A Meta-learning Approach for Few-shot Network Intrusion Detection Using Depthwise Separable Convolution

Guo Li[1,*] and MingHua Wang[2]

[1]*College of Intelligent Manufacturing and Electrical Engineering, Nanyang Normal University, Nanyang, Henan 473000, China*
[2]*Shandong Gete Aviation Technology Co., Ltd, Jinan, ShanDong 250000, China*
E-mail: liguo198011@163.com; 13064053502@163.com
[*]*Corresponding Author*

Received 18 December 2024; Accepted 14 February 2025

Abstract

As cyberattacks become more frequent and sophisticated, network intrusion detection systems (IDS) play a critical role in safeguarding networks. However, traditional IDS models face challenges in detecting new, unseen attacks and typically require large volumes of labeled data for effective training. To address these issues, we propose a novel intrusion detection model based on meta-learning, integrating depthwise separable convolution (DSC). This model leverages few-shot learning to detect rare and emerging attack types with minimal labeled data. By using meta-learning, our model can rapidly adapt to new tasks, offering greater flexibility and scalability in various network scenarios. Experimental results on the CIC-DDoS2019 and CIC-IDS2017 datasets demonstrate that our model achieves competitive accuracy compared to state-of-the-art methods, even with fewer training samples. It also shows superior performance in terms of both detection accuracy and training efficiency, while being more resource-efficient, making it suitable for

Journal of ICT Standardization, Vol. 12_4, 443–470.
doi: 10.13052/jicts2245-800X.1245

deployment in resource-constrained environments. In conclusion, our model offers a promising solution for network intrusion detection, enhancing the ability to detect new and emerging threats while ensuring computational efficiency for real-world applications.

Keywords: Network intrusion detection, meta-learning, depthwise separable convolution, few-shot learning, attack detection.

1 Introduction

With the rapid development of information technology, network security issues have become increasingly prominent, posing a significant global challenge. The methods of cyberattacks have become more complex and diverse, and traditional network security defense mechanisms often struggle to cope with the ever-changing threats [1, 2]. In particular, in the field of network intrusion detection, how to quickly identify and effectively respond to novel attacks in a complex and dynamic network environment has become one of the key research issues. A network intrusion detection system (NIDS) can effectively identify malicious behaviors by monitoring network traffic and analyzing data packets [3, 4]. However, traditional intrusion detection methods often fail when faced with unknown or complex attack patterns, especially when attackers use encryption, obfuscation, or other techniques to hide their actions, posing significant challenges to conventional detection mechanisms [5, 6].

Meta-learning, an innovative approach in machine learning, has recently attracted significant attention in network security for its ability to adapt quickly to new tasks or environments using limited sample data [7]. The core concept of meta-learning is to accumulate experience across multiple tasks so that the system can quickly adjust and optimize its learning strategy when faced with unknown problems, thereby improving its ability to respond to new types of network attacks [8, 9]. Applying meta-learning to network intrusion detection systems enables the system to not only quickly identify new attacks but also adjust its detection strategy within limited data and time, enhancing the flexibility and responsiveness of network security protection [10, 11]. Especially when facing few-shot complex network threats, the system can maintain high detection accuracy. Compared to traditional methods, meta-learning helps improve the system's dynamic adaptability, enabling it to rapidly adjust when confronted with different types of attacks.

Past research has primarily focused on static methods based on feature extraction and model training [12, 13]. However, these methods typically rely on large amounts of labeled data and perform poorly when dealing with new attack samples. Nevertheless, these methods still face challenges such as high computational overhead, poor model interpretability, and limited adaptability to small datasets or novel attacks [14, 15]. While deep learning models demonstrate powerful capabilities when processing large-scale data, they still have shortcomings in rapidly responding to new types of attacks and reducing false positives. Current research trends are gradually moving toward enhancing the self-adaptation and real-time response capabilities of intrusion detection systems [16]. Attempts based on transfer learning, incremental learning, and similar methods have provided some new ideas for solving these problems, but in practical applications these methods often rely on large amounts of historical data and still lack adaptability to different attack scenarios. In contrast, meta-learning, as a "learning to learn" approach, shows significant advantages in quickly adapting to new tasks and addressing few-shot problems, making it a promising solution to enhance the flexibility and robustness of network intrusion detection systems, particularly in complex and dynamic environments [17].

This paper presents a meta-learning-based model designed to improve the adaptability and real-time response of network intrusion detection systems against emerging attacks. By combining depthwise separable convolutions and a meta-learning framework, the model can quickly adapt to different types of attacks with few training samples and achieve efficient detection in the face of evolving network threats.

The main contributions of this paper include:

- We propose a meta-learning-based dynamic response intrusion detection framework that can quickly adapt to new attack patterns with limited samples, enhancing the system's ability to detect unknown attacks.
- We design an efficient meta-learning algorithm that optimizes the model's learning process based on network traffic features, thereby improving the accuracy of complex attack recognition and the system's real-time response capabilities.
- We perform comprehensive experiments to evaluate the performance of our proposed method under diverse attack scenarios. The results demonstrate that our approach achieves superior detection accuracy, lower false positive rates, and faster response times compared to conventional detection techniques.

2 Related Work

2.1 Research Progress of Network Intrusion Detection Systems (NIDS)

Network intrusion detection systems (NIDS) are important tools for protecting computer networks from malicious attacks. With the rapid development of network technologies and the continuous evolution of attack methods, the limitations of traditional NIDS approaches in handling novel attacks have gradually become apparent. Therefore, improving the accuracy, efficiency, and adaptability of intrusion detection systems has become a research hotspot in the field of network security in recent years [18].

Traditional intrusion detection methods are generally categorized into three types: rule-based detection, feature-based detection, and statistical-based detection. Rule-based detection methods rely on predefined attack signatures, matching data packets in network traffic with attack signature databases to identify potential intrusions [19, 20]. Due to the constant modification of attack strategies by attackers, and the variability of attack techniques, rule-based detection methods are unable to detect new or mutated attack patterns in a timely manner. With the rise of machine learning techniques, feature-based intrusion detection methods have gradually become mainstream [21]. These methods typically extract features from network traffic and then use machine learning algorithms (such as decision trees, support vector machines, k-nearest neighbors, etc.) for classification and decision-making [22, 23]. Compared to traditional rule-based detection methods, feature-based detection methods can improve the ability to recognize unknown attacks to some extent. In particular, through feature selection and feature engineering, useful patterns can be extracted from large amounts of network traffic, enabling detection systems to more accurately capture attack behaviors [24, 25]. However, feature-based detection methods still face several challenges, particularly in cases of data imbalance, feature redundancy, and attack pattern diversity, where the performance of existing models may significantly degrade [26].

Deep neural networks (DNN), convolutional neural networks (CNN), and recurrent neural networks (RNN) have been widely applied to NIDS, particularly in handling high-dimensional data and complex attack patterns, where deep learning methods have shown strong capabilities [27, 28]. Deep learning can automatically extract features from raw data, avoiding the complexity and manual intervention of traditional feature engineering. However, the training process of deep learning methods usually requires a large amount of labeled

data, and there are still significant challenges when facing data scarcity or rapidly changing attack patterns. Additionally, the black-box nature of deep learning models poses difficulties in interpretability and real-time response, especially in the field of network security, where quick and clear decision support is required [29]. To address the shortcomings of deep learning methods, some studies have started to explore network intrusion detection systems based on transfer learning. Transfer learning transfers knowledge from existing tasks to new tasks, reducing the need for labeled data and improving the model's adaptability to new environments [30]. Transfer learning has significant advantages in network security scenarios where data is scarce and the environment is highly variable. Although transfer learning can enhance the generalization ability of systems, existing transfer learning methods still cannot fully solve the adaptability issues in fast-changing attack scenarios due to the high dynamism of attack patterns and network environments.

In summary, existing NIDS methods perform well against known attacks but still have significant limitations when dealing with novel, unknown attacks and complex attack patterns. Therefore, how to improve the flexibility of intrusion detection systems, enhancing the system's real-time response capabilities, has become a major direction for current research. Future research needs to explore new learning strategies, particularly innovations in few-shot learning, adaptive learning, and real-time detection, to cope with the ever-evolving network security threats.

2.2 Application of Meta-learning in Network Security

With the continuous evolution of network attack methods, traditional network intrusion detection systems (NIDS) face numerous challenges, particularly in addressing novel attacks and few-shot learning tasks. Meta-learning, as a "learning how to learn" approach, has received widespread attention in the field of network security in recent years. Meta-learning allows models to quickly adapt to new tasks by accumulating experience from multiple tasks, particularly showing significant advantages in few-shot learning scenarios [31]. The complexity and variability of network attack patterns mean that traditional NIDS methods typically rely on large amounts of labeled data for training, making them often incapable of responding quickly to novel or mutated attacks [32]. In contrast, meta-learning can effectively reduce the reliance on large-scale labeled data by accumulating knowledge from multiple tasks, allowing for rapid adaptation when encountering new types of attacks [33].

In network intrusion detection, the application of meta-learning mainly focuses on enhancing the model's adaptability to new attack patterns. The diversity and constant change of network attack behaviors require NIDS to be highly flexible and capable of detecting new attack methods in real-time [34, 35]. Traditional intrusion detection systems typically perform pattern recognition based on specific training data, but this approach may lead to significant performance degradation when facing new attacks. Meta-learning, through the learning of "meta-tasks," allows models to accumulate experience from multiple attack tasks and quickly adjust learning strategies to deal with new attack types [36]. This makes meta-learning methods capable of quickly optimizing models and improving generalization in data-scarce scenarios through transfer learning [37]. Furthermore, meta-learning can also enhance the cross-domain adaptability of NIDS. Since attack patterns and features may vary across different network environments, traditional NIDS methods often struggle with poor cross-domain adaptability [37, 38]. Meta-learning allows models to be trained across multiple network environments, enabling them to learn shared features from different tasks, thus improving the model's adaptability in diverse environments [39]. For example, transfer learning can effectively transfer existing knowledge when addressing attacks in different network environments, reducing the dependence on environment-specific data and improving model applicability [40]. In terms of specific applications, many studies have attempted to combine meta-learning with deep learning, reinforcement learning, and other techniques to further improve NIDS performance. For instance, combining meta-learning with generative adversarial networks (GANs) can train models by generating synthetic attack samples, enhancing the model's ability to recognize novel attacks. By combining meta-learning with reinforcement learning, the model can not only recognize known attacks but also adjust detection strategies based on real-time feedback to cope with ever-changing attack patterns [41]. These methods, supported by the meta-learning framework, enable intrusion detection systems to make more precise decisions when facing complex attack scenarios. Moreover, meta-learning is closely related to continual learning, which aims to enable models to continuously optimize and update their capabilities as new data flows in. In the field of network security, attack patterns are dynamically changing, so NIDS systems must have the ability for continual learning to ensure timely responses to new attacks [42]. By integrating the advantages of meta-learning and continual learning, it can effectively avoid catastrophic forgetting, allowing models to retain the ability to detect old attacks while learning new ones, thereby improving the system's

long-term adaptability. Although meta-learning holds significant potential for application in network intrusion detection, several challenges remain in its practical implementation. First, the computational overhead of meta-learning algorithms can be high, especially when dealing with high-dimensional data, making the training process more complex and increasing system resource requirements. Second, maintaining high detection accuracy with a small number of samples, especially when facing previously unseen attacks, remains a critical issue. Additionally, the interpretability of meta-learning models is an important consideration, particularly in the field of network security, where the decision-making process of the system must be transparent and traceable. Therefore, improving the interpretability of meta-learning models while optimizing computational efficiency will be key directions for future research [43].

Overall, meta-learning provides a flexible and efficient solution for network intrusion detection, particularly demonstrating great potential in addressing novel attacks, cross-domain adaptability, and few-shot learning. With the continuous development of related technologies, meta-learning-based intrusion detection systems are expected to play a greater role in improving detection accuracy, reducing false positive rates, and enhancing real-time response capabilities in the future.

3 Method

3.1 Overview of the Model Structure

The structure of this model consists of two parts: the inner network uses a depthwise separable convolution network structure, while the outer layer introduces the meta-learning framework. The core of the inner structure is depthwise separable convolution (DSC), which reduces computational complexity and enhances feature extraction efficiency by decomposing the standard convolution operation into depthwise convolution and pointwise convolution. Figure 1 illustrates the structure of the proposed model. Specifically, depthwise convolution performs convolution operations independently on each input channel, reducing computational complexity. Pointwise convolution then performs a linear combination of the channels to fuse feature information across different channels, thereby more effectively extracting important features from network traffic. This structure not only reduces computational cost but also strengthens the model's performance, especially in scenarios with insufficient data or novel attacks, exhibiting stronger adaptability.

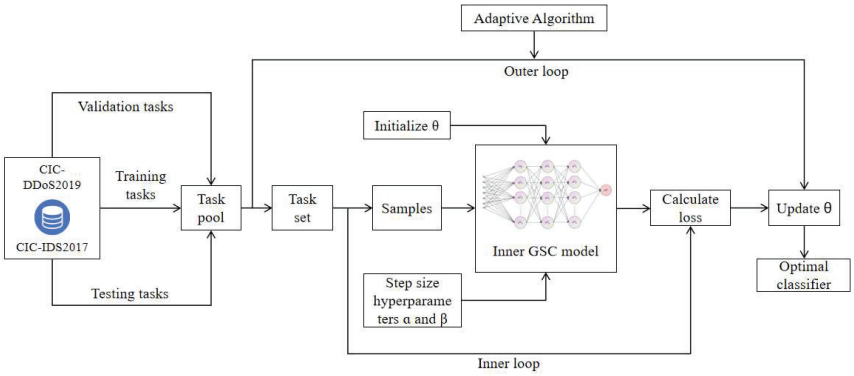

Figure 1 Overall framework of the proposed model.

In the outer layer of the model, we introduce the meta-learning framework, combined with adaptive optimization algorithms (such as Adam) to optimize task-specific parameter update rules. The main advantage of meta-learning is its ability to leverage prior experience, enabling the model to quickly adapt to new tasks. In the context of network intrusion detection, new attack patterns, and the constantly changing network environment require models to be flexible and adaptive. Meta-learning, through the "meta-training" process, continuously optimizes learning strategies, helping the model to react swiftly to different types of attacks and network environments. By employing meta-learning, the model is able to continually optimize shared knowledge across tasks, enhancing its generalization ability, and ensuring that the detection system can maintain high accuracy when facing novel or mutated attacks.

3.2 Depthwise Separable Convolution

DSC is the core component of the inner layer of our model, designed to reduce computational cost and enhance feature extraction capabilities. The standard convolution operation computes convolutions for every input–output channel pair, leading to a quadratic increase in computational cost, which becomes a bottleneck in deep neural networks. In contrast, depthwise separable convolution breaks this process into two stages: depthwise convolution, where each input channel is convolved with its own filter, and pointwise convolution, which uses a 1×1 filter to combine the outputs. This separation reduces the computational complexity significantly. In the depthwise convolution step, each input channel is processed independently

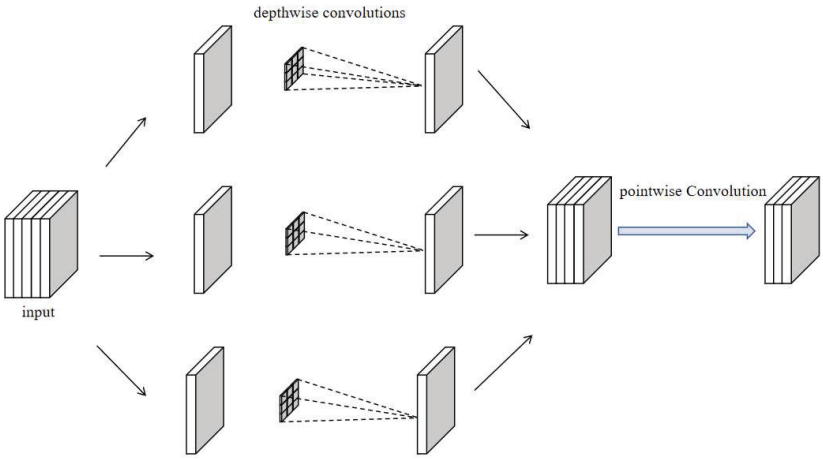

Figure 2 Depthwise separable convolution architecture, consisting of depthwise and pointwise convolutions to reduce computational complexity and model parameters.

with its corresponding filter. This reduces the computational cost by focusing on a single input channel at a time. Figure 2 illustrates the structure of the depthwise separable convolution.

Mathematically, depthwise separable convolution can be expressed as follows:

$$\mathrm{Y_d} = \mathrm{X} * \mathrm{W_d} \tag{1}$$

where $\mathrm{Y_d}$ is the output feature map after depthwise convolution, X is the input feature map and $\mathrm{W_d}$ is the depthwise convolution filter applied to each input channel independently. This step involves applying a 1×1 convolution across all the output channels, allowing for a linear combination of features. The pointwise convolution operation is represented by the following equation:

$$\mathrm{Y_p} = \mathrm{Y_d} * \mathrm{W_p} \tag{2}$$

where $\mathrm{Y_p}$ is the output feature map after pointwise convolution. Together, these two operations – depthwise convolution and pointwise convolution – combine to form depthwise separable convolution. The overall DSC operation can be expressed as:

$$\mathrm{Y} = \mathrm{X} * \mathrm{W_d} * \mathrm{W_p} \tag{3}$$

where Y represents the final output after both depthwise and pointwise convolutions. One of the primary advantages of DSC is the significant reduction

in the number of parameters and computational complexity compared to standard convolutions. The reduction in the number of parameters can be expressed mathematically as:

$$\text{Parameters}_{\text{DSC}} = (\text{C}_{\text{in}} \cdot \text{K}_{\text{d}}) + (\text{C}_{\text{out}} \cdot \text{K}_{\text{p}}) \tag{4}$$

where C_{in} is the number of input channels, C_{out} is the number of output channels, K_{d} is the kernel size for depthwise convolution, and K_{p} is the kernel size for pointwise convolution This reduction in parameters significantly reduces the computational burden, especially in deep networks with a large number of channels. In contrast, traditional convolution requires $\text{C}_{\text{in}} \cdot \text{C}_{\text{out}} \cdot \text{K}^2$ parameters (where K is the kernel size), which leads to a much larger number of parameters and higher computational cost.

By using DSC, we can achieve the same level of feature extraction performance while dramatically reducing the computational overhead, especially in large-scale networks or those with limited data. This efficiency is particularly valuable when dealing with limited labeled data or emerging attack patterns, where rapid adaptation and accurate feature extraction are crucial. To mitigate overfitting, we include a regularization term in the loss function:

$$\mathcal{L}_{\text{DSC}} = \lambda \cdot (\|\text{W}_{\text{d}}\|_2^2 + \|\text{W}_{\text{p}}\|_2^2) \tag{5}$$

where $\mathcal{L}_{\text{DSC}}$ is the regularization loss term, W_{d} and W_{p} are the depthwise and pointwise convolution filters, and λ is a regularization hyperparameter. This regularization helps maintain model simplicity and prevents overfitting, especially when dealing with limited labeled data.

Additionally, to compute the output size after applying depthwise separable convolution, we need to account for padding, stride, and kernel size. Given the input size $\text{H} \times \text{W}$, kernel size K, stride S, and padding P, the output size O for depthwise convolution (and the following pointwise convolution) can be computed as:

$$\text{O} = \left\lfloor \frac{\text{H} - \text{K} + 2\text{P}}{\text{S}} + 1 \right\rfloor. \tag{6}$$

This formula helps in determining the size of the output feature map after convolution, which is important for understanding how depthwise separable convolution impacts the network architecture.

By incorporating depthwise separable convolution into the inner layers of our model, we can capture complex features from network traffic while maintaining a low computational overhead. This efficiency is particularly

valuable when dealing with limited labeled data or emerging attack patterns, where rapid adaptation and accurate feature extraction are crucial.

3.3 Meta-learning Framework and Adaptive Optimization

By incorporating depthwise separable convolution into the inner layers of our model, we can capture complex features from network traffic while maintaining a low computational overhead. This efficiency is particularly valuable when dealing with limited labeled data or emerging attack patterns, as it allows the model to adapt quickly to new, unseen attacks. In this study, we introduce a meta-learning framework to enhance the adaptability of the network intrusion detection system, particularly when faced with new attack patterns and limited labeled data. Meta-learning focuses on training models to generalize across tasks, enabling rapid adaptation and optimization in new environments. To achieve this, we employ the model-agnostic meta-learning (MAML) algorithm, combined with adaptive optimization algorithms (such as Adam), to improve the model's learning efficiency and robustness in few-shot scenarios. During meta-training, the model learns how to extract common features across different network intrusion detection tasks by training on multiple tasks. Each task uses its samples to update the model parameters, allowing the model to adjust to similar task variations. This adaptation process enables the model to generalize better to unseen attacks and quickly learn to detect emerging threats.

Mathematically, the core of meta-learning is achieved through gradient updates for rapid adaptation. In each task $\mathrm{T_i}$, the model parameters θ undergo one or more gradient updates to adapt to the current task. The update rule for each gradient step is:

$$\theta_{\mathrm{i}}' = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_\theta) \tag{7}$$

where θ represents the updated model parameters, α is the learning rate, and $\mathcal{L}(\theta$,Ti) is the loss function for task $\mathrm{T_i}$, with $\nabla\theta$ representing the gradient of the loss function with respect to the model parameters. This step allows the model to adjust its parameters based on the current task's characteristics, effectively "learning to learn."

During meta-training, the objective is to minimize the loss across all tasks and optimize the initial parameters θ so that the model can quickly adapt to new tasks. The meta-objective function is:

$$\min_\theta \sum_{\mathrm{T_i} \sim \mathrm{p(T)}} \mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_{\theta - \alpha \nabla_\theta \mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_\theta)}) \tag{8}$$

where $\mathrm{p}(\mathrm{T})$ denotes the distribution of tasks, $\mathcal{L}_{\mathrm{T_i}}$ is the loss function for task $\mathrm{T_i}$, and θ is the model parameters with α representing the gradient update step size. The goal is to optimize the initial parameters θ so that they generalize well across tasks and allow the model to quickly adjust during meta-updating, minimizing the task-specific loss.

To achieve more efficient optimization, we combine the Adam optimizer, which dynamically adjusts the learning rate for each parameter, making the training process more stable. The update rules for the Adam optimizer are as follows:

$$\mathrm{m_t} = \beta_1 \mathrm{m_{t-1}} + (1-\beta_1)\nabla_\theta \mathcal{L}_\mathrm{t} \tag{9}$$

$$\mathrm{v_t} = \beta_2 \mathrm{v_{t-1}} + (1-\beta_2)(\nabla_\theta \mathcal{L}_\mathrm{t})^2 \tag{10}$$

$$\widehat{\mathrm{m}}_\mathrm{t} = \frac{\mathrm{m_t}}{1-\beta_1^\mathrm{t}}, \quad \widehat{\mathrm{v}}_\mathrm{t} = \frac{\mathrm{v_t}}{1-\beta_2^\mathrm{t}} \tag{11}$$

$$\theta_\mathrm{t} = \theta_{\mathrm{t-1}} - \eta \frac{\widehat{\mathrm{m}}_\mathrm{t}}{\sqrt{\widehat{\mathrm{v}}_\mathrm{t}} + \epsilon} \tag{12}$$

where $\mathrm{m_t}$ and $\mathrm{v_t}$ are estimates of the first and second moments, β_1 and β_2 are the decay rates of the momentum, and $\widehat{\mathrm{m}}_\mathrm{t}$ and $\widehat{\mathrm{v}}_\mathrm{t}$ are bias-corrected estimates. These updates ensure that the model's parameters are fine-tuned quickly, which is crucial for rapid adaptation.

By combining the meta-learning framework with adaptive optimization, the model rapidly adjusts and optimizes its parameters when confronted with new tasks. Each gradient update is based on the features of the current task, enabling the model to quickly respond to new attacks and adjust its detection strategy. Through continuous training and fine-tuning, the model improves its ability to recognize and adapt to emerging attacks. Meta-learning helps the model develop a set of parameters that generalize well across various scenarios, while adaptive optimization ensures these parameters are fine-tuned effectively. This approach enhances the model's accuracy and response times, even with limited labeled data, allowing it to detect known attacks efficiently and quickly identify unknown ones, providing real-time security protection in dynamic and evolving network environments.

3.4 Loss Function

In our model, the loss function is designed to optimize the performance of the intrusion detection system under the meta-learning framework, allowing

it to quickly adapt to new attack types with minimal data. The overall loss function is composed of two main components: the task-specific loss used in the inner loop for model adaptation and the meta-training loss used in the outer loop for model optimization.

The task-specific loss for each individual intrusion detection task $\mathrm{T_i}$ is based on cross-entropy loss, which is commonly used for binary classification tasks such as detecting normal or malicious network traffic. The loss for a single task is defined as:

$$\mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_\theta) = -\sum_{\mathrm{j}=1}^{\mathrm{N}} (\mathrm{y_j log}(\mathrm{f}_\theta(\mathrm{x_j})) + (1-\mathrm{y_j})\mathrm{log}(1-\mathrm{f}_\theta(\mathrm{x_j}))) \tag{13}$$

where $\mathrm{f}_\theta(\mathrm{x_j})$ is the predicted probability that $\mathrm{x_j}$ is an attack, $\mathrm{y_j}$ is the true label (1 for attack, 0 for normal), and N is the number of samples in task $\mathrm{T_i}$.

During meta-training, the model learns a set of parameters θ that generalizes across different intrusion detection tasks. To achieve this, we optimize the model's performance across multiple tasks using the following meta-training objective:

$$\mathcal{L}_{\mathrm{meta}}(\theta) = \sum_{\mathrm{T_i}\sim\mathrm{p(T)}} \mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_{\theta_\mathrm{i}'}) \tag{14}$$

where $\mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_{\theta_\mathrm{i}'})$ is the loss for task $\mathrm{T_i}$ after the model parameters θ are adapted for the specific task via gradient updates.

To update the model parameters, we use a gradient descent-based optimization method. The model parameters are updated by:

$$\theta_\mathrm{i}' = \theta - \alpha\nabla_\theta\mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_\theta) \tag{15}$$

where α is the learning rate used for the inner loop update.

The meta-update is performed by optimizing the initial parameters θ to ensure they are well-suited for rapid adaptation across tasks. The meta-update step involves adjusting the parameters based on the gradients from multiple tasks:

$$\theta \leftarrow \theta - \beta\nabla_\theta \sum_{\mathrm{T_i}\sim\mathrm{p(T)}} \mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_{\theta_\mathrm{i}'}) \tag{16}$$

where β is the meta-step size or learning rate. Additionally, to prevent overfitting and improve the generalization ability of the model, a regularization term is introduced. This term penalizes large model parameters and is represented by the L2 norm of θ:

$$\mathrm{R}(\theta) = \lambda\|\theta\|^2 \tag{17}$$

where λ is the regularization hyperparameter that controls the strength of the penalty Finally, the overall meta-objective, which combines the task-specific loss, meta-update loss, and regularization, is:

$$\mathcal{L}_{\text{meta-reg}}(\theta) = \sum_{\mathrm{T_i} \sim \mathrm{p(T)}} \mathcal{L}_{\mathrm{T_i}}(\mathrm{f}_{\theta_i'}) + \lambda \|\theta\|^2. \tag{18}$$

This objective ensures that the model is optimized for both fast adaptation and robust generalization across a variety of network intrusion tasks.

4 Experiment

4.1 Experimental Environment

The experiments were conducted on a system with an Intel Core i7-10700K processor (8 cores, 16 threads, 3.8 GHz), NVIDIA GeForce RTX 3080 GPU (10GB GDDR6X), 32 GB of DDR4 RAM, and a 1 TB NVMe SSD. The software environment included Ubuntu 20.04 LTS, Python 3.8, and deep learning frameworks such as PyTorch 1.10.0 and TensorFlow 2.6.0. Libraries like NumPy 1.21.2, SciPy 1.7.1, and scikit-learn 0.24.2 were used for data processing and evaluation. The Adam optimizer and CUDA 11.2 were employed to ensure efficient model training with GPU acceleration. This configuration enabled fast processing and effective training of the network intrusion detection model.

4.2 Dataset

For the evaluation of our proposed intrusion detection model, we selected two widely-used and well-documented datasets: CIC-DDoS2019 [44] and CIC-IDS2017 [45]. These datasets are specifically designed for network intrusion detection and include both normal and attack traffic, making them ideal for assessing the model's ability to detect various types of intrusions under realistic network conditions.

CIC-DDoS2019: This dataset includes traffic data from various types of DDoS (distributed denial of service) attacks, as well as normal network traffic. The CIC-DDoS2019 dataset includes a variety of attack types, such as HTTP flooding, DNS flooding, and other volumetric attacks, allowing for comprehensive testing of detection systems in the face of large-scale and high-intensity DDoS attacks. The dataset contains both training and testing sets, with network traffic captured at different times of the day to reflect various real-world conditions.

CIC-IDS2017: Another dataset from the Canadian Institute for Cybersecurity, the CIC-IDS2017 dataset includes a broader range of attacks, such as DoS (denial of service), R2L (remote to local), U2R (user to root), and probing attacks. This dataset provides labeled data for both training and testing purposes, with traffic captured from a realistic network environment. It is designed to help evaluate models on a diverse set of attack types, enabling the study of the model's generalization capabilities across different scenarios and attack vectors.

Both datasets are highly valuable for testing and benchmarking intrusion detection models, particularly in scenarios where there is a need to detect a range of known and unknown attack patterns. By using these datasets, we aim to evaluate the effectiveness, accuracy, and adaptability of our proposed model under different network security scenarios.

4.3 Data Analysis and Preprocessing

In this study, we selected two datasets: CIC-DDoS2019 and NSL-KDD. To meet the requirements of few-shot learning, we applied data pre-sampling, sample balancing, and matrix conversion to ensure the datasets are suitable for training in a meta-learning model.

Initially, we performed random and stratified sampling for each dataset, limiting the number of samples for each attack category to 5000. The benign samples were stratified and sampled proportionally. For CIC-DDoS2019, we kept major attack types, such as WebDDoS and DNSDoS, while for NSL-KDD, we sampled based on attack categories like DoS, Probe, and U2R. Some datasets exhibited class imbalance, particularly in the NSL-KDD dataset, where U2R attack samples were limited. To address this, we applied the SMOTE algorithm for oversampling the minority classes and used GMM undersampling for the majority classes. This balancing technique helped ensure that all attack categories were equally represented, reducing bias in model training.

To make the data suitable for deep learning models, we converted the text-based, numerical flow data into 2D matrices. Initially, we performed feature selection to remove irrelevant features (e.g., Flow ID, Source IP). For CIC-DDoS2019, we retained 78 features, and for NSL-KDD, we kept 38 features. Subsequently, each data instance was transformed into a 28×28 matrix, with feature values distributed evenly across the matrix using equal interval filling. This approach allowed the processed data to be input into the depthwise separable convolution model effectively. After preprocessing, the

resulting datasets provided a balanced and representative set of training data for model learning.

These preprocessing steps ensure high-quality, structured data for training, laying a strong foundation for the model's learning process and accurate predictions.

4.4 Experiment Design and Meta-training

In the meta-learning framework, the dataset is divided into meta-training, meta-validation, and meta-testing sets. The dataset contains 70 categories. We randomly selected 48 categories for meta-training, 14 for meta-validation, and 16 for meta-testing. The number of tasks in each set is determined by category combinations, resulting in approximately 1.7 million tasks for meta-training, 1120 tasks for meta-validation, and 2560 tasks for meta-testing. Each task involves a five-way classification with different shot settings: 1-shot, 5-shot, and 10-shot. In the five-way 1-shot setting, one sample per category is used for training, and five and ten samples per category are used, respectively. The test set for each task contains 12 samples.

The experiments run on a server with key hyperparameters, including the CNN gradient update step size (α), meta-learner update step size (β), the number of tasks for meta-updating, and gradient update iterations for the inner model. Training time is kept under 1 hour for most tasks, with longer training for higher shot settings. During training, the loss function is calculated every 100 iterations to monitor convergence. Each iteration uses five tasks, with a total of 32,000 tasks for meta-training.

4.5 Evaluation Metrics

In this study, we evaluated the model using accuracy based on the confusion matrix, as this is a standard metric for classification tasks. Cross-validation was applied during training to ensure the reliability and robustness of the results, particularly given the small sample sizes used in meta-learning.

5 Results

5.1 Performance Evaluation of Meta-learning and the Baseline Model

In Table 1, we present the performance comparison between our (meta-learning based depthwise separable convolution) and the baseline DSC model

Table 1 Comparison results with baseline models in few-shot intrusion detection

Experiment Setup	Model	Acc1 (%)	Acc5 (%)	Best Accuracy (%)
Five-way 1-shot	DSC	45.12	64.25	66.33
	Ours	74.89	80.37	81.27
Five-way 5-shot	DSC	47.28	72.19	79.11
	Ours	83.64	88.19	89.03
Five-way 10-shot	DSC	40.81	69.72	77.32
	Ours	86.53	91.47	92.62

across the three experimental setups. The results include the accuracy after one random gradient descent (Acc1), the accuracy after five gradient descents (Acc5), and the best accuracy observed after 10 iterations (best accuracy). In each experimental setup, five categories are randomly selected for training, and a small number of samples are used for learning. During testing, we calculate Acc1 (accuracy after one random gradient descent), Acc5 (accuracy after five random gradient descents), and best accuracy (best accuracy after 10 iterations).

In the five-way 1-shot setting, the ours model outperforms the DSC model with Acc1 improving from 45.12% to 74.89%, and shows a significant advantage in subsequent updates, with Acc5 reaching 80.37% and the best accuracy at 81.27%. In all three setups, ours continues to show superior performance, particularly in Acc1 and best accuracy, which reached 86.53% and 92.62%, respectively. These results indicate that combining the meta-learning framework with depthwise separable convolutions (ours) significantly enhances the performance of the few-shot intrusion detection model, especially in scenarios with limited data. The DSC model, while still offering strong performance, particularly in terms of computational efficiency and training time, falls short when it comes to handling new or unseen attack types in the few-shot learning scenario. Ours demonstrates a clear advantage in Acc1 and best accuracy, suggesting that the meta-learning framework enhances the model's ability to generalize and adapt to new, unseen attacks, especially when limited data is available for training. However, DSC does have an advantage in environments where training speed and resource consumption are the main concerns. In terms of model stability, while ours shows impressive performance improvements in terms of Acc5 and best accuracy, the model does experience fluctuations in its initial performance due to the nature of meta-learning. Specifically, the five-way 1-shot setup shows a noticeable improvement in Acc1 but may also experience higher false alarm rates compared to DSC. This is a common

challenge in few-shot learning models, where overfitting to small datasets can lead to initial instability. Nonetheless, with further iterations, ours shows a robust and stable performance, particularly in Acc5 and best accuracy. This demonstrates that while the meta-learning framework introduces more complexity, it significantly contributes to improving the overall performance of the model, especially in detecting new attack types. Compared to the traditional DSC model, ours exhibits a stronger adaptability and robustness when facing new attack patterns in few-shot learning scenarios. However, the trade-off between accuracy and computational cost must be considered when applying the model in real-time environments. Further improvements could be made to optimize training time and resource consumption without compromising performance.

5.2 Comparison with State-of-the-art Models

In this section, we compare our proposed Meta-DSC model with several state-of-the-art intrusion detection models, using the CIC-DDoS2019 and CIC-IDS2017 datasets for evaluation.

Table 2 shows that our model consistently achieves the highest detection accuracy across both the CIC-DDoS2019 and CIC-IDS2017 datasets, outperforming other state-of-the-art models. Our model achieved an accuracy of 92.62% with a sample size of 1200 and 91.45% with 1500 samples on the CIC-DDoS2019 dataset. These results significantly surpass those of other models, such as IE-DBN (88.20%) and CNN-LSTM (85.50%). Furthermore, on the CIC-IDS2017 dataset, our model also delivered impressive results, with an accuracy of 91.20% and 90.80% for sample sizes of 1200 and 1500, respectively. In comparison, models such as MASiNet and ResTranBiLSTM achieved lower accuracy, 89.10% and 90.50%, respectively. These results highlight the robustness of our model in handling different types of intrusion detection tasks, particularly under low-resource conditions with relatively small datasets. Despite the smaller sample sizes, ours excels in classifying a larger number of categories (up to 12 on CIC-DDoS2019), demonstrating its effectiveness for real-time intrusion detection. The high performance, combined with the model's ability to quickly adapt to new tasks through meta-learning, makes our model a highly efficient solution for modern intrusion detection systems, especially in resource-constrained environments where computational efficiency is critical. While the Meta-DSC model outperforms other models in terms of accuracy, it is important to note the trade-offs regarding computational cost and training time. While

Table 2 Comparison of detection accuracy between meta-DSC and state-of-the-art models

Model	Dataset	Categories	Classification Setup	Feature Type	Sample Size	Accuracy
IE-DBN [46]	CIC-DDoS2019	10	Five-way	Statistic + DBN	2000	88.20%
CNN-LSTM [47]	CIC-DDoS2019	10	Five-way	CNN + LSTM	2500	85.50%
DBN-ELM [48]	CIC-DDoS2019	8	Five-way	DBN + ELM	1800	87.30%
MASiNet [49]	CIC-IDS2017	8	Five-way	CNN + BiLSTM	3000	89.10%
Res-TranBiLSTM [50]	CIC-IDS2017	8	Five-way	ResNet + BiLSTM	3200	90.50%
Ours	CIC-DDoS2019	12	Five-way	Depthwise separable conv	1200	92.62%
Ours	CIC-DDoS2019	10	Five-way	Depthwise separable conv	1500	91.45%
Ours	CIC-IDS2017	10	Five-way	Depthwise separable conv	1200	91.20%
Ours	CIC-IDS2017	10	Five-way	Depthwise separable conv	1500	90.80%

Table 3 Comparison of model performance across CIC-DDoS2019 and CIC-IDS2017 datasets

	CIC-DDoS2019				CIC-IDS2017			
Model	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)	Parameters (M)	Flops (G)	Inference Time (ms)	Training Time (s)
IE-DBN	10.5	30.2	28.4	500	10.5	30.2	27.8	520
CNN-LSTM	14.7	55.3	34.6	820	14.7	55.3	33.9	840
DBN-ELM	8.3	22.1	26.1	450	8.3	22.1	25.5	470
MASiNet	20.2	65	40.2	1000	20.2	65	38.8	1030
Res-TranBiLSTM	25.3	85.7	43.5	1250	25.3	85.7	42.1	1280
Ours	7.1	19.8	22.5	360	7.1	19.8	21.9	380

the meta-learning approach provides a significant performance boost, it also introduces higher computational demands compared to simpler models like IE-DBN or CNN-LSTM, which are less complex and faster to train. The superior accuracy of ours comes with the cost of increased training time, making it less suitable in scenarios where real-time detection is crucial and resources are limited. However, for many practical applications, especially in environments where accuracy is prioritized over speed, our model's robustness and ability to generalize across tasks are significant advantages. Despite these challenges, ours remains a highly effective solution for intrusion detection, particularly in scenarios where datasets are small or involve new, unseen attack types. The combination of depthwise separable convolutions with the meta-learning framework enables the model to rapidly adapt to new attack patterns with minimal data, providing a valuable tool for real-time intrusion detection.

Table 3 compares the performance of various intrusion detection models on the CIC-DDoS2019 and CIC-IDS2017 datasets. Our model demonstrates a clear advantage in terms of computational efficiency. It has the fewest parameters and lowest flops compared to other models like Res-TranBiLSTM and MASiNet, making it highly efficient in terms of computational resources. Additionally, our model achieves faster inference times, taking only 22.5 ms for CIC-DDoS2019 and 21.9 ms for CIC-IDS2017, which is significantly lower than the inference times of more complex models such as MASiNet (40.2 ms) and Res-TranBiLSTM (43.5 ms). Furthermore, the training time for our model is also notably shorter, requiring just 360 seconds for CIC-DDoS2019 and 380 seconds for CIC-IDS2017, much less than models like Res-TranBiLSTM (1250 seconds) and MASiNet (1000 seconds). These results show that our model offers a significant trade-off between performance and computational efficiency. Our model achieves impressive detection accuracy while using far fewer computational resources compared

to more complex models. The lower parameters and flops contribute to the overall faster inference times and shorter training times, making it more suitable for deployment in real-time intrusion detection systems, especially in environments with constrained computational resources. While our model achieves superior results in terms of computational efficiency, it is important to note that it also delivers comparable or better performance in terms of detection accuracy, despite the smaller number of parameters. This highlights the strength of the depthwise separable convolution approach used in our model, which helps to balance both accuracy and computational efficiency. In contrast, models like MASiNet and Res-TranBiLSTM, while achieving higher accuracy, come with significantly larger models and require more computational resources, both in terms of parameters and flops, resulting in slower inference times and longer training periods. These models are more complex and may be better suited for scenarios where higher computational resources are available but may not be the best choice in resource-constrained environments.

5.3 Loss Curves for Different Few-shot Learning Configurations

The loss curves for the three configurations are shown in Figure 3. Initially, all three configurations experience a rapid decline in loss, with the five-way 1-shot setting showing the steepest drop, as expected with fewer training samples. The curves for five-way 5-shot and five-way 10-shot show

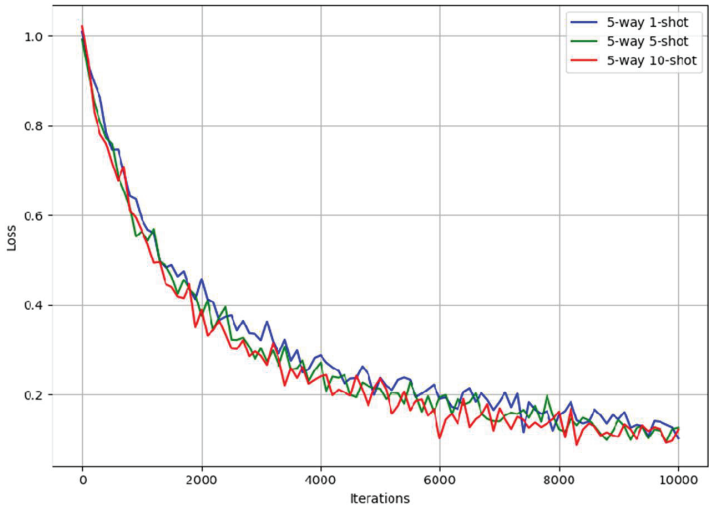

Figure 3 Loss curves.

a more gradual decrease in loss, reflecting the added stability provided by the larger sample sizes. This demonstrates that with the increase in samples, the model's training process becomes more stable, and the model is able to better generalize from the data. After approximately 4000 iterations, the loss curves for all three settings stabilize, indicating that the model has reached a point of convergence where the error no longer significantly decreases. This stabilization reflects the model's ability to learn effectively over time. However, the five-way 1-shot configuration experiences some fluctuation throughout the training process due to the challenges associated with few-shot learning. This result aligns with the fact that, with fewer samples, the model struggles to generalize and tends to overfit to the limited data, leading to these fluctuations. On the other hand, the five-way 5-shot and five-way 10-shot configurations exhibit smoother and more stable curves, suggesting that with more training samples, the model is able to achieve more consistent performance. This suggests that increasing the number of samples improves the model's ability to converge and adapt to the task. The five-way 10-shot configuration shows the least fluctuation, indicating the most stable learning process and optimal performance, with the model's loss stabilizing at a lower value compared to the other configurations. These results highlight the importance of sample size in improving model stability, particularly in few-shot learning settings where data scarcity can introduce significant instability in the learning process. Furthermore, while the five-way 1-shot configuration demonstrates a faster decline in loss early on, it eventually faces challenges in reaching a lower loss compared to the other configurations. This may also be an indicator of higher false alarm rates or detection errors in this configuration, which is common in few-shot settings. The five-way 10-shot configuration, however, demonstrates superior stability and lower loss values, which likely correlates with improved detection rates and lower false alarm rates, as the model is able to generalize better and learn from more data. Overall, the loss curves demonstrate the model's ability to efficiently learn and stabilize even with varying sample sizes. The training process is effective in all three settings once sufficient iterations are completed, but larger sample sizes contribute to more stable convergence and likely better overall performance in terms of detection accuracy and model robustness.

6 Conclusions

In this study, we proposed a meta-learning based intrusion detection model, Meta-DSC, which integrates depthwise separable convolution (DSC) with

meta-learning techniques. We conducted experiments on two widely-used datasets, CIC-DDoS2019 and CIC-IDS2017, to evaluate the model's effectiveness in few-shot learning scenarios. The results demonstrated that our model achieved superior accuracy in detecting various types of network intrusions, even when trained with a limited number of samples. Compared to existing methods, Meta-DSC not only outperformed in terms of classification accuracy but also showed significant improvements in training efficiency and computational resource consumption, making it a promising approach for real-world intrusion detection tasks.

Despite the promising results, there are some limitations in our model. First, while the model performed well on the CIC-DDoS2019 and CIC-IDS2017 datasets, its generalization ability to other types of network attack datasets remains untested. To enhance the robustness of the model, future work will explore its performance on more diverse datasets, including those from other domains, to assess its generalization ability across a broader range of attack types. Additionally, while the model's training time is efficient, it faces challenges with very large-scale datasets. Future research will focus on optimizing the model's architecture and leveraging hardware acceleration to improve its scalability in large-scale environments.

This work introduces a novel intrusion detection approach by integrating meta-learning with few-shot learning techniques. Further research will investigate expanding the model's applicability by testing it across more varied attack scenarios and network environments, with a focus on improving interpretability for practical deployment in complex settings.

Funding

This work was supported by the Science and Technology Project of Henan (Optical image compression and encryption based on depth learning and high-quality reconstruction, 242102210058).

References

[1] X. Fang, M. Xu, S. Xu, and P. Zhao, "A deep learning framework for predicting cyber attacks rates," *EURASIP Journal on Information security*, vol. 2019, pp. 1–11, 2019.

[2] J. Zhang, L. Pan, Q.-L. Han, C. Chen, S. Wen, and Y. Xiang, "Deep learning based attack detection for cyber-physical system cybersecurity:

A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 3, pp. 377–391, 2021.

[3] L. Ashiku and C. Dagli, "Network intrusion detection system using deep learning," *Procedia Computer Science*, vol. 185, pp. 239–247, 2021.

[4] D. Chou and M. Jiang, "A survey on data-driven network intrusion detection," *ACM Computing Surveys (CSUR)*, vol. 54, no. 9, pp. 1–36, 2021.

[5] Z. Yang et al., "A systematic literature review of methods and datasets for anomaly-based network intrusion detection," *Computers & Security*, vol. 116, p. 102675, 2022.

[6] L. Zhang, J. Liu, Y. Wei, D. An, and X. Ning, "Self-supervised learning-based multi-source spectral fusion for fruit quality evaluation: A case study in mango fruit ripeness prediction," *Information Fusion*, vol. 117, p. 102814, 2025.

[7] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5149–5169, 2021.

[8] C. Lu, X. Wang, A. Yang, Y. Liu, and Z. Dong, "A Few-Shot-Based Model-Agnostic Meta-Learning for Intrusion Detection in Security of Internet of Things," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21309–21321, 2023.

[9] C. Xu, J. Shen, and X. Du, "A method of few-shot network intrusion detection based on meta-learning framework," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3540–3552, 2020.

[10] F. Rustam, A. Raza, M. Qasim, S. K. Posa, and A. D. Jurcut, "A novel approach for real-time server-based attack detection using meta-learning," *IEEE Access*, vol. 12, pp. 39614–39627, 2024.

[11] A. Sohail, B. Ayisha, I. Hameed, M. M. Zafar, H. Alquhayz, and A. Khan, "Deep neural networks based meta-learning for network intrusion detection," *arXiv preprint arXiv:2302.09394*, 2023.

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

[13] J. Huang, X. Yu, D. An, X. Ning, J. Liu, and P. Tiwari, "Uniformity and deformation: A benchmark for multi-fish real-time tracking in the farming," *Expert Systems with Applications*, vol. 264, p. 125653, 2025.

[14] Y. Fu, Y. Du, Z. Cao, Q. Li, and W. Xiang, "A deep learning model for network intrusion detection with imbalanced data," *Electronics*, vol. 11, no. 6, p. 898, 2022.

[15] D. Li, L. Deng, M. Lee, and H. Wang, "IoT data feature extraction and intrusion detection system for smart cities based on deep migration learning," *International journal of information management*, vol. 49, pp. 533–545, 2019.

[16] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *Ieee Access*, vol. 7, pp. 64366–64374, 2019.

[17] H. Xu and Y. Wang, "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)*, 2022: IEEE, pp. 1188–1194, doi: 10.1109/ICCASIT55263.2022.9986665.

[18] X. Huang, S. Zhu, and Y. Ren, "A Semantic Matching Method of E-Government Information Resources Knowledge Fusion Service Driven by User Decisions," *Journal of Organizational & End User Computing*, vol. 35, no. 1, 2023.

[19] O. H. Abdulganiyu, T. Ait Tchakoucht, and Y. K. Saheed, "A systematic literature review for network intrusion detection system (IDS)," *International journal of information security*, vol. 22, no. 5, pp. 1125–1162, 2023.

[20] Y. S. Almutairi, B. Alhazmi, and A. A. Munshi, "Network intrusion detection using machine learning techniques," *Advances in Science and Technology Research Journal*, vol. 16, no. 3, pp. 193–206, 2022.

[21] M. Li and W. Xiao, "Research on the Effect of E-Leadership on Employee Innovation Behavior in the Context of "Self" and "Relationship"," *Journal of Organizational & End User Computing*, vol. 35, no. 1, 2023.

[22] H. Choi, M. Kim, G. Lee, and W. Kim, "Unsupervised learning approach for network intrusion detection system using autoencoders," *The Journal of Supercomputing*, vol. 75, pp. 5597–5621, 2019.

[23] A. Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, "A survey of neural networks usage for intrusion detection systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 497–514, 2021.

[24] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[25] M. J. Idrissi et al., "Fed-anids: Federated learning for anomaly-based network intrusion detection systems," *Expert Systems with Applications*, vol. 234, p. 121000, 2023.

[26] L. A. Nguyen, I. Miciæ, N.-T. Nguyen, and S. Stanimiroviæ, "Depth-Bounded Fuzzy Bisimulation for Fuzzy Modal Logic," *Cybernetics and Systems*, pp. 1–18, 2023.

[27] T. Rupa Devi and S. Badugu, "A review on network intrusion detection system using machine learning," in *International Conference on E-Business and Telecommunications*, 2019: Springer, pp. 598–607.

[28] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *Ict Express*, vol. 4, no. 2, pp. 95–99, 2018.

[29] E. Suwannalai and C. Polprasert, "Network intrusion detection systems using adversarial reinforcement learning with deep Q-network," in *2020 18th International Conference on ICT and Knowledge Engineering (ICT&KE)*, 2020: IEEE, pp. 1–7.

[30] A. Aruna Kumari, A. Bhagat, and S. Kumar Henge, "Classification of Diabetic Retinopathy Severity Using Deep Learning Techniques on Retinal Images," *Cybernetics and Systems*, pp. 1–25, 2024.

[31] K. Fan, W. Zhang, G. Liu, and H. He, "FMSA: a meta-learning framework-based fast model stealing attack technique against intelligent network intrusion detection systems," *Cybersecurity*, vol. 6, no. 1, p. 35, 2023.

[32] M. Kim, "ML/CGAN: Network attack analysis using CGAN as meta-learning," *IEEE Communications Letters*, vol. 25, no. 2, pp. 499–502, 2020.

[33] M. Sannidhan, J. E. Martis, R. S. Nayak, S. K. Aithal, and K. Sudeepa, "Detection of antibiotic constituent in Aspergillus flavus using quantum convolutional neural network," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 14, no. 1, pp. 1–26, 2023.

[34] F. Liu, M. Li, X. Liu, T. Xue, J. Ren, and C. Zhang, "A review of federated meta-learning and its application in cyberspace security," *Electronics*, vol. 12, no. 15, p. 3295, 2023.

[35] M. Rafiei, M. Maheri, and H. R. Rabiee, "Privacy Challenges in Meta-Learning: An Investigation on Model-Agnostic Meta-Learning," *arXiv preprint arXiv:2406.00249*, 2024.

[36] Z. Wang, M. Li, H. Ou, S. Pang, and Z. Yue, "A Few-Shot Malicious Encrypted Traffic Detection Approach Based on Model-Agnostic

Meta-Learning," *Security and Communication Networks*, vol. 2023, no. 1, p. 3629831, 2023.

[37] A. Kodipalli, S. L. Fernandes, S. K. Dasar, and T. Ismail, "Computational framework of inverted fuzzy C-means and quantum convolutional neural network towards accurate detection of ovarian tumors," *International Journal of E-Health and Medical Communications (IJEHMC)*, vol. 14, no. 1, pp. 1–16, 2023.

[38] Y. Zhou et al., "Optimization of automated garbage recognition model based on resnet-50 and weakly supervised cnn for sustainable urban development," *Alexandria Engineering Journal*, vol. 108, pp. 415–427, 2024.

[39] Z. Wan, "A Meta-Learning based IDS," Purdue University Graduate School, 2024.

[40] A. Yang et al., "Application of meta-learning in cyberspace security: A survey," *Digital Communications and Networks*, vol. 9, no. 1, pp. 67–78, 2023.

[41] J. Zhao, Q. Li, Y. Hong, and M. Shen, "MetaRockETC: Adaptive Encrypted Traffic Classification in Complex Network Environments via Time Series Analysis and Meta-Learning," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 2460–2476, 2024.

[42] U. Zukaib, X. Cui, C. Zheng, D. Liang, and S. U. Din, "Meta-Fed IDS: Meta-Learning and Federated Learning Based Fog-Cloud Approach to Detect Known and Zero-Day Cyber Attacks in IoMT Networks," *Journal of Parallel and Distributed Computing*, vol. 192, p. 104934, 2024.

[43] S. Wang, R. Jiang, Z. Wang, and Y. Zhou, "Deep learning-based anomaly detection and log analysis for computer networks," *arXiv preprint arXiv:2407.05639*, 2024.

[44] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," in *2019 international carnahan conference on security technology (ICCST)*, 2019: IEEE, pp. 1–8, doi: 10.1109/CCST.2019.8888419.

[45] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, 2018.

[46] H. Jia, J. Liu, M. Zhang, X. He, and W. Sun, "Network intrusion detection based on IE-DBN model," *Computer Communications*, vol. 178, pp. 131–140, 2021.

[47] A. Halbouni, T. S. Gunawan, M. H. Habaebi, M. Halbouni, M. Kartiwi, and R. Ahmad, "CNN-LSTM: hybrid deep neural network for network intrusion detection system," *IEEE Access*, vol. 10, pp. 99837–99849, 2022.

[48] D. Liang and P. Pan, "Research on intrusion detection based on improved DBN-ELM," in *2019 international conference on communications, information system and computer engineering (CISCE)*, 2019: IEEE, pp. 495–499, doi: 10.1109/CISCE.2019.00115.

[49] Y. Wu et al., "MASiNet: Network Intrusion Detection for IoT Security Based on Meta-Learning Framework," *IEEE Internet of Things Journal*, vol. 11, pp. 25136–25146, 2024, doi: 10.1109/JIOT.2024.3395629.

[50] S. Wang, W. Xu, and Y. Liu, "Res-TranBiLSTM: An intelligent approach for intrusion detection in the Internet of Things," *Computer Networks*, vol. 235, p. 109982, 2023.

Biographies

Guo Li was born in Henan, China, in 1980. From 1999 to 2009, he studied at Airforce Engineering University and received his bachelor's degree in 2003. He received his Master's degree in 2006 and his Doctor's degree in 2009. Currently, he works in Nanyang Normal University. He has published ten papers, five of which has been indexed by SCI and EI. His research interests are included intelligent information processing and loT.

MingHua Wang was born in Anhui, China, in 1974. From 1992 to 1996, he studied at Air Force Telecommunications Engineering College and received his bachelor's degree in 1996. From 2005 to 2007, he studied in ShanDong University and received his Master's degree in 2007. Currently, he works in Shandong Gete Aviation Technology Co., Ltd. His research interests include flight information processing and IoT.