electronics

MDPI

Article

Boosting Few-Shot Network Intrusion Detection with Adaptive Feature Fusion Mechanism

Jue Bo [1], Kai Chen [2,*], Shenghui Li [2] and Pengyi Gao [2]

[1] College of Cyber Security, National Cybersecurity Talent and Innovation Base, Huazhong University of Science and Technology, Wuhan 430040, China; bo5@hust.edu.cn
[2] College of Cyber Security, Main Campus, Huazhong University of Science and Technology, Wuhan 430074, China; lishenghui@hust.edu.cn (S.L.); pengyi_gao@hust.edu.cn (P.G.)
* Correspondence: kchen@hust.edu.cn

Abstract: In network security, intrusion detection systems (IDSs) are essential for maintaining network integrity. Traditional IDSs primarily use supervised learning, relying on extensive datasets for effective training, which limits their ability to address rapidly evolving cyber threats, especially with limited data samples. To overcome this, prior research has applied meta-learning methods to distinguish between normal and malicious network traffic, showing promising results mainly in binary classification scenarios. However, challenges remain in model information acquisition within few-shot learning (FSL) frameworks. This study introduces a metric-based meta-learning strategy that constructs prototypes for each sample category, improving the model's ability to manage multi-class scenarios. Additionally, we propose an Adaptive Feature Fusion (AFF) mechanism that dynamically integrates statistical features and binary data streams to extract meaningful insights from limited datasets, thereby enhancing the effectiveness of IDSs in few-shot learning contexts. By introducing a metric-based meta-learning method and the Adaptive Feature Fusion mechanism, this study provides a feasible solution for developing a high-accuracy, multi-class few-shot intrusion detection system. A series of experiments show that this approach significantly improves the effectiveness of the intrusion detection system, achieving an impressive accuracy of 97.78% in multi-class tasks, even when the sample size is reduced to just one.

Keywords: network intrusion detection; few-shot learning; adaptive feature fusion; meta-learning

check for updates

Citation: Bo, J.; Chen, K.; Li, S.; Gao, P. Boosting Few-Shot Network Intrusion Detection with Adaptive Feature Fusion Mechanism. *Electronics* **2024**, *13*, 4560. https://doi.org/10.3390/electronics13224560

Academic Editors: Zbigniew Kotulski and Aryya Gangopadhyay

Received: 18 September 2024
Revised: 4 November 2024
Accepted: 13 November 2024
Published: 20 November 2024

1. Introduction

With the proliferation of diverse and increasingly sophisticated cyber threats, network intrusion detection has emerged as a crucial tool for identifying malicious network traffic and maintaining the security of digital ecosystems [1]. As the hardware capabilities of devices continue to advance, many network intrusion detection systems have integrated machine learning and deep learning models to enhance their detection capabilities [2]. These systems, when combined with the deep learning method, rely more heavily on ample training samples to achieve improved generalization compared to traditional rule-based intrusion detection systems. However, the ever-evolving landscape of novel network attack methodologies poses significant challenges, especially when these emerging network attacks are characterized by a scarcity of attack samples. Traditional supervised intrusion detection systems, reliant on a large number of samples, face difficulties in promptly detecting and defending against these new types of network attacks.

To address the challenges outlined above, recent advancements in network intrusion detection have increasingly focused on the integration of meta-learning mechanisms. In this context, few-shot learning is particularly noteworthy, as it enables systems to learn from a minimal number of samples, thereby enhancing their adaptability to new and previously unseen threats. This capability is crucial for countering novel cyber threats, where traditional systems often struggle due to insufficient data.

Meta-learning frameworks are specifically designed to mitigate the issue of few-shot learning for emerging threats, such as zero-day vulnerabilities [3]. Prior research [4–6] has demonstrated that the incorporation of meta-learning frameworks into network intrusion detection systems significantly improves their ability to manage few-shot scenarios. By effectively extracting and transferring knowledge from specific tasks, these frameworks enable models to achieve remarkable recognition performance, even with limited sample sizes. Furthermore, our intrusion detection method is founded on a metric-based meta-learning approach, which enables it to effectively address multi-class scenarios. We hypothesize that by establishing a prototype for each category and classifying new samples through distance measurement, this approach will enable the system to proficiently manage multi-classification problems.

Adequate information is crucial for model construction and is a key element in ensuring the accuracy of intrusion detection systems. However, dependency solely on meta-learning algorithms cannot fully address the information loss caused by the few-shot problem. Therefore, in this system, we introduce an Adaptive Feature Fusion mechanism to augment the data acquired by the model from a representational perspective. Typically, network traffic in intrusion detection can be represented in two forms: one presents data packets in a binary data stream format, offering the advantage of input that closely resembles the original data and retains fine details; the other involves the extraction of network features from traffic, transforming data packets into a series of intuitive statistical features. AFF dynamically combines these two representations of network traffic, adaptively integrating the strengths of both. In cases of network traffic characterized by significant differences in macro-level statistical methods, AFF prioritizes statistical feature representation, while in other scenarios, it focuses on the binary data stream representation. In scenarios with a limited number of samples, AFF efficiently extracts valuable information from the samples, thereby mitigating the problem of insufficient information to a certain extent. We hypothesize that our proposed method will significantly enhance the classification accuracy in both binary and multi-class tasks when compared to traditional methods that do not incorporate the Adaptive Feature Fusion mechanism. This enhancement is attributed to the AFF mechanism's ability to extract more valuable information from a fixed number of samples, thereby improving the differentiation of various network traffic patterns.

The primary contributions of this paper are as follows:

1. We develop an intrusion detection method using a metric-based meta-learning approach, which not only distinguishes between normal and malicious network traffic but also excels in handling multi-class tasks in few-shot scenarios.
2. We introduce an Adaptive Feature Fusion mechanism, which effectively combines two types of traffic data representations, thereby enriching the information content of each sample and enhancing the efficacy of few-shot learning in network intrusion detection.
3. We conduct extensive experiments on the network intrusion detection method incorporating AFF. The results demonstrate its excellent performance in both binary and multi-class scenarios. Furthermore, the feasibility experiments validate its practical applicability.

2. Related Work

2.1. Intrusion Detection Based on Deep Learning

As computational resources for neural networks continue to expand, a prevalent trend in applying deep learning involves employing the proposed neural network architectures for feature extraction to classify traffic of different categories [7]. A highly scalable hybrid deep neural network (DNN) was introduced by [8], which underwent benchmark testing across multiple datasets. The experiments demonstrated that deep learning methods can capture more profound feature information, providing an advantage over previous research in intrusion detection tasks.

However, while the HAST-IDS introduced by [9] utilizes CNN for spatial feature extraction and RNN for temporal features—achieving a detection rate of 96.96% on the ISCX-2012 dataset—its reliance on specific architectures limits adaptability to varied traffic conditions. Similarly, the DST-IDS proposed by [10] combines packet-based and flow-based techniques, transforming raw packets into 28 × 28 × 1 images, resulting in an accuracy of 96.27% on the CICIDS2017 dataset. While effective, both systems demonstrate a lack of integration of statistical feature representation, potentially constraining their overall performance.

In contrast to prior intrusion detection systems, which frequently focus exclusively on either statistical feature representation or binary data stream representation, FS-IDS [11] aims to enhance classification accuracy within the binary data stream representation by integrating statistical feature information. However, its lack of adaptiveness to specific task requirements limits the maximization of information utilization, suggesting a gap in flexibility compared to models that leverage both feature types effectively.

2.2. Few-Shot Learning

Traditional deep learning methods often rely on a large amount of labeled data, which can be resource intensive and unattainable in some cases. In situations with limited labeled data samples, deep learning models can struggle to generalize effectively due to overfitting issues. In contrast, humans, equipped with prior knowledge and relevant backgrounds, can rapidly transfer their acquired knowledge to new task scenarios, thereby effectively completing tasks even in few-shot learning scenarios [12].

In recent years, meta-learning has gained popularity and been widely applied to the challenges of few-shot learning. Generally, meta-learning can be seen as a process of extracting knowledge from multiple similar learning tasks and improving subsequent performance through experience [13], often referred to as "learning to learn". Existing meta-learning methods for few-shot learning typically involve learning from a limited number of target tasks with a large number of similar tasks to create a classifier that can adapt to various scenarios [14]. The Meta-Learning Algorithmic Model (MAML) introduced by [15] achieves rapid convergence with minimal data by updating parameter values to change gradient-based optima, thereby adapting to new application scenarios. Matching networks [16] draw inspiration from some concepts in Metric Learning within deep learning and enhance network capabilities using external memory. Prototypical networks [17] are also metric-based meta-learning methods, directly determining sample categories by their proximity to prototypes, which are the mean representations of all samples within each category in a shallow feature space.

2.3. Application of Few-Shot Learning in the Field of Intrusion Detection

Confronted with emerging novel network attacks, traditional intrusion detection relying on a plethora of labeled samples struggles to effectively detect threats. Consequently, some studies seek to integrate few-shot learning into intrusion detection systems, leveraging meta-learning frameworks to achieve effective detection even in scenarios with extremely limited sample sizes. In the paper [18], the FC-Net is introduced as the first intrusion detection system to incorporate a meta-learning framework. FC-Net, with a CNN as its backbone, transforms data packets into 16×16 RGB images as input, achieving a notable accuracy of 94.64% on the CICIDS2017 dataset. Built upon the MAML-based meta-learning framework, ref. [19] establishes a few-shot intrusion detection task. Leveraging the learn to forget mechanism, the model achieves rapid convergence, ultimately attaining an accuracy of 94.66% in the binary classification task on the CICIDS2017 dataset.

While these studies validate the feasibility of meta-learning models in addressing few-shot challenges in intrusion detection, they primarily focus on binary classification tasks. There remains a critical need for solutions tailored specifically for multi-class classification tasks. Thus, this paper builds on the enhanced performance in binary scenarios to conduct comprehensive experiments targeting multi-class intrusion detection, thereby distinguishing its contributions in addressing the identified gaps.

3. Problem Formulation

In few-shot scenarios, the samples within the task set constitute the meta-training set, while the few samples corresponding to the target task constitute the meta-test set. Typically, in the realm of few-shot learning, a model is trained to differentiate among instances from N different categories by being given K instances for each category, constituting what is known as K-shot N-way tasks. As shown in Figure 1, during a training episode or task, a subset of categories is randomly selected from the training set, and a portion of instances from each category is randomly sampled to form the support set. Additional instances are selected from the remaining categories to construct the query set. With the support set provided, the model is trained iteratively to minimize the prediction loss on the query set.

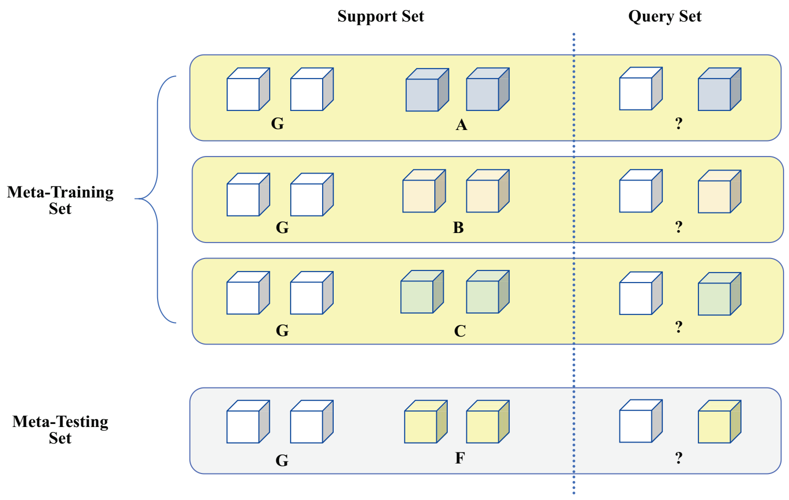

Figure 1. An abstract illustration demonstrating the meta-learning process. G represents normal traffic, A, B, C and F represent malicious traffic, with F having very few samples. The symbol ? denotes the classification of the sample into its respective category.

In the field of network intrusion detection, we define a task T_i as a binary classification task distinguishing between normal (G) and a specific type of malicious sample (A, B, C, or F). G represents normal traffic, while A, B, C, and F represent different malicious traffic types. Types A, B, and C have ample labeled samples in the dataset, whereas F is a new type with only K samples available. The primary goal is to perform task T_f by learning from K samples each of type F and normal type G, forming a classifier f. This classifier will then identify type F malicious samples among unknown samples. The support set for this task is $S_f = \{(x_1, y_1), \ldots, (x_{2K}, y_{2K})\}$, where $y_i \in \{G, F\}$. Given the small K, these $2K$ samples alone are insufficient for T_f.

To address this, we leverage tasks T_a, T_b, and T_c, which are structurally similar to T_f but use malicious samples of types A, B, and C, respectively. For example, task T_a uses K samples, each of types A and G, forming $S_a = \{(x_1, y_1), \ldots, (x_{2K}, y_{2K})\}$. These tasks simulate the support and test sets for T_f. In meta-learning terms, T_{train} (comprising tasks like T_a, T_b, and T_c) and T_{test} (comprising tasks like T_f) are the meta-training and meta-testing sets, respectively.

4. Methodology

4.1. Metric-Based Meta-Learning Model

The approach of metric-based meta-learning involves utilizing the distances between categories as indicators to facilitate the learning process. In this paper, we use a prototypical network as the foundational approach and illustrate the implementation of AFF. In a single training episode of K-shot N-way tasks, the support set, denoted as $S = \{(e_1, y_1), \ldots (e_N, y_N)\}$, consists of samples with feature vectors $e_i \in \mathbb{R}_D$, where D represents the dimensionality of the input feature vectors that encapsulate the characteristics of each sample. The corresponding labels $y_i \in \{1, \ldots, K\}$ indicate the categories of these samples. The prototypical network computes a representation $P_k \in \mathbb{R}_M$ for each category k using the support set, often referred to as the prototype, where M signifies the dimensionality of the prototype representations that summarize the information from the corresponding feature vectors, facilitating classification. Each category's prototype is essentially the mean feature embedding of the samples belonging to that category in the support set, with feature embedding generated by the embedding function $f_\phi : \mathbb{R}_D \to \mathbb{R}_M$, where ϕ represents learnable parameters:

$$P_k = \frac{1}{|S_k|} \sum_{(e_i, y_i) \subset S_k} f_\phi(e_i) \tag{1}$$

Following this, the prototype network classifies each sample in the query set based on the distances to various class prototypes. In this paper, we employ the distance d for calculating distances, and the class distribution for sample e is determined by the softmax corresponding to distances in the embedding space:

$$p_\phi(y = k|e) = \frac{exp(-d(f_\phi(e), P_k))}{\sum_{k'} exp(-d(f_\phi(e), P_{k'}))} \tag{2}$$

In practical implementation, the distance function d used is the Euclidean distance because of its excellent computational efficiency and effectiveness as demonstrated in previous work [20,21].

4.2. Network Traffic Representation

Binary data stream representation. From a transmission perspective, transmitted data packets are effectively converted from bitstreams into electrical signals for network transmission. These data packets are then parsed by protocols other than the physical layer in the Open Systems Interconnection (OSI) protocol suite. In this context, a network traffic flow can be defined as the sequence of packets exchanged between two endpoints in a network connection, typically beginning with a TCP connection establishment (SYN) and concluding with the connection termination (FIN). As a result, data packets within network traffic can be seen as a combination of payloads and data packet headers from various protocol layers, such as IP and TCP. Each high-level data packet header contains semantic information, including elements like version, length, and identifier. However, due to the utilization of different protocols and varying content, achieving uniform representation and constant sizes for data packets can be challenging.

To address this issue, nPrint [22] represents packets in their raw binary data form, ensuring that semantic information can be accurately extracted through predefined formats. By preserving a fixed number of bytes for each protocol header and aligning the data packets with padding, nPrint ensures consistency in format. Figure 2 illustrates the partitioning and meaning of the data.

Recent work [23,24] has introduced a trend of transforming each byte in data traffic into corresponding pixels in grayscale images and demonstrated the effectiveness of this approach. However, due to the nonuniform sizes and formats of data packets, the straightforward stacking of raw binary data does not effectively leverage the feature extraction capabilities of CNN architectures, which rely on gathering information from pixels and their adjacent pixels.

To address this issue, our research aligns the binary data stream and maps bits from data packets to pixels in grayscale images, thereby creating image-based representations. In order to balance information retention and computational complexity, we select the first five data packets from each network traffic flow and the first 256 bytes from each packet.

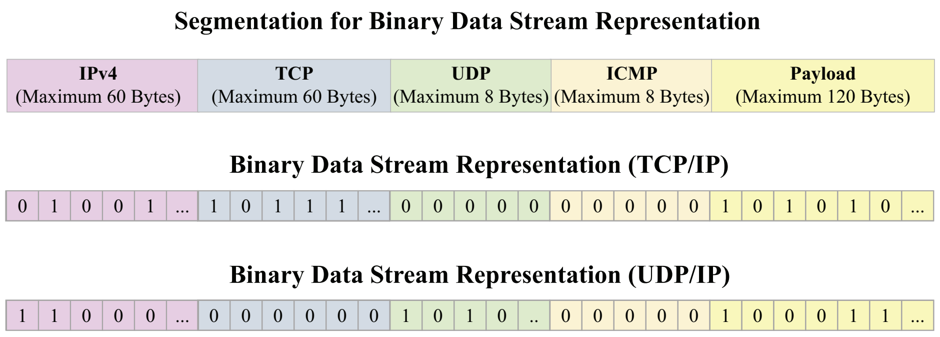

Figure 2. Segmentation and examples of binary data stream representation.

Subsequently, we arrange the 256 bits from each packet into a 16 × 16 grid to form grayscale images, stacking the packets in chronological order. This method maximizes the feature extraction capabilities of the convolutional neural network (CNN) while effectively representing the binary data stream. Each grayscale image in the sequence can be regarded as a snapshot of the data packets, illustrating various layers of information derived from the binary stream. The specific process for handling the binary data stream representation is illustrated in Figure 3.

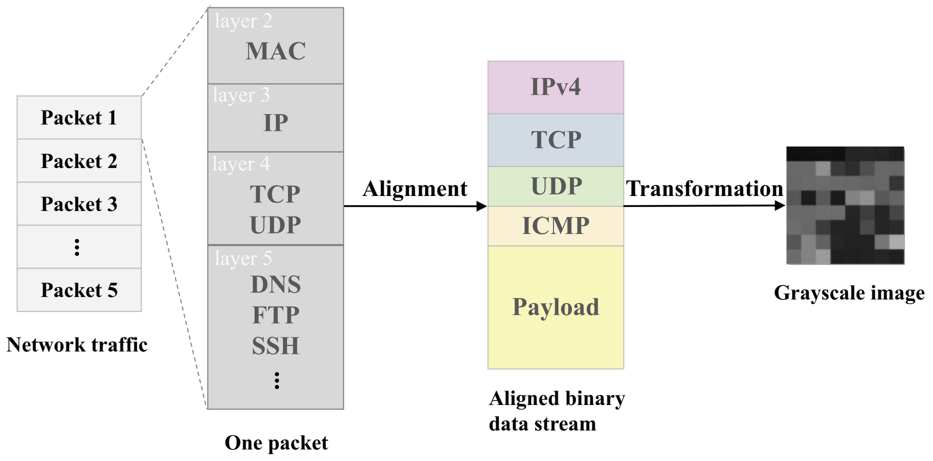

Figure 3. Pipeline of handling a binary data stream representation.

Statistical feature. In this study, we use the CICFlowMeter-4.0 tool to extract statistical features from network traffic, such as flow size, flow duration, and packet count. We omit sensitive information such as IP addresses and port numbers, so 76 out of the initially extracted features are retained as the final set of statistical features. Given the significant variations in the value ranges of different features, we perform data cleansing and subsequently apply min-max normalization to standardize the data and expedite model convergence. For each feature dimension, the normalization formula is as follows:

$$x_i^* = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{3}$$

where max (x_i) and min (x_i) represent the maximum and minimum values of the specific feature across all samples. This data transformation maps the values of each feature to the range $[-1, 1]$. The application of min-max normalization mitigates the influence of dimensional disparities within the data and yields the final set of statistical features.

4.3. Adaptive Feature Fusion Mechanism

Statistical features and binary data stream features provide significantly different types of information for the same network traffic. Statistical features mainly take a macroscopic approach, summarizing important characteristics of the given network traffic, providing a relatively comprehensive overview. On the other hand, binary flow data focus more on the detailed information contained within each packet, including the semantic details of various protocol headers at different layers and payload contents.

To effectively harness the information from both representations and further enhance performance in few-shot learning (FSL), we introduce an Adaptive Feature Fusion (AFF) mechanism. This mechanism is designed to fuse heterogeneous features while dynamically adjusting its emphasis based on specific target tasks and variations in training set samples. In this context, the AFF mechanism can dynamically adjust the focus on these two feature types based on the specific network traffic scenario. For instance, during a DDoS attack, where traffic patterns may shift rapidly and unpredictably, the AFF mechanism can prioritize binary data stream features to capture the nuanced behavior of malicious packets while still retaining the broader statistical context. By fine-tuning the fusion coefficients, the system can enhance its sensitivity to real-time threats, ensuring that critical features are highlighted to improve detection accuracy. Specifically, if a spike in packet size or frequency is detected, the AFF mechanism may increase the weight of binary features to focus on the packet contents that reveal the attack's nature, thereby improving the system's responsiveness to evolving attack strategies. Consequently, upon the integration of the AFF mechanism, it transforms the feature vectors for each traffic class in the prototype network into convex combinations of feature vectors generated from the two representation types. For each sample i, the computation of its feature vector e_i is as follows:

$$e_i = \alpha \cdot g(r_i) + (1 - \alpha) \cdot h(t_i) \tag{4}$$

where $h(t_i)$ represents the feature vector derived from statistical feature t_i after extraction, and $g(r_i)$ is the feature vector derived from binary flow data representation r_i after extraction. The two types of feature vectors should be generated within the same dimensional space to ensure that AFF correctly produces the final fused feature vector. In this equation, α is the adaptive fusion coefficient, and its magnitude depends on the feature vectors $h(t_i)$ of each sample. The specific calculation method is as follows:

$$\alpha = \frac{1}{1 + exp(-a(h(t_i)))} \tag{5}$$

where a denotes the adaptive fusion network. The generated α values range from (0, 1) and essentially represent the weight of the binary data stream feature vector $g(r_i)$ within the final feature vector e_i. Depending on the specific scenario, the value of the adaptive blending coefficient α can fluctuate, thereby modifying the reliance of e_i on the two types of feature vectors.

4.4. Overall Architecture

The overall architecture of AFF is depicted in Figure 4. AFF consists of two components: a feature extraction network and an adaptive fusion network. The feature extraction network comprises two distinct architectures, each responsible for extracting features from the sample's binary data stream representation and statistical features, generating feature vectors of the same dimensions. The adaptive fusion network uses the feature vector corresponding to the statistical features as input and outputs an adaptive fusion coefficient within the range (0,1). This coefficient is used to generate the sample's final feature vector according to Formula 4. The progressive operation of these two constituents, situated between the intrusion detection model's input and the prototype network, leads to a subsequent improvement in the overall model's performance. The overall architecture of AFF is depicted in Figure 4.

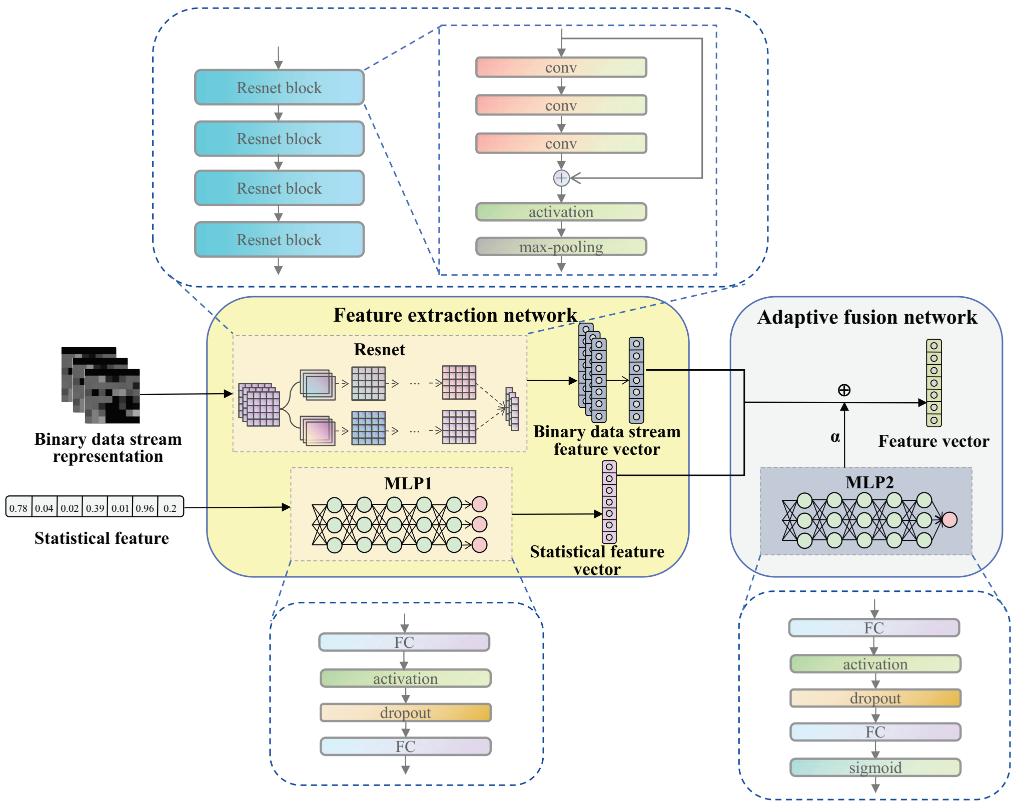

Figure 4. Overall architecture of AFF.

Binary data stream feature extraction network. After data preprocessing, the representation of binary flow data essentially consists of 5 grayscale images, each with a size of 16 × 16 pixels. To expedite network convergence, enhance model accuracy, and improve generalization capabilities, we choose a Residual Network (ResNet) architecture over the traditional convolutional network structure for binary data stream feature vector extraction. Subsequently, a vector is generated for each data packet in the network traffic, which corresponds to a grayscale image. As the input representations are alignment processed, the multiple feature vectors obtained after passing through the ResNet maintain their alignment characteristics. This allows for dimension reduction by taking the average of the multiple feature vectors that constitute a three-dimensional tensor. Ultimately, this process yields the feature vector for binary flow data. The ResNet architecture is composed of four residual blocks, and each convolutional layer within these blocks has a kernel size of 5 × 5 and a stride of 1. Rectified Linear Unit (ReLU) serves as the activation function, while max-pooling layers with a kernel size of 2 and a stride of 2 are present only in the first three residual blocks.

Statistical feature extraction network. The input statistical feature are represented as a 76-dimensional vector. The feature extraction network for statistical features is a nonlinear multilayer perceptron (MLP), composed of fully connected layers, activation functions, dropout layers, and additional fully connected layers as shown in Figure 4. The activation function used is ReLU. The statistical features are finally transformed into a 512-dimensional feature vector, which matches the dimensionality of the binary data stream feature vector.

Adaptive fusion network. The adaptive fusion network also employs an MLP with a network structure similar to that of the statistical feature's feature extraction network. However, a sigmoid function is added before the output, allowing the input 512-dimensional statistical feature vector to generate an adaptive fusion coefficient α within the range (0,1). Following the convex combination of the two types of feature vectors, each sample's feature vector is used as input for the prototypical network.

4.5. Training Strategy

To illustrate, Algorithm 1 provides insight into how the prototypical network operates with the introduction of the Adaptive Feature Fusion Mechanism.

Algorithm 1: Training an episode for the prototypical network with AFF. M is the total number of classes in the training set. N is the number of classes in each episode. K is the number of supports for each class. K_Q is the number of queries for each class. R and T are the feature vector sets for binary data stream and statistical features.

```
Data: Training set $D_{train} = \{((r_i, t_i), y_i)\}$ where $y_i \in \{1, ..., M\}, r_i \in R, t_i \in T$.
      $D^k_{train} = \{((r_i, t_i), y_i), y_i = k\}$.
Result: Episode loss $L(\theta)$ for the sampled episode.
$V \leftarrow RandomSample(\{1, ..., M\}, N)$;
for $k$ in $V$ do
    $S_k \leftarrow RandomSample(D^k_{train}, K)$;
    $Q_k \leftarrow RandomSample(D^k_{train} \backslash S_k, K_Q)$;
    $S^{'}_k \leftarrow \emptyset$;
    for $(r_i, t_i)$ in $S_k$ do
        $\alpha \leftarrow \frac{1}{1+exp(-a(h(t_i)))}$;
        $e_i \leftarrow \alpha \cdot g(r_i) + (1-\alpha) \cdot h(t_i)$;
        $S^{'}_k \leftarrow S^{'}_k + (e_i, y_i)$;
    end
    $Q^{'}_k \leftarrow \emptyset$;
    for $(r_i, t_i)$ in $Q_k$ do
        $\alpha \leftarrow \frac{1}{1+exp(-a(h(t_i)))}$;
        $e_i \leftarrow \alpha \cdot g(r_i) + (1-\alpha) \cdot h(t_i)$;
        $Q^{'}_k \leftarrow Q^{'}_k + (e_i, y_i)$;
    end
    $P_k = \frac{1}{N} \sum_{(e_i, y_i) \subset S^{'}_k} f_\phi(e_i)$
end
$L(\theta) \leftarrow 0$;
for $k$ in $\{1, ..., N\}$ do
    for $(e_i, y_i)$ in $Q^{'}_k$ do
        $L(\theta) \leftarrow L(\theta) + \frac{1}{N \times K}[d(f_\phi(e_i), P_k) + log\sum_N exp(-d(f_\phi(e_i), P_k))]$;
    end
end
```

5. Experiments and Analysis

5.1. Datasets

Due to the lack of datasets specifically generated for few-shot learning scenarios in the field of intrusion detection, a common practice is to select widely used datasets from the network security domain as the foundation and construct few-shot tasks by redefining samples. In order to evaluate the proposed model, we select the CICIDIS2017 [25] and ISCX2012 [26] datasets as benchmarks, with the following reasons:

1. To achieve the acquisition of two different data representations, the chosen datasets should provide unprocessed network traffic files rather than pre-extracted feature vectors.
2. The selected datasets have corresponding labels for each sample, which simplifies the process of model learning and metric calculation.
3. The selected datasets encompass a variety of protocols and types of attacks, making it possible to simulate real-world network attack scenarios effectively.

These datasets are widely recognized for their comprehensive coverage of various attack types and protocols, providing a solid foundation for training and evaluating intrusion detection systems. After filtering and reconstruction, normal traffic and five attack types from the CICIDS2017 dataset, along with normal traffic and four attack types from the ISCX2012 dataset, are combined into a new dataset without altering or selecting samples. Detailed information about the two reconstructed datasets is provided in Table 1.

Table 1. Coding and numbers for the types of traffic.

Benchmark	Code	Type of Traffic	Numbers
CICIDS2017	$DDoS_c$	DDoS with LOIT	45,168
	DoS_c	DoS(All types)	29651
	FTP_c	FTP-BruteForce	3958
	$Infiltration_c$	Infiltration	66,913
	SSH_c	SSH-BruteForce	2464
	$Benign_c$	Benign	241,041
ISCX2012	$DDoS_i$	DDoS	19,282
	DoS_i	HTTP DoS	3204
	$Infiltration_i$	Infiltration	8984
	SSH_i	SSH-BruteForce	4957
	$Benign_i$	Benign	168,878

For the convenience of presenting and analyzing the results of subsequent experiments, we assign specific codes to each type of traffic, with the main code representing the traffic category and the subscript indicating its original dataset. The subscript "i" indicates data from ISCX2012, while the subscript "c" indicates data from CICIDS2017.

5.2. Metrics

Intrusion detection is fundamentally a classification problem, making accuracy the most commonly used performance metric. Accuracy measures the proportion of correctly classified samples against the total test samples, reflecting the model's predictive capability. Recall is also critical in this domain, indicating the ratio of true positives to the total positives identified by the model, which is essential for assessing the detection of malicious samples. In multi-class experiments, we use micro-precision and micro-recall, which provide the average precision and recall across all categories as key evaluation metrics.

5.3. Experimental Settings

In order to comprehensively and deeply investigate the proposed approach, we design three categories of experiments with different objectives.

The first category of experiments aims to explore whether the prototype network incorporating the AFF mechanism can effectively address intrusion detection problems in a few-shot scenario and whether the proposed method outperforms existing solutions. To this end, the first category of experiments is conducted based on the restructured CICIDS2017 dataset and is divided into two settings: binary classification tasks and multi-class classification tasks. In the binary classification tasks, we select four attack categories from the dataset as the data source and 70% of the benign samples for the meta-training set, reserving one category for the meta-test set. The test set comprises all attacks, including the few-shot attack category, along with the remaining normal data flow samples. By choosing different attack categories as the source of the test set, as the number of combinations C(5,1) = 5, we can construct five parallel experiments. In order to explore the impact of different sample sizes K on detection performance, this experiment is conducted under the settings of $K = 1$ and $K = 5$.

For the multi-classification tasks of intrusion detection, our objective is for the model to correctly classify traditional attacks with a substantial number of samples and, concurrently, accurately identify categories of attacks with limited samples. Hence, we opt for three types of attacks, along with normal data flow samples, to compose the meta-training set. Additionally, we select one attack category to simulate a scenario with few shots for constructing the dataset. In the multi-class classification tasks, a total of 5 experiments are conducted, and the sources of meta-training and meta-testing data for each experiment are shown in Table 2. Similar to the binary classification tasks, the multi-class classification tasks are also performed under the settings of $K = 1$ and $K = 5$.

Table 2. Multi-classification tasks settings.

Number	Meta-Testing	Meta-Training
1	DDoS$_c$	FTP$_c$ Infiltration$_c$ SSH$_c$ Benign$_c$
2	DoS$_c$	DDoS$_c$ Infiltration$_c$ SSH$_c$ Benign$_c$
3	FTP$_c$	DDoS$_c$ DoS$_c$ SSH$_c$ Benign$_c$
4	Infiltration$_c$	DDoS$_c$ DoS$_c$ FTP$_c$ Benign$_c$
5	SSH$_c$	DoS$_c$ FTP$_c$ Infiltration$_c$ Benign$_c$

The second category of experiments aims to investigate the effectiveness of the AFF mechanism, examining whether the adaptive fusion mechanism can indeed enhance the performance of the prototype network. We conduct ablation experiments on binary and multi-class classification tasks under the $K = 5$ setting using the restructured CICIDS2017 dataset to observe the enhancement effect of the AFF mechanism on the prototype network.

The third category of experiments aims to explore the feasibility of the proposed model in practical applications. We conduct cross experiments on the restructured CICIDS2017 and ISCX2012 datasets. By selecting one dataset as the source of meta-training and meta-testing while using the other to simulate real-world data, we observe the performance of the proposed model when the source of meta-training and meta-testing samples differs. This experiment is conducted under the $K = 5$ setting and includes binary and multi-class classification tasks.

5.4. Classification Experiments on the Reconstructed CICIDS2017 Dataset

The test results for binary classification tasks in the first category of experiments are shown in Tables 3 and 4. As shown below, the proposed method performs well in both binary and multi-class classification intrusion detection tasks, consistently maintaining high accuracy and recall values during testing. The value of the adaptive mixing coefficient α is shown in Table A1 in the Appendix A under various experimental settings.

Table 3. Detection results of binary classification tasks on reconstructed CICIDS2017 dataset.

Few-Shot Type	K = 1		K = 5	
	Acc (%)	Rec (%)	Acc (%)	Rec (%)
DDoS$_c$	99.34	99.57	99.74	99.84
DoS$_c$	97.30	98.59	99.76	99.55
FTP$_c$	99.98	99.99	99.98	99.99
Infiltration$_c$	93.49	96.16	97.74	98.26
SSH$_c$	99.97	99.99	99.99	99.99
Overall	97.62	98.86	99.44	99.53

Table 4. Detection results of multi-classification tasks on reconstructed CICIDS2017 dataset.

Few-Shot Type	**K = 1**		**K = 5**	
	Acc (%)	**Rec (%)**	**Acc (%)**	**Rec (%)**
$DDoS_c$	98.87	98.93	99.15	99.32
DoS_c	95.35	97.58	97.15	98.32
FTP_c	99.92	99.98	99.93	99.99
$Infiltration_c$	94.85	95.02	98.71	99.56
SSH_c	99.89	99.92	99.95	99.99
Overall	97.78	98.29	98.98	99.44

An analysis of these two tables leads to the following conclusions:

1. The choice of few-shot types has a noticeable impact on the experimental results, a phenomenon that is consistent in both binary and multi-class tasks. For example, selecting SSH_c as the few-shot type results in favorable performance in both binary and multi-class tasks, while choosing $Infiltration_c$ leads to a significant decrease in accuracy and recall in both types of experiments.
2. The selection of the K value has a clear influence on the results. When K is increased from 1 to 5, various data groups in both types of experiments show varying degrees of improvement. Such results are reasonable since more samples imply that the model can extract more useful information, providing a more reliable basis for classifying training set samples.

Subsequently, we compare our proposed method with existing research that uses the same benchmark dataset, CICIDS2017. The results are presented in Tables 5 and 6. Table 5 demonstrates that, in binary classification tasks, the prototype network incorporating the AFF mechanism stands out among existing few-shot intrusion detection methods, displaying a clear advantage.

Intrusion detection for multi-class classification in few-shot scenarios is a relatively recent research area, with few existing comparative studies available for reference. As illustrated in Table 6, besides the approach proposed in this paper and SPN and Cs, the other methods rely on a large volume of samples for training and do not utilize few-shot learning techniques. Table 6 further illustrates that, even when working with highly constrained sample sizes, the method outlined in this paper consistently delivers commendable detection performance. This accomplishment carries paramount significance within the dynamic landscape of real network environments characterized by the emergence of novel attack scenarios.

Table 5. Comparison of binary classification detection results and the number of samples in the proposed method and related research works.

Method	**Feature Type**	**Number**	**Acc (%)**	**Rec (%)**
Siamese Capsule [26]	Binary Data Stream	5	93.87	96.45
Siamese Capsule [26]	Binary Data Stream	20	95.56	98.31
FC-Net [18]	Binary Data Stream	5	94.33	99.17
FC-Net [18]	Binary Data Stream	10	94.64	99.62
FCAD [27]	Statistical	5	94.30	N/A
FCAD [27]	Statistical	10	97.10	N/A
FCAD [27]	Statistical	20	98.60	N/A
FS-IDS [11]	Binary Data Stream + Statistical	5	97.51	99.00
Few-Shot with L2F [19]	Statistical	10	94.66	96.68
AFF-ProtoNets	Binary Data Stream + Statistical	1	97.62	98.86
AFF-ProtoNets	Binary Data Stream + Statistical	5	99.44	99.53

Table 6. Comparison of multi-classification detection results and the number of samples in the proposed method and related research works.

Method	Feature Type	Number	Acc (%)	Rec (%)
DNN	Statistical	93,500	97.71	96.64
CFS-BA-ensemble [28]	Statistical	125,973	99.30	99.21
MAGNETO-B [29]	Binary Data Stream + Statistical	100,000	99.28	N/A
MAGNETO-S [29]	Binary Data Stream + Statistical	100,000	99.24	N/A
CES-IDS [30]	Binary Data Stream + Statistical	137,183	98.67	N/A
LSTM-CNN-MAM [31]	Binary Data Stream + Statistical	20,562	99.92	99.78
MEMBER [32]	Binary Data Stream + Statistical	222,288	99.42	N/A
Graph2vec [33]	Statistical	16,379	99.63	99.36
SPN [34]	Binary Data Stream	5	93.78	92.44
Cs [35]	Statistical	1000	95.20	N/A
AFF-ProtoNets	Binary Data Stream + Statistical	1	97.78	98.29
AFF-ProtoNets	Binary Data Stream + Statistical	5	98.98	99.44

5.5. AFF Effectiveness Experiments

To evaluate the effectiveness of the AFF mechanism, a series of ablation experiments is conducted under consistent data preprocessing conditions. With K set to 5, the experimental results for both the prototype network and the prototype network integrated with the AFF mechanism in binary and multi-class tasks are presented in Figure 5. Figure 5 illustrates the comparison between two models: one that employs the AFF mechanism, integrating two types of features (left), and another that relies solely on binary data stream (right). In each heatmap, the vertical axis represents the types of meta-testing traffic in the experiment, while the horizontal axis indicates the types of traffic involved. The values in the heatmap range from 0.95 to 1.00, with darker colors signifying higher classification accuracy for the corresponding types of traffic.

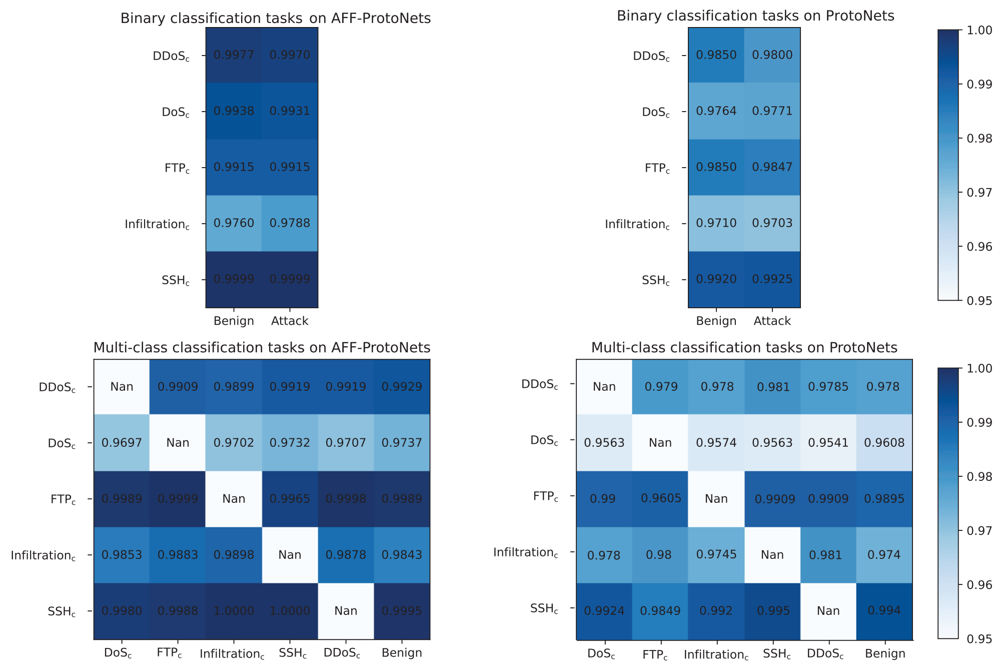

Figure 5. Accuracy of ablation experiments for binary and multi-class tasks. This figure shows the accuracy results from the ablation experiments.

Figure 5 clearly demonstrates a substantial improvement in the performance of the prototype network across both binary and multi-class tasks due to the presence of the AFF mechanism, as indicated by the darker colors in the heatmap on the left compared to the one on the right. This enhanced accuracy in traffic classification for both tasks can be explained by two key perspectives that highlight the overall improvements across different types of samples within each task:

1. From the perspective of the fundamental reasons for few-shot learning shortcomings, a key factor is the information loss due to insufficient samples. The AFF mechanism addresses this by integrating two types of features, effectively supplementing the missing information and optimizing the overall model.
2. From a theoretical standpoint, the prototype network without the AFF mechanism serves as a special case of our proposed model, relying exclusively on binary data stream for traffic classification. Consequently, the introduction of the AFF mechanism guarantees that the overall performance will be at least as strong as that of the model that relies solely on a single type of feature.

5.6. Feasibility Experiments

To simulate the real-world application of our trained model, we design a series of cross-experiments based on the restructured CICIDS2017 and ISCX2012 datasets. Under the condition of K = 5, we select four attack types' traffic from the CICIDS2017 dataset along with normal traffic to form the meta-training set, while we use the entire range of traffic types from the ISCX2012 dataset as few-shot types for testing. The results for both binary and multi-class tasks are illustrated in Figure 6.

The left image in Figure 6 illustrates the model's performance in the binary classification scenario, where the test set includes malicious traffic samples of DDoS$_i$, DoS$_i$, Infiltration$_i$, and SSH$_i$. The right image represents the results of the multi-class experiment under the same experimental settings. Even in the multi-class experiment of the cross-validation, the overall model maintains a commendable performance, achieving an accuracy of 90.36%. It is noteworthy that the cross-experiment is conducted under extreme conditions with $K = 5$, testing the model on a new dataset with significant variations from the training dataset. Therefore, the results of the cross-experiment suggest that the model exhibits robust stability and superior performance in environments with substantial differences.

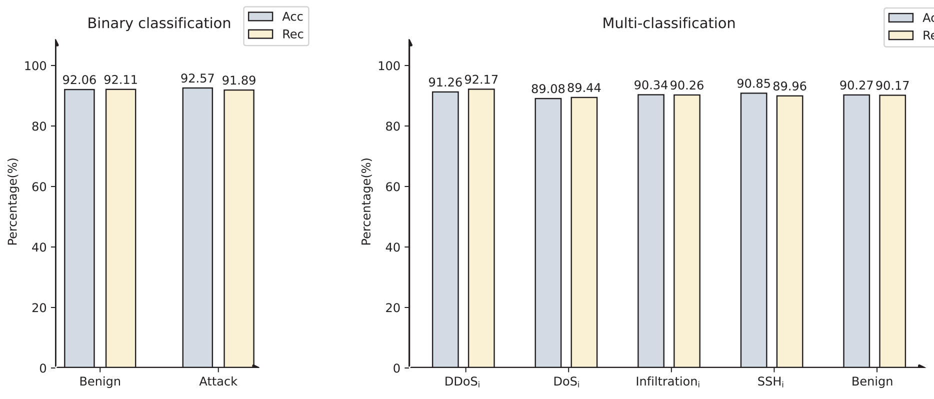

Figure 6. Accuracy and recall of feasibility experiments on the reconstructed ISCX2012 dataset.

6. Applicability

In this subsection, we further clarify the applicability of the proposed approach and the experimental design, highlighting the following two limitations:

- Unencrypted traffic samples. The research and experiments presented in this study are predicated on the assumption that the traffic samples utilized are unencrypted. In instances where traffic is encrypted, the information contained within the payload section of the binary data stream representation may be compromised, as the underlying data become inaccessible for analysis. This limitation poses a significant challenge for intrusion detection systems, potentially diminishing the effectiveness of our proposed method in such scenarios. Furthermore, it is essential to highlight that the traffic samples from the CICIDS2017 and ISCX2012 datasets employed in the experiments are also unencrypted; consequently, the results obtained remain uncertain in the context of encrypted traffic and warrant further investigation.

- Features of samples are limited. To ensure real-time performance in intrusion detection systems, we extract features by selecting the first five data packets from each network traffic flow and the initial 256 bytes from each packet. This selection is informed by findings from [22], which suggest that header information (such as IP and TCP headers) and the early data within the payload are typically the most informative for sample classification. However, this methodology entails certain limitations; specifically, it may lead to the loss of critical information for some malicious payloads that are present later in the network traffic flow or that extend beyond the first 256 bytes. Such limitations could significantly affect the classification accuracy for these samples.

7. Conclusions

In the present paper, we introduced a metric-based meta-learning framework designed to enhance few-shot intrusion detection for both binary and multi-class classification tasks. By employing metric-based meta-learning to create prototypes after feature extraction and integrating an Adaptive Feature Fusion mechanism, our approach effectively reduces information loss, a significant challenge in few-shot learning scenarios.

The results indicate that the proposed method achieves a detection rate of 98.98% with five-shot samples, outperforming existing few-shot learning techniques. Additionally, ablation studies confirm that the AFF mechanism enhances accuracy by mitigating information loss. Cross-validation on the CICIDS2017 and ISCX2012 datasets further supports the model's robustness, with accuracy rates of 92.32% and 90.36% for binary and multi-class tasks, respectively.

Despite these findings, further evaluation of the model's performance in real-world scenarios is needed to assess its generalization capabilities. Moreover, addressing the challenge of completely unseen samples presents an important area for future research. Our future work will focus on improving generalization through methods like meta-regularization and domain adaptation, as well as exploring zero-shot learning to enhance the model's applicability in dynamic environments. These efforts aim to further advance the practical utility of our approach in real-world intrusion detection systems.

Author Contributions: Conceptualization, J.B. and K.C.; methodology, J.B.; software, J.B.; validation, J.B. and P.G.; formal analysis, K.C. and P.G.; investigation, J.B. and S.L.; resources, S.L.; writing—original draft preparation, J.B.; writing—review and editing, K.C.; visualization, J.B.; supervision, K.C. and S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

In this section, Table A1 provides the values of the adaptive mixing coefficient α under various experimental settings. The table reveals that, depending on the specific configurations, which encompass different classification tasks, K values, and the selection of few-shot traffic types, the adaptive mixing coefficient α is dynamically adjusted to suit the circumstances.

Table A1. The values of the adaptive mixing coefficient α under various experimental settings.

α	Binary Classification		Multi-Classification	
	K = 1	K = 5	K = 1	K = 5
DDoS_c	0.9549	0.7508	0.7138	0.6353
DoS_c	0.5090	0.5384	0.7034	0.7911
FTP_c	0.9419	0.4798	0.9095	0.7258
Infiltration_c	0.4349	0.4206	0.6480	0.6047
SSH_c	0.9511	0.4666	0.9281	0.7590

References

1. Illavarason, P.; Sundaram, B.K. A Study of Intrusion Detection System using Machine Learning Classification Algorithm based on different feature selection approach. In Proceedings of the 2019 Third International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC), Palladam, India, 12–14 December 2019; pp. 295–299.
2. Patel, A.; Qassim, Q.; Wills, C. A survey of intrusion detection and prevention systems. *Inf. Manag. Comput. Secur.* **2010**, *18*, 277–290. [CrossRef]
3. Bilge, L.; Dumitraş, T. Before we knew it: An empirical study of zero-day attacks in the real world. In Proceedings of the 2012 ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 16–18 October 2012; pp. 833–844.
4. Douze, M.; Szlam, A.; Hariharan, B.; Jégou, H. Low-shot learning with large-scale diffusion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 3349–3358.
5. Li, A.; Luo, T.; Lu, Z.; Xiang, T.; Wang, L. Large-scale few-shot learning: Knowledge transfer with class hierarchy. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 7212–7220.
6. Ren, M.; Triantafillou, E.; Ravi, S.; Snell, J.; Swersky, K.; Tenenbaum, J.B.; Larochelle, H.; Zemel, R.S. Meta-learning for semi-supervised few-shot classification. *arXiv* **2018**, arXiv:1803.00676.
7. Bhatia, V.; Choudhary, S.; Ramkumar, K.R. A comparative study on various intrusion detection techniques using machine learning and neural network. In Proceedings of the 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 4–5 June 2020; pp. 232–236.
8. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
9. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. *IEEE Access* **2017**, *6*, 1792–1806. [CrossRef]
10. Qiu, W.; Ma, Y.; Chen, X.; Yu, H.; Chen, L. Hybrid intrusion detection system based on Dempster-Shafer evidence theory. *Comput. Secur.* **2022**, *117*, 102709. [CrossRef]
11. Ouyang, Y.; Li, B.; Kong, Q.; Song, H.; Li, T. FS-IDS: A Novel Few-Shot Learning Based Intrusion Detection System for SCADA Networks. In Proceedings of the ICC 2021-IEEE International Conference on Communications, Montreal, Canada, 14–23 June 2021; pp. 1–6. [CrossRef]
12. Wang, Y.; Yao, Q.; Kwok, J.; Ni, L. Few-shot learning: A survey. *arXiv* **2019**, arXiv:1904.05046.
13. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [CrossRef]
14. Duan, R.; Li, D.; Tong, Q.; Yang, T.; Liu, X.; Liu, X. A survey of few-shot learning: An effective method for intrusion detection. *Secur. Commun. Netw.* **2021**, *2021*, 4259629. [CrossRef]
15. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
16. Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D. Matching networks for one shot learning. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 1–10.
17. Snell, J.; Swersky, K.; Zemel, R. Prototypical networks for few-shot learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 114. [CrossRef]
18. Xu, C.; Shen, J.; Du, X. A method of few-shot network intrusion detection based on meta-learning framework. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 3540–3552.
19. Shi, Z.; Xing, M.; Zhang, J.; Wu, B.H. Few-Shot Network Intrusion Detection Based on Model-Agnostic Meta-Learning with L2F Method. In Proceedings of the 2023 IEEE Wireless Communications and Networking Conference (WCNC), Glasgow, UK, 26–29 March 2023; pp. 1–6. [CrossRef]
20. Oreshkin, B.; Rodríguez López, P.; Lacoste, A. TADAM: Task dependent adaptive metric for improved few-shot learning. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
21. Xing, C.; Rostamzadeh, N.; Oreshkin, B.; Pinheiro, P.O. Adaptive Cross-Modal Few-shot Learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32.
22. Holland, J.; Schmitt, P.; Feamster, N.; Mittal, P. New directions in automated traffic analysis. In Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, 15–19 November 2021; pp. 3366–3383.

23. Yang, J.; Li, H.; Shao, S.; Zou, F.; Wu, Y. FS-IDS: A framework for intrusion detection based on few-shot learning. *Comput. Secur.* **2022**, *122*, 102899. [CrossRef]
24. Kim, T.; Suh, S.C.; Kim, H.; Kim, J.; Kim, J. An encoding technique for CNN-based network anomaly detection. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 2960–2965. [CrossRef]
25. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* **2018**, *1*, 108–116. [CrossRef]
26. Shiravi, A.; Shiravi, H.; Tavallaee, M.; Ghorbani, A.A. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Comput. Secur.* **2012**, *31*, 357–374. [CrossRef]
27. Wang, Z.M.; Tian, J.Y.; Qin, J.; Fang, H.; Chen, L.M. A few-shot learning-based siamese capsule network for intrusion detection with imbalanced training data. *Comput. Intell. Neurosci.* **2021**, *2021*, 7126913. [CrossRef]
28. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [CrossRef]
29. Andresini, G.; Appice, A.; De Rose, L.; Malerba, D. GAN augmentation to deal with imbalance in imaging-based intrusion detection. *Future Gener. Comput. Syst.* **2021**, *123*, 108–127. [CrossRef]
30. Gupta, N.; Jindal, V.; Bedi, P. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. *Comput. Secur.* **2022**, *112*, 102499. [CrossRef]
31. Liu, J.; Song, X.; Zhou, Y.; Peng, X.; Zhang, Y.; Liu, P.; Wu, D.; Zhu, C. Deep anomaly detection in packet payload. *Neurocomputing* **2022**, *485*, 205–218. [CrossRef]
32. Lan, J.; Liu, X.; Li, B.; Sun, J.; Li, B.; Zhao, J. MEMBER: A multi-task learning model with hybrid deep features for network intrusion detection. *Comput. Secur.* **2022**, *123*, 102919. [CrossRef]
33. Hu, X.; Gao, W.; Cheng, G.; Li, R.; Zhou, Y.; Wu, H. Towards early and accurate network intrusion detection using graph embedding. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 5817–5831. [CrossRef]
34. Miao, G.; Wu, G.; Zhang, Z.; Tong, Y.; Lu, B. SPN: A method of few-shot traffic classification with out-of-distribution detection based on Siamese Prototypical Network. *IEEE Access* **2023**, *11*, 114403–114414. [CrossRef]
35. He, J.; Yao, L.; Li, X.; Khan, M.K.; Niu, W.; Zhang, X.; Li, F. Model-agnostic generation-enhanced technology for few-shot intrusion detection. *Appl. Intell.* **2024**, *54*, 3181–3204. [CrossRef]