PTN-IDS: Prototypical Network Solution for the Few-shot Detection in Intrusion Detection Systems

Nadia Niknami, Vahid Mahzoon, and Jie Wu
Center for Networked Computing, Temple University, USA

***Abstract*—Local Area Networks (LANs), as interconnected networks, are susceptible to numerous security threats. Existing intrusion detection systems (IDS) heavily rely on large, fully-labeled datasets to have accurate detection, facing challenges when only a few malicious samples are available. In addition, previous studies have identified the deterioration of IDS's performance when the test dataset deviates from the training dataset distribution. To mitigate these issues, we propose a Prototypical Network-based IDS within a meta-learning framework. Our method adopts a Few-Shot Learning (FSL) approach, aiming to distinguish and compare network traffic samples to classify them as either normal or malicious. Notably, our model not only identifies benign or malicious traffic but also accurately identifies the specific types of attacks. We evaluate the effectiveness of our approach in different scenarios for few-shot network intrusion detection using real-world network traffic data. Additionally, we conduct a comprehensive sensitivity analysis to assess the impact of key factors such as model hyperparameters, support set size, attack type distribution, and distance metrics within the prototypical network model.**

***Index Terms*—Few-shot learning, Intrusion Detection System(IDS), Meta-learning, Multi-class classification, Prototypical Network(PTN).**

I. Introduction

Network intrusion detection (IDS) [1] plays a pivotal role in ensuring network security, particularly in the context of local area networks (LANs), which are interconnected networks vulnerable to various security threats [2]. In recent years, the application of deep learning (DL) [3]–[5] technology in intrusion detection has attracted considerable attention from researchers. Leveraging DL techniques, IDSs aim to classify network traffic into normal and attack categories, thereby fortifying network defenses against potential threats. Numerous studies have demonstrated the efficacy of DL-based IDS, showcasing their stability and high detection rates [6] [7]. However, despite the advancements in DL-based IDS, several challenges persist with existing approaches, especially within LAN environments. Issues such as imbalanced training data, elevated false alarm rates, and the inability to detect unknown attacks remain unresolved. The detection of malicious attack traffic within LANs is imperative for network security, especially in light of emerging threats such as Zero-day attacks [8]. These attacks exploit vulnerabilities on the same day they are discovered, presenting a significant challenge for traditional intrusion detection systems reliant on supervised machine learning approaches. Fig. 1 illustrates a LAN with a structured defense strategy against cybersecurity threats. The red-highlighted computers represent compromised systems or attackers within the network, indicating a security breach in sections 2 and 5, as well as among some consumers and in Branch 2. These threats are actively being monitored by the green-highlighted components, dedicated to traffic monitoring and an IDS. The IDSs deployed within this network are effective at identifying known attacks based on pre-existing signatures. However, they face significant challenges when it comes to detecting unseen or Zero-day attacks.

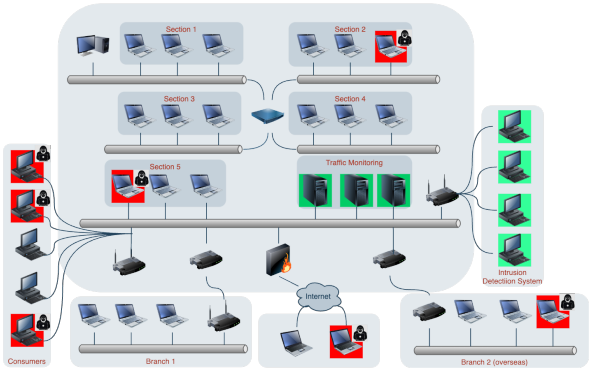


Fig. 1: Zero-day Attacks in Local Area Networks.

Changes in network topology or traffic scale can significantly impact the performance of an IDS in various ways. Firstly, alterations in these factors can lead to an increase in both false positives (misclassifying normal traffic as malicious) and false negatives (failing to detect actual intrusions). This discrepancy occurs because the IDS may struggle to adapt its detection mechanisms to the new environment. Moreover, substantial changes can overwhelm the computational resources of the IDS, resulting in performance degradation or even system failures. Adapting IDS to the new conditions may pose challenges, necessitating recalibration or retraining, which is often time-consuming and resource-intensive. Therefore, addressing changes in network topology or traffic scale is crucial to maintain the effectiveness of IDS.

In our analysis, we consider two key challenges for IDSs. The first challenge is *Domain Shift*, which refers to differences between the distribution of data in the training domain and the deployment or test domain. This disparity can cause the model to struggle with generalization, leading to a decline in IDS performance. *Attack Diversity* presents another challenge for IDSs, where models trained on one set of attacks may not generalize well to novel attack types or variations encountered in the new environment. Designing distinct IDS for every

network type and potential attack scenario remains impractical. In scenarios with a scarcity of labeled traffic samples, IDS' effectiveness diminishes, highlighting the significance of addressing the few-shot intrusion detection problem within LAN environments. Studies have highlighted the potential of Few-Shot Learning (FSL) [9] in mitigating the drawbacks of DL, notably by reducing the time and resources needed for dataset collection and labeling [10]. FSL emerges as a promising approach, seeking to enhance classification performance using minimal labeled data. Similar to human learning processes, FSL leverages prior knowledge to adapt to new tasks, requiring only limited data for task acquisition. Few-shot classification tasks train a classifier on novel classes not present in the training data, utilizing only a few examples for each new class.

In this paper, we propose a Prototypical Network (PTN) classification approach to effectively categorize network traffic into various classes and detect novel attacks. The model is trained on labeled traffic data and is evaluated for its ability to identify new attack types when only limited labeled samples are available. Our primary focus is developing a few-shot-based classification algorithm suitable for practical scenarios with limited and potentially erroneous data. Addressing the challenge of learning from a small training dataset is crucial, and our approach emphasizes extracting meaningful information to enhance classification performance. Additionally, we aim to maximize data utilization to enable efficient learning with minimal labeled samples. Leveraging a meta-learning framework, our neural network learns to differentiate between samples during training, even when encountering tasks not encountered before. This paper makes the key contributions to addressing the existing issues above as follows:

- We propose a PTN-based IDS capable of detecting unseen attacks using only a few labeled samples with acceptable accuracy. Our model not only distinguishes between benign and malicious traffic but also accurately identifies specific types of attacks.
- We address the challenges of Zero-day attacks and domain shift in intrusion detection by considering two different datasets containing a wide range of attacks, demonstrating the performance of our approach.
- Our proposed network intrusion detection method is universal and not limited to specific attack types. It leverages learned prior knowledge to detect new types of samples based on a limited number of labels in a target dataset.
- We conduct a comprehensive sensitivity analysis of the PTN model, examining hyperparameters, support set size, attack type distribution, and distance metrics to understand the model's performance and robustness in real-world scenarios.

II. Background and related works

A. Few-shot Learning

Few-shot learning presents a unique challenge in machine learning, where the availability of labeled samples for training is severely limited, often leading to overfitting. To address this, few-shot learning (FSL) divides the dataset into meta-training and meta-testing sets, employing a training procedure consisting of multiple episodes. In practical scenarios, the model may be tasked with classifying instances from N different classes, each with only K examples, termed an N-way K-shot task. During meta-training, two batches of N-way K-shot data are sampled in each episode, forming a support set $\mathcal{S}$ and a query set $\mathcal{Q}$. The model is then trained using the support set and evaluated using the query set. Therefore, each episode encompasses its own training and testing sets, allowing the model to learn to solve specific tasks with only $N \times K$ samples [11]–[13].

B. Prototypical Networks (PTN)

PTN [14] learns a metric space where classification is conducted by computing distances to prototype representations of each class. Compared to recent approaches for FSL, PTNs exhibit a simpler inductive bias that proves beneficial in the limited-data regime, yielding excellent results. The PTN framework is grounded in the notion that an embedding exists wherein points cluster around a single prototype representation for each class. To achieve this, we employ a neural network to learn a non-linear mapping of the input into an embedding space. The prototype for each class is then determined as the mean of its support set in the embedding space. Classification is straightforward: for an embedded query point, the nearest class prototype is identified. Each class is accompanied by meta-data providing a high-level description of the class, rather than a small number of labeled examples. The Prototypical Network demonstrates remarkable performance across diverse applications, including image classification [14], text classification [15], and radio frequency fingerprinting [16].

C. Few-shot Learning in Intrusion Detection

Detection based on only a limited number of attack samples is considered to be a few-shot intrusion detection problem, for which some researchers have begun to apply meta-learning ideas and related algorithms to network intrusion detection and its data analysis [17]–[20]. Xu *et al.* [21] were the first to apply meta-learning to network intrusion detection. They took raw traffic bytes as input and achieved few-shot traffic classification by training on meta-tasks. Meta-learning models can quickly adjust through a small amount of new data to adapt to the new situation. It has advantages in quick learning and adaptation to new tasks and scenarios, such as strong generalization ability, low resource overhead, and easy scene transfer. Yang *et al.* [22] proposed an improved traffic classification model, FS-IDS, which improved the performance of model in few-shot classification by integrating raw traffic and traffic statistical features. However, FS-IDS is only applicable for detecting specific malicious samples and is unable to detect unknown attacks. Lu *et al.* [23] propose FSL solutions based on Model-Agnostic Meta-Learning (MAML) to detect anomalous traffic with only few samples. MAML is a popular FSL approach that continuously updates the model parameters

through a meta-learning process and a fine-tuning phase to quickly adapt to new FSL task.

Through our investigation of few-shot intrusion detection models, we have observed that most researchers concentrate on model-based meta-learning. Their proposed models rely on network data using a complex deep neural network during training. Also, they investigate the performance on a single dataset while overlooking the domain shift problem. In this paper, we introduce a Prototypical Network-based IDS capable of identifying unseen attacks with only a few labeled samples, achieving acceptable accuracy. Our model not only distinguishes between benign and malicious traffic, but also accurately identifies specific attack types. By addressing the challenges of Zero-day attacks and domain shift, we evaluate our approach on two diverse datasets containing a wide array of attacks, showcasing its performance.

III. PTN-IDS: Prototypical Network-based IDS

IDS can be reliable in detecting various types of attacks if sufficient data are available from all the attack types. To verify this, we start by conducting a multi-class classification on CICIDS2017 dataset [24] using a Feed Forward Neural Network and we were able to get 0.97 as accuracy on test dataset. This result shows that if we have sufficient labeled data from each attack and benign case, we can get a very high accuracy. By extracting a 25-dimensional embedding from the neural network's last hidden layer and applying t-SNE [25], we obtained a 2D representation illustrated in Fig. 2 that visually confirms the effectiveness of our model in clustering and therefore a high accuracy. Also, it can be observed that there is considerable overlap between various traffic classes.

However, in this paper, we suppose that sufficient labeled data from each attack is not available and we aim to tackle such a challenging problem. To do so, we propose a Prototypical Network-based IDS within a meta-learning framework. Our method adopts a FSL approach, aiming to distinguish and compare network traffic samples to classify them as either normal or malicious. Furthermore, we aim to investigate whether the classifier is able to transfer the learned patterns when evaluated on a new dataset originating from a different network than the one used for training. Consider datasets CICIDS2017 and CICIDS2018 [24], the number of attack types, the number of flows belonging to each attack type, and the ratio of the number of flows in each attack type to total flows are different in each dataset. Contrary to grouped binary classification training, where all attack classes are grouped under a single label, we performed single-attack experiments, utilizing samples pertaining to a singular class of attack, alongside samples representing benign traffic. We aim to examine the transferability of knowledge between different attacks for generalization across datasets, considering the nature and modalities of specific attacks, as well as the correlation between attack type and generalization.

We approach this problem as a binary classification task, wherein each type of attack is labeled as *Malicious*, while

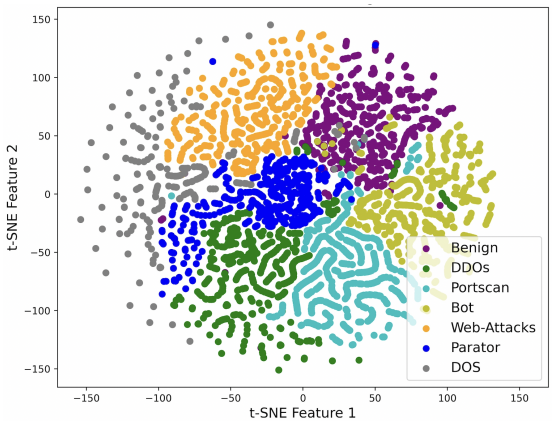


Fig. 2: Multi-class classification embedding.

the remaining traffic is labeled as *Benign*. To ensure balanced training, we adjust both the CICIDS2017 and CICIDS2018 datasets to have an equal number of instances for each label. We treat each label as a distinct subset. Our methodology involves training a feed-forward neural network on the vertical axis (source task) for binary classification and evaluating its performance on the horizontal axis (target task) under similar conditions. In Figs. 3 and 4, the results demonstrate nearly perfect classification performance when the models are trained and tested on the same task. However, in a cross-dataset scenario, where models are trained and tested on different datasets, the classification accuracy is largely equivalent to random chance, except for a few combinations of attacks and datasets. In these figures, "All" signifies training on all datasets, while "leave-one-out" indicates training on all subsets except the one being tested. Additionally, we explore an alternative scenario where the model is trained on all subsets from the CICIDS2017 dataset and then tested on the CICIDS2018 dataset, and vice versa. Furthermore, in most cases of cross-dataset testing, the models show poor generalizability due to significant domain shifts. Therefore, we identify two main issues in the results of this experiment as *Detecting Zero-day Attack* and *Domain Shift*.

A. Problem Definition

In this section, we provide more details. We define the first problem as *Detecting Zero-day Attack* as follows:

Problem 1 (Detecting Zero-day Attack): A Zero-day attack, in the context of training and testing neural networks, is a cyberattack exploiting a vulnerability for which no prior training data exists to train the model on it.

Few-shot learning is a promising field in DL, which aims to train a model only on a small number of labeled training data. It sounds fascinating in intrusion detection, since it solves the problem of Zero-day attack detection. From another angle, current approaches to anomaly detection assume similar feature distributions for training and test data sets.

The majority of the proposed ML-based IDSs are evaluated only on domain-specific datasets, i.e., the training and eval-

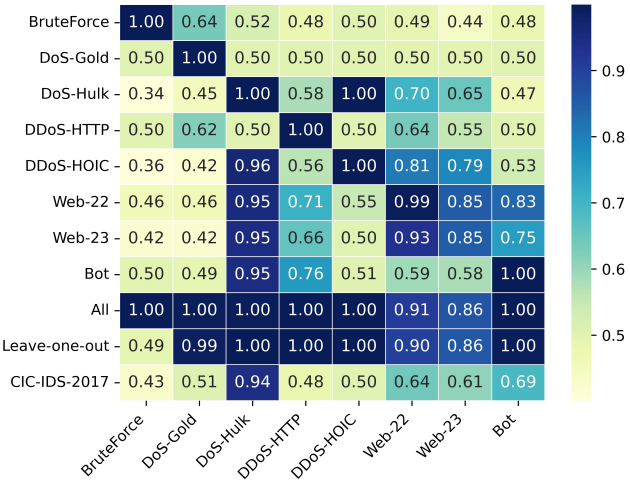


Fig. 3: The detection accuracy across different training and testing tasks within the CICIDS2018 dataset.

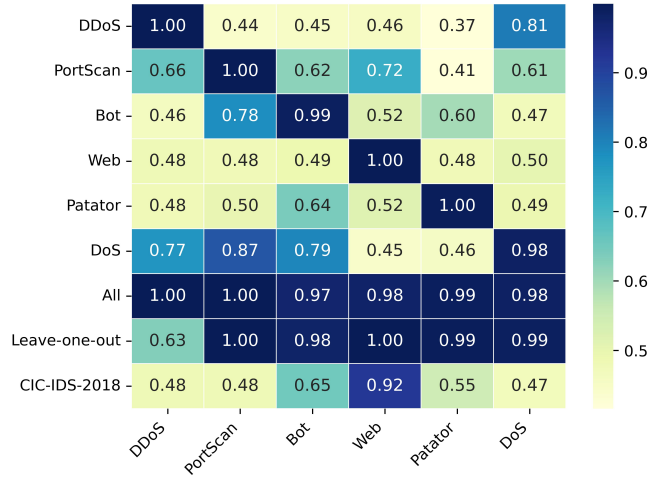


Fig. 4: The detection accuracy across different training and testing tasks within the CICIDS2017 dataset.

uation samples are drawn from the same dataset, and cross-domain evaluation is rarely considered. Therefore, these models fail to perform well when there is a distribution difference between the train (i.e., source) and test (i.e., target) data. We define the second problem as *Domain Shift* as follows:

Problem 2 (Domain Shift): Domain Shift in the context of training and testing neural networks refers to a scenario where the distribution of data in the testing phase significantly differs from that of the training phase.

Domain Shift refers to the differences between the distribution of data in the training domain and the deployment domain. As a result, the model may struggle to generalize to the new domain due to discrepancies in data characteristics, leading to a decline in performance. In Figs. 3 and 4, cross-dataset experiments show some generalization capabilities depending on the attack type and also on the specific train-test combination used for cross-dataset evaluation.

In Fig. 3, the last row illustrates the scenario where the source dataset is CICIDS2018, and the test dataset is CICIDS2017. Similarly, in Fig. 4, the last row depicts the scenario in which the source data set is CICIDS2017 and the test data set is CICIDS2018. These datasets originate from different networks with varying hardware and software environments, and they feature different types of attacks. The accuracy values highlight the challenge of domain shift, leading to a deterioration in the performance of IDS for most attack types. We investigate the extent to which detection becomes challenging in the presence of domain shift.

B. Methodology

In this section, we elaborate on the architecture of our proposed IDS framework based on FSL. We introduce our approach as "*PTN-IDS*" which is abbreviation of "*PTN-based Intrusion Detection System*". The methodology of this paper is shown in Fig. 5. A network traffic dataset will be divided into two distinct tasks: a source task and a target task. The division is structured so that there is no overlap in the label spaces (attack types) between these two tasks. The goal of the embedding function is to extract essential features, reduce the dimension of data, and retain all the information of the original data to the maximum extent. Various embedding models can be used to extract features. Here, we choose the Feed Forward Neural Network as our embedding function.

We use PTNs for the problem of few-shot classification, where a classifier must generalize to new classes not seen in the training set, given only a small number of examples of each new class. PTNs learn a metric space in which classification can be performed by computing distances to prototype representations of each class. In training our PTN-based IDS, we begin by constructing multiple tasks from our source task. This involves the creation of support sets and query sets for each task. Once the tasks are defined, we process the support set through the neural network. This step is crucial as it transforms the input data into embeddings, which are high-dimensional vectors representing the features of each input. These embeddings capture the essential characteristics of the data, enabling the model to learn more effectively. Next, we calculate prototypes for each label within the support set.

A prototype represents the mean vector of the embeddings corresponding to each class or label. Subsequently, the model is trained to compare the embeddings of the query set with these prototypes, with the aim of minimizing the distance between the query embeddings and the corresponding class prototype. In the Euclidean-based distance metric module, we utilize the Euclidean distance as the metric to measure the distance of each query point to the calculated prototypes from the support points. Specifically, the distance metric module calculates the Euclidean distance between the received query point and prototypes of all classes. By applying the softmax function, the module outputs a probability distribution of received query samples over different classes. The model then assigns the query sample to the class with the highest

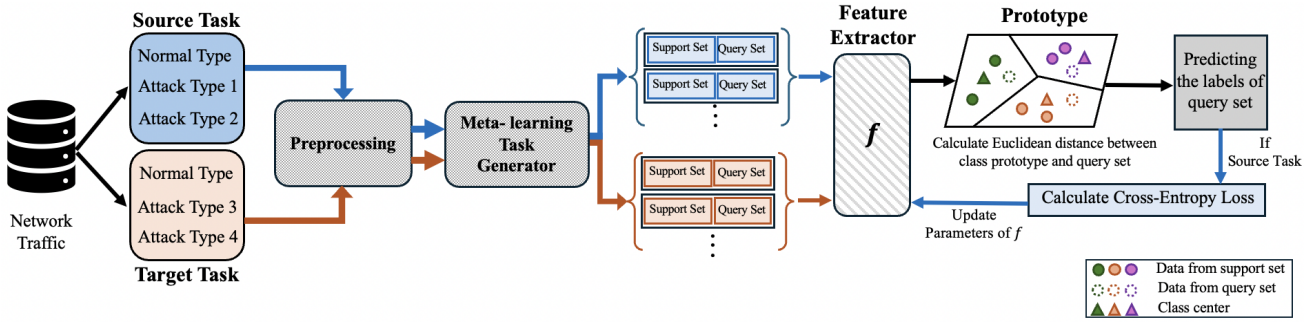


Fig. 5: Proposed PTN-based intrusion detection system.

Algorithm 1 Meta-learning Task Generator

Input: D: dataset, Label, N-way, K-shot, N-query
Output: Meta task set $M = \{M_j\}_{j=1}^{N\text{-tasks}}$, support set S_j, query set Q_j
for j=1 **to** $N - tasks$ **do**
  $L \leftarrow$ Randomly sample N-way labels from Label
  for each label in L **do**
    $S_j \leftarrow$ Randomly sample K-shot samples from D
    $Q_j \leftarrow$ Randomly sample N-query samples from D
  end for
  $M_j \leftarrow \{S_j, Q_j\}$
end for

probability, corresponding to the nearest prototype in the embedding space.

C. Few-Shot Training Strategy

PTN-IDS is a neural network designed for FSL, and therefore, its underlying training method is different from those employed in conventional DNNs. As described in Section II, the whole dataset is no longer simply divided into a training set and a testing one. Instead, a meta-training set containing multiple tasks needs to be generated so that each task includes a sample set and a query set for simulating the meta-testing set that comprises a support set and a test set. Once the task generator constructs a specific task, i.e., the support set and query set, the feature extraction network extracts their embedding features. The distance metric module calculates the prototype representation of each class according to the support point, and identifies categories of query points based on its predefined metric measure. Finally, on the basis of discriminant results, the loss function is computed and is optimized through back-propagation.

Algorithm 1 provides the pseudocode of *meta-learning task generator* in our approach. Given a dataset D along with label information $Label$ and parameters N-way, K-shot, and N-query, the algorithm generates a set of meta-tasks. Each meta-task consists of a support set S_j and a query set Q_j, where j iterates over the number of tasks to generate. First, PTN-IDS proceeds to generate N-tasks, each representing a meta-task. For each meta-task, a set of N-way labels is randomly sampled from the provided label information. Subsequently, for each label in the sampled label set L, the algorithm randomly selects K-shot instances from the dataset D to form the support set S_j. Additionally, N-query instances are randomly sampled from the dataset D to construct the query set Q_j. These support and query sets together constitute the meta-task M_j. This process continues until N-tasks are generated.

Algorithm 2 PTN-IDS

Input: Meta-learning task $M = \{M_j\}_{j=1}^{N-tasks}$, distance function $d(x_1, x_2)$, N-way, K-shot
Output: Trained PTN (f_W) with parameters W
Randomly initialize network parameters W
while Accuracy is improving **do**
  $M_k = (S_k, Q_k) \leftarrow$ Sample a task from M
  $f_W(S_k) \leftarrow$ Calculate embedding for S_k
  $f_W(Q_k) \leftarrow$ Calculate embedding for Q_k
  $\rho \leftarrow$ Initialize a vector of size N-way for prototypes
  for each class c in S_k **do**
    $\rho(c) \leftarrow \frac{1}{\text{K-shot}} \sum_{(X,c)\in S_k} f_W(X)$
  end for
  $L \leftarrow 0$ ▷ Initialize the loss for the episode
  $P \leftarrow$ Initialize a vector for $p(y_i|X)$
  for each (X, y) in Q_k **do**
    $P \leftarrow \text{softmax}\left(-d\left(f_W(X), \rho\right)\right)$
    $L \leftarrow L + \text{CrossEntropy}\left(P, y\right)$
  end for
  Perform Adam optimizer on W to minimize L
end while

Algorithm 2 provides the pseudocode for training our PTN. Once the tasks are defined, we first randomly initialize the weights W of our neural network f_W. Then we learn these parameters in multiple epochs. In each epoch of training, we randomly sample a task and calculate the embedding for both support and query sets through the neural network. Then, we find the prototype for each label as the mean vector of the support set embeddings corresponding to each class or label denoted by $\rho(k)$ for label k. Then, given each sample of the query set, we find the distance between the embedding of this sample and a prototype k as $d\left(f_W(X), \rho(k)\right)$. The array of distances between the embedding of the query sample and the class prototypes $d\left(f_W(X), \rho\right)$, represents the dissimilarities.

To transform these distances into a similarity measure, we negate them prior to applying the softmax function. The resulting softmax output yields the likelihood $p(y_i|x)$ of the class labels given the query sample. Given this and the ground-truth label, we could obtain the CrossEntropy and update the parameters using the Adam optimizer [26]. We ultimately assess the PTN model's generalizability by applying it to target tasks which have been not seen during the training phase.

IV. Evaluation

In order to evaluate our method, experiments were conducted on the CICIDS2017 and CICIDS2018 datasets [24]. CICIDS2018 is an extension of the CICIDS2017 dataset, containing additional network traffic data. These datasets originate from distinct networks, each representing different hardware and software environments, and feature varying types of attacks. As a result, detecting intrusions becomes more challenging, allowing a thorough evaluation of the adaptability of our proposed method. The CICIDS2017 dataset was generated from real network recordings. The similarities between CICIDS2017 and CICIDS2018 in terms of temporal and spatial characteristics, as well as collection configuration, provide the necessary conditions to conduct the evaluation experiment. Notably, the CICIDS2018 dataset is larger than the CICIDS2017 dataset, containing over 80 million flows compared to approximately 3 million flows in the CICIDS2017 dataset. For simplicity, we categorized the different types of attacks into six groups in both datasets. Table I offers a detailed overview of this categorization.

TABLE I: Categorizing labels in datasets.

Label	Category in CICIDS2017	Category in CICIDS2018
Benign	Benign	Benign
DoS	DoS Hulk DoS Slowloris DoS Slowhttp DoS GoldenEye Heartbleed	DoS Hulk DoS Slowloris DoS Slowhttp DoS GoldenEye
Web Attack	BruteForce-XSS BruteForce-Web SQL Injection	BruteForce-XSS BruteForce-Web SQL Injection
DDoS	DDoS	DDoS-HOIC DDoS-LOIC-UDP DDoS-LOIC-HTTP
Brute-Force	FTP-Patator SSH-Patator	FTP-Patator SSH-Patator
Bot	Bot	Bot
PortScan	PortScan	-

All the kinds of attacks we selected had a sufficient amount of samples. According to the data sources of the meta-training and meta-testing sets, the experiments can be divided into some types as follows:

- *Type 1 (Zero-day Attack)*: These experiments are binary classifications performed on the CICIDS2017 dataset. Any traffic not labeled as benign is considered an attack.
- *Type 2 (Zero-day Attack)*: These experiments are multi-class classifications conducted on the CICIDS2017 dataset. Multi-class classification extends beyond the binary classification of benign versus attack traffic, aiming to not only detect non-benign traffic but also accurately identify the specific type of attack. For the multi-class classification experiments, we designate $n = 2$ and $n = 3$.
- *Type 3 (Domain-shift)*: These experiments are multi-class classification conducted on the basis of type 2 experiments but on both datasets. In this experiment, 6 types of attack traffic in CICIDS2017 were used as the source task in the meta-training set, and 6 similar types of attack traffic in CICIDS2018 were considered as the target task.

To preprocess the data for both datasets, several steps were undertaken. Firstly, data samples containing NaN values were eliminated. Secondly, data samples featuring negative values for attributes that necessitate non-negative values were discarded. Next, one-hot encoding is applied to the Protocol column, resulting in three new features. To avoid redundancy, only two of these characteristics are retained. Subsequently, columns representing Source IP, Source Port, Destination IP, and Destination Port were removed, leaving us with 78 features and one column designated for labeling. Finally, the features are standardized, a process involving centering them around zero by subtracting the mean and scaling them to unit variance. During the experiment, we consider n as the number of classes (attack types) in the target task, with the benign case always included in the target task.

A. Experimental Results on Different n and k-shot values

In this section, we explored three scenarios on CICIDS2017: Scenario 1 includes DDoS in the target task; Scenario 2 includes Web Attack and DoS; and Scenario 3 involves Web Attack, DoS, and PortScan. Table II shows the comparison of the baseline approach to PTN-IDS across different n and k-shot values. With $n = 1$, we perform a binary classification task. For $n = 2$ and $n = 3$, we have multi-class classification in target task. *Baseline* in this experiment refers to a use case in which we train our neural network on the source task in the traditional manner (train and test rather than meta-learning setting).

There are no support and query sets in the source data but there are support and query sets in the target task. It is a kind of binary classification in which the trained model is tested on the query sets of the target task as a binary classification without using any support set. Our methodology considers the problem as multi-class classification, but baseline is a binary classification which is a much simpler problem than multi-class classification. We define this kind of baseline to demonstrate that even in a more challenging problem, PTN-IDS outperforms other approaches over different n-values.

The results in Table II demonstrate that our proposed method with different k-shot significantly outperforms the baseline. A detailed sensitivity analysis indicates that employing 5 samples per label (5-shot) notably improves the metrics

TABLE II: Comparison of Baseline and Proposed Method across Different n-values

Models	Scenario1: n=1		Scenario2: n=2		Scenario3: n=3	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Baseline	0.6271	0.5814	0.5957	0.5488	0.5067	0.4541
1-shot Proposed	0.7918	0.7651	0.7014	0.6797	0.6345	0.5924
5-shot Proposed	0.9102	0.9067	0.8297	0.8232	0.7946	0.7785
10-shot Proposed	0.9312	0.9296	0.8445	0.8370	0.8186	0.8084

TABLE III: Comparison of using Different Distance Function in PTN with 5-shot.

Models	Scenario1: n=1		Scenario2: n=2		Scenario3: n=3	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Euclidean Distance	0.9102	0.9067	0.8297	0.8232	0.7946	0.7785
Manhattan Distance	0.8860	0.8779	0.8134	0.8035	0.7797	0.7682
Cosine Distance	0.9098	0.9048	0.7285	0.6964	0.7691	0.7560

TABLE IV: Accuracy of Different Methods on Domain Shift Problem.

Method	**Individual Classes**						**Overall Accuracy**
	Benign	**DDoS**	**BruteForce**	**Bot**	**Web**	**DoS**	
Baseline	0.9948	0.0399	0.0000	0.0000	0.0000	0.0000	0.1724
Proposed 5-shot	0.7069	0.8700	0.5404	0.5689	0.7649	0.5551	0.6677
Proposed 5-shot + Finetuning	0.8262	1.0000	0.9940	0.9588	0.9812	0.9446	0.9508

compared to the use of a single sample per label (1-shot). However, there is not much difference between 5-shot and 10-shot. This motivates us to use 5-shot in the rest of the paper. Specifically, when limited to just one sample (one-shot learning), our model achieves an accuracy of 79%, surpassing that of the baseline. Subsequently, with the inclusion of 10 samples ($k = 10$), the model exhibits a notable increase in accuracy to 93%, which significantly exceeds the baseline accuracy of 62%. The enhanced accuracy and F1-score confirm the reliability of the IDS in accurately identifying real attacks while minimizing both false positives and false negatives.

B. Experimental Results on Different Distance Function

Table III provides a comprehensive comparison of various distance functions within PTNs under a 5-shot learning scenario. The findings underscore the significance of selecting an appropriate distance metric in the context of PTNs. The results suggest that Euclidean distance outperforms both Manhattan distance and Cosine distance, exhibiting superior accuracy and F1-score performance. This observation underscores the efficacy of Euclidean distance in capturing the underlying relationships between query instances and prototype representations within the feature space. Therefore, we adopted Euclidean distance as the preferred distance function throughout the remainder of this paper. This strategic decision aims to enhance the robustness and efficacy of the proposed methodology in handling diverse FSL tasks across various domains.

C. Experimental Results on Varying Source Tasks for $n = 1$

In this experiment, we investigated a binary classification task ($n = 1$) using PTN-IDS. We included three attacks in the source task and placed another attack in the target task. According to the results in Fig. 6, the tasks involving Web Attack or PortScan as the target consistently show high performance across different combinations of source tasks and Web Attack has higher accuracy compared to PortScan. In contrast, tasks targeting BruteForce display variable performance levels depending on the source task combination. In particular, the accuracy decreases significantly when DoS is excluded from the source tasks for BruteForce as the target task.

D. Experimental Results on Zero-day Attacks

In this experiment, we evaluate the performance of PTN-IDS in the case of an un-seen attack in the target task. We consider two scenarios with different source and target tasks: In *Scenario 1*, source task is {DDoS, PortScan, Bot} and target task is {Web, BruteForce, DoS}. In *Scenario 2*, source task is {Web, BroutForce, DoS} and target task is {DDoS, PortScan, Bot}. The results shown in Fig. 7 demonstrate that our proposed methodology is able to classify the Zero-day (target) attacks with high accuracy for different scenarios while also maintaining high accuracy in identifying source attacks. It demonstrates that the model not only differentiates between attack types and benign activities, but also precisely identifies the specific type of attack that has occurred. As k increases, we see a further improvement in classification accuracy; however, the difference is not very significant between 5, 10, and 20.

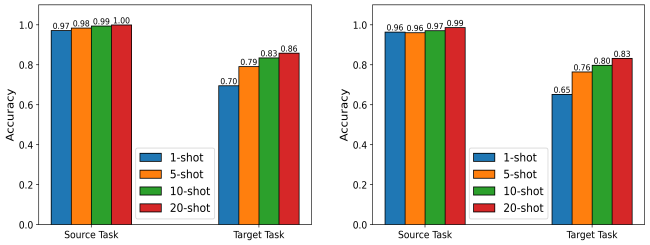
E. Experimental Results on Domain Shift Problem

CICIDS2017 and CICIDS2018 databases are from two networks corresponding to different hardware and software environments, and the types of attacks are also different between them. Therefore, the detection of attacks is more challenging in the case of domain shift. In this section, we evaluate the adaptability of the proposed method. Fig. 8 illustrates the results of our model with the source task on CICIDS2017 and the target task on CICIDS2018 for 1-shot and 5-shot.

Baseline in this experiment refers to a use case in which a Feed Forward Network has been trained on the source task

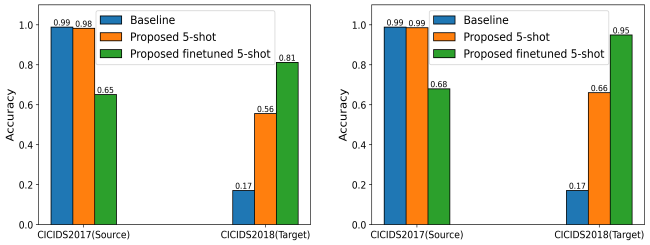
(a) Excluding PortScan Attack. (b) Excluding Web Attack. (c) Excluding DoS Attack.

Fig. 6: Comparing accuracy on varying sources and specific attack as target. The value in row i and column j represents the accuracy of detecting attack j in the target set, which consists of attacks i and j. The source set includes all attack types excluding attacks i and j, as well as the specific attack being excluded.



(a) Scenario 1 (b) Scenario 2

Fig. 7: Detecting Zero-day attack in different scenarios.



(a) 1-shot (b) 5-shot

Fig. 8: Evaluation for domain shift problem.

(CICIDS2017 data has been balanced in terms of labels) in a traditional manner as a multi-class classification and then tested on the query sets of the target task (CICIDS2018) without using the support sets. In the baseline, the model trained on CICIDS2017 struggles to generalize well on CICIDS2018 due to domain shift between the datasets, often misclassifying attacks as benign. However, our 5-shot learning PTN, without any fine-tuning, not only significantly improves the accuracy of CICIDS2018 compared to the baseline, but also maintains a high accuracy on the source task, CICIDS2017. When *Fine-tuned* method on a few tasks of CICIDS2018, our PTN outperforms other methodologies in accuracy on the CICIDS2018 dataset. However, this fine-tuning leads to a decrease in accuracy on the CICIDS2017 dataset, highlighting the challenges posed by the existing domain shift between the two datasets. Both 1-shot and 5-shot learning outperform the baseline.

The accuracy of various methods with 5-shot was evaluated across different classes of attacks on CICIDS2018, as summarized in Table IV. Accuracy values for benign case shows that our methodology involves a trade-off where the model sacrifices some accuracy on the benign class to significantly improve detection of underrepresented classes. This is a common scenario where enhancing sensitivity to rare events can reduce performance on more frequent ones. In the baseline method, the model misclassifies most of the attacks as benign case, resulting in significantly low accuracy. In particular with the detection of DDoS attacks, the baseline achieves an accuracy of only 3.99%. However, with the proposed 5-shot learning approach, significant improvements were observed across all types of attacks. For instance, the proposed 5-shot method achieved a remarkable accuracy of 87% in detecting DDoS attacks. Additionally, incorporating fine-tuning further enhanced the model's performance, leading to an accuracy of 100% for DDoS detection. This highlights the effectiveness of leveraging FSL and fine-tuning techniques in enhancing the accuracy of attack detection systems.

V. Conclusion

In this paper, we designed a new IDS based on a PTN. Embeddings of the support set and test data were generated with a neural network, and a similarity measurement was used to evaluate the distance between the test data embeddings and each embedding in the support embeddings. Our analysis showed that PTN-IDS, particularly with 5-shot learning, significantly outperformed the baseline method across different scenarios. The use of Euclidean distance in PTNs demonstrated superior performance compared to Manhattan distance and Cosine Distance, establishing it as the preferred distance function. Furthermore, our approach exhibited robustness in classifying Zero-day attacks and demonstrated adaptability to domain shift between datasets. We further fine-tuned the PTN to increase accuracy in the target task. Additionally, our study highlighted the effectiveness of FSL in scenarios with limited labeled data. Our method was able to reach a high accuracy, using 5 samples from each label without any fine-tuning.

REFERENCES

[1] N. Niknami and J. Wu, "Enhancing load balancing by intrusion detection system chain on sdn data plane," in *Proc. of the IEEE Conf. on Communications and Network Security (CNS)*, 2022, pp. 264–272.

[2] M. Ahmed, A. N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," *Journal of Network and Computer Applications*, vol. 60, pp. 19–31, 2016.

[3] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

[4] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM computing surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[5] F. Wei, H. Li, Z. Zhao, and H. Hu, "xnids: Explaining deep learning-based network intrusion detection systems for active intrusion responses," in *Proceedings of the 32nd USENIX Security Symposium (USENIX Security)*, 2023, pp. 4337–4354.

[6] J. Lansky, S. Ali, M. Mohammadi, M. K. Majeed, S. H. T. Karim, S. Rashidi, M. Hosseinzadeh, and A. M. Rahmani, "Deep learning-based intrusion detection systems: a systematic review," *IEEE Access*, vol. 9, pp. 101 574–101 599, 2021.

[7] D. Akgun, S. Hizal, and U. Cavusoglu, "A new ddos attacks intrusion detection model based on deep learning for cybersecurity," *Computers & Security*, vol. 118, p. 102748, 2022.

[8] L. Bilge and T. Dumitraş, "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*, 2012, pp. 833–844.

[9] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM computing surveys (csur)*, vol. 53, no. 3, pp. 1–34, 2020.

[10] R. Duan, D. Li, Q. Tong, T. Yang, X. Liu, and X. Liu, "A survey of few-shot learning: an effective method for intrusion detection," *Security and Communication Networks*, vol. 2021, pp. 1–10, 2021.

[11] Y. Xie, H. Wang, B. Yu, and C. Zhang, "Secure collaborative few-shot learning," *Knowledge-Based Systems*, vol. 203, p. 106157, 2020.

[12] A. Parnami and M. Lee, "Learning from few examples: A summary of approaches to few-shot learning," *arXiv preprint arXiv:2203.04291*, 2022.

[13] H. Gharoun, F. Momenifar, F. Chen, and A. H. Gandomi, "Meta-learning approaches for few-shot learning: A survey of recent advances," *arXiv preprint arXiv:2303.07502*, 2023.

[14] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[15] S. Sun, Q. Sun, K. Zhou, and T. Lv, "Hierarchical attention prototypical networks for few-shot text classification," in *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 2019, pp. 476–485.

[16] S. Mackey, T. Zhao, X. Wang, and S. Mao, "Cross-domain adaptation for rf fingerprinting using prototypical networks," in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, pp. 812–813.

[17] S. Xu, X. Han, T. Tian, B. Jiang, Z. Lu, and C. Zhang, "Few-shot network traffic anomaly detection based on siamese neural network," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2023, pp. 3012–3017.

[18] L. Du, Z. Gu, Y. Wang, L. Wang, and Y. Jia, "A few-shot class-incremental learning method for network intrusion detection," *IEEE Transactions on Network and Service Management*, 2023.

[19] Z. Shi, M. Xing, J. Zhang, and B. H. Wu, "Few-shot network intrusion detection based on model-agnostic meta-learning with l2f method," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2023, pp. 1–6.

[20] T. Althiyabi, I. Ahmad, and M. O. Alassafi, "Enhancing iot security: A few-shot learning approach for intrusion detection," *Mathematics*, vol. 12, no. 7, p. 1055, 2024.

[21] 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.

[22] J. Yang, H. Li, S. Shao, F. Zou, and Y. Wu, "Fs-ids: A framework for intrusion detection based on few-shot learning," *Computers & Security*, vol. 122, p. 102899, 2022.

[23] 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*, 2023.

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

[25] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning research*, vol. 9, no. 11, 2008.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.