shao, S., Zou, F. & Wu, Y. Fs-ids: A framework for intrusion detection based on few-shot learning. *Comput. Secur.* **122**, 102899 (2022).
40. Liu, Y. et al. Semi-supervised few-shot network intrusion detection based on meta-learning. In *2023 IEEE International Conferences on Internet of Things (iThings) and IEEE Green Computing & Communications (GreenCom) and IEEE Cyber, Physical & Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics)*, 495–502 (2023).
41. Yan, Y., Yang, Y., Shen, F., Gao, M. & Gu, Y. Gde model: A variable intrusion detection model for few-shot attack. *J. King Saud Univ. Comput. Inf. Sci.* **35**, 101796 (2023).
42. Hindy, H. et al. Leveraging siamese networks for one-shot intrusion detection model. *J. Intell. Inf. Syst.* **60**, 407–436 (2023).
43. Yang, C. et al. Few-shot encrypted traffic classification via multi-task representation enhanced meta-learning. *Comput. Netw.* **228**, 109731 (2023).
44. Hu, Y., Wu, J., Li, G., Li, J. & Cheng, J. Privacy-preserving few-shot traffic detection against advanced persistent threats via federated meta learning. *IEEE Trans. Netw Sci. Eng.* **11**, 2549–2560 (2024).
45. Yan, Y., Yang, Y., Shen, F., Gao, M. & Gu, Y. Meta learning-based few-shot intrusion detection for 5g-enabled industrial internet. *Complex Intell. Syst.* **10**, 4589–4608 (2024).
46. Wang, X. et al. Federated deep learning for anomaly detection in the internet of things. *Comput. Electr. Eng.* **108**, 108651 (2023).
47. Zhu, Y. et al. Black box attack and network intrusion detection using machine learning for malicious traffic. *Comput. Secur.* **123**, 102922 (2022).
48. Alsaffar, A. M., Nouri-Baygi, M. & Zolbanin, H. M. Shielding networks: enhancing intrusion detection with hybrid feature selection and stack ensemble learning. *J. Big Data* **11**, 133 (2024).
49. Roshan, M. K. & Zafar, A. Boosting robustness of network intrusion detection systems: A novel two phase defense strategy against untargeted white-box optimization adversarial attack. *Expert Syst. Appl.* **249**, 123567 (2024).
50. Gyamfi, E. & Jurcut, A. D. Novel online network intrusion detection system for industrial IoT based on oi-svdd and as-elm. *IEEE Internet Things J.* **10**, 3827–3839 (2023).
51. Hashemi, M. J., Keller, E. & Tizpaz-Niari, S. Detecting unseen anomalies in network systems by leveraging neural networks. *IEEE Trans. Netw. Serv. Manage.* **20**, 2515–2528 (2022).
52. Xu, B., He, G. & Zhu, H. Me-box: A reliable method to detect malicious encrypted traffic. *J. Inf. Secur. Appl.* **59**, 102823 (2021).
53. Zebin, T., Rezvy, S. & Luo, Y. An explainable AI-based intrusion detection system for DNS over HTTPS (DoH) attacks. *IEEE Trans. Inf. Forensics Secur.* **17**, 2339–2349 (2022).
54. Deng, L., Li, G., Han, S., Shi, L. & Xie, Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* **108**, 485–532 (2020).
55. Minh, D., Wang, H. X., Li, Y. F. & Nguyen, T. N. Explainable artificial intelligence: a comprehensive review. *Artif. Intell. Rev.* **55**, 3503–3568 (2022).
56. Chefer, H., Gur, S. & Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 782–791 (2021).
57. Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y. & Tari, Z. Explainable intrusion detection for cyber defences in the internet of things: Opportunities and solutions. *IEEE Commun. Surveys Tutorials* **25**, 1775–1807 (2023).

Acknowledgements

This research was supported by the National Natural Science Foundation of China (Grant No. 62302197), Zhejiang Provincial Natural Science Foundation of China (Grant No. LQ23F020006), Jiaxing City Science and Technology Project (Grant No. 2024AY40010, Grant No. 2024AD10045) and China Postdoctoral Science Foundation under Grant Number 2024M752366.

Author contributions

C.X. conceived the idea, Y.Z. designed the experiments, Z.W. analysed the results. C.X., Y.Z. and Z.W. wrote the article. C.X. and J.Y. supervised the manuscript. All authors reviewed the manuscript.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to C.X.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.