DEML: Data-Enhanced Meta-Learning Method for IoT APT Traffic Detection

Jia Hu[1], Weina Niu[1,2(✉)], Qingjun Yuan[3], Lingfeng Yao[1], Junpeng He[1], Yanfeng Zhang[4], and Xiaosong Zhang[1,2]

[1] School of Computer Science and Engineering, Insitute for Cyber Security, University of Electronic Science and Technology of China (UESTC), Chengdu 611731, China
niuweina1@126.com

[2] Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518000, China

[3] Henan Key Laboratory of Network Cryptography Technology, and Key Laboratory of Cyberspace Security, Ministry of Education, Zhengzhou 450001, China

[4] Sichuan Police College, Intelligent Policing Key Laboratory of Sichuan Province, Luzhou 646000, China

Abstract. Advanced Persistent Threat (APT) is one of the most representative attacks that pose significant challenges to Internet of Things (IoT) security due to its stealthiness, dynamism, and adaptability. To detect IoT APT, machine learning-based methods are proposed to extract traffic features and mine attack semantics automatically. However, IoT APT traffic sample in actual scenarios is unbalanced and scarce, which affects the detection performance of existing methods. To resolve these challenges, we propose a data-enhanced meta-learning (DEML) method for detecting IoT APT traffic in this paper. Specifically, DEML uses non-functional feature-based generative adversarial network (NFGAN) to extend IoT APT traffic samples. DEML also uses a meta-learning model to further enhance the learning ability to IoT APT samples (including newly generated and original IoT APT traffic samples). We conduct experiments on a hybrid dataset where benign traffic comes from IoT-23 and APT traffic comes from Contagio. Experimental results show that our method outperforms the existing data enhancement methods. In addition, DEML achieves a detection accuracy of 99.35%, which is better than the baseline models in IoT APT traffic detection.

Keywords: IoT Security · APT traffic detection · Meta-learning · Generating adversarial networks

1 Introduction

In recent years, Internet of Things (IoT) has been applied to smart medical [1], smart agriculture [2], smart city [3], smart transportation [4] and other fields. The widespread adoption of IoT has brought many benefits, but it has also

S. Goel and P. R. Nunes de Souza (Eds.): ICDF2C 2023, LNICST 570, pp. 212–226, 2024.
https://doi.org/10.1007/978-3-031-56580-9_13

raised serious security concerns due to the vulnerability of IoT devices. Advanced Persistent Threat (APT) is one of the most representative attacks that poses significant threats to the security of the IoT due to its variability, high impact and difficult defense. For example, in 2019, the Russian hacker group Sandworm Team attacked an energy company in Ukraine and caused serious damage to its power system by exploiting the company's IoT devices [5]. In July 2020, the UK's National Cyber Security Centre agency released a report revealing an APT attack launched by the Russian hacker group APT29 that targeted medical institutions, vaccine manufacturers, and research organizations in the UK in an attempt to steal sensitive information related to COVID-19 vaccines [6]. Therefore, IoT APT attack detection is particularly important in IoT security.

Log-based detection and network traffic-based detection are two commonly used APT detection methods [7]. Log-based detection method usually use pattern recognition and correlation analysis to find anomalies in log data [8]. However, they often require large amounts of memory and computing resources to store and analyze massive amounts of log data, which brings challenges to resource-constrained IoT environments [9]. Traditional network traffic-based detection methods use predefined rules or signatures to detect network attacks, but they usually require manual update and maintenance rules or signatures [10]. To address these issues, machine learning-based methods have been introduced. These methods typically use machine learning algorithms to automatically learn and recognize anomalous or malicious behaviors in network traffic [11]. They are not only applicable to networks of different sizes, but also capable of detecting complex and variable attacks. However, machine learning-based methods require a large amount of attack traffic data to learn their behavioral characteristics. In the actual network environment, the available IoT APT attack traffic samples are unbalanced and scarce, which affects the detection performance of existing machine learning-based methods.

Data augmentation is an effective method to improve the model performance when dealing with a small amount of data. A common method in data augmentation is generative adversarial network (GAN), which continuously improves the model performance by adversarial training between the generator and the discriminator, making the generated data similar to the real data. The realism of the data generated by GAN is proportional to the amount of training data. So, when training samples are scarce, GAN may perform poorly [12].

To address this problem, we propose a data-enhanced meta-learning (DEML) method, which uses a non-functional feature-based GAN (NFGAN) to augment IoT APT traffic for better training of meta-learning models. Specifically, DEML first divides the original IoT APT traffic feature vector into functional and non-functional parts based on the feature importance calculated by gradient boosting decision trees. Next, DEML combines the generated non-functional parts using meta-learning based NFGAN with the preserved functional features to obtain the generated APT traffic features, thereby enhancing their realism. Then, DEML adopts the meta-learning model using original traffic and generated APT attack

traffic to effectively discover APT traffic in the case of limited APT attack samples. In summary, our contributions are as follows:

1 We design a non-functional feature-based generative adversarial network (NFGAN) to mitigate the scarcity and imbalance of APT traffic data. It not only uses a meta-learning framework to enhance the learning of APT few sample feature, but also generates only the non-functional part of APT traffic sample to ensure the authenticity of the generated APT samples.
2 We propose a data-enhanced meta-learning method, called DEML, to detect APT attacks in IoT. DEML not only uses the generated APT samples through NFGAN to expand APT traffic, but also utilizes the meta-knowledge learned through multi-tasking to further enhance the model performance.
3 We conduct experiments on the hybrid dataset, and the experimental results show that our proposed approach achieves better performance than the state-of-the-art data enhancement methods. In addition, it outperforms commonly used machine learning models for APT traffic detection.

The remainder of this paper is arranged as follows: Sect. 2 describes the related work on APT detection. In Sect. 3, our proposed DEML is illustrated in detail. The analysis of the experimental results of DEML is presented in Sect. 4. In Sect. 5, conclusions are drawn.

2 Related Work

This section provides a review of related work on APT detection methods based on log analysis and network traffic analysis.

2.1 Log-Based Detection Methods

Log-based detection methods discover potential attacks through analyzing logs from monitored devices. For example, Niu *et al.* [13] proposed a method to detect APT malware command and control (C&C) domains by analyzing DNS logs. Li *et al.* [14] combined semantic embedding and temporal embedding to train a uniform attention-based BiLSTM model for log anomaly detection. Yang *et al.* [15] proposed a log-based anomaly detection method, PLELog, by combining attention mechanism and gated recursive network structure. Cheng *et al.* [16] proposed an APT Alert and Log Correlation Framework (APTALCM). The framework first used network posture to reconstruct APT attack scenarios. Then, the SimRank-based cyber situation instance similarity measurement was introduced to compute the similarity of network posture instances. APTALCM correlated APT alert instance logs based on similarities between instances to identify attacker intent. Li *et al.* [17] proposed a federal learning-based framework for APT prediction, APTPMFL. The framework was deployed in an edge computing environment to train a model using multiple APT attack patterns in a distributed learning fashion. The trained model can be implemented to predict the probability of APT attacks in IoT scenarios. However, log-based detection methods are limited in the IoT environment. This method is difficult to handle large and heterogeneous log data on resource-constrained devices [9].

2.2 Network-Based Detection Methods

Traditional network-based detection methods discover potential attacks combined network traffic characteristics with rules or signatures. For example, Liu *et al.* [18] proposed a probing routes-based approach, PRDSA, to detect sinkhole attacks in IoT. Lyu *et al.* [19] proposed an anomaly-based method to detect DoS attacks in IoT. Venkatraman *et al.* [19] designed a hybrid intrusion detection system based on timed automation controller, which successfully detected zero-day attacks, DoS attacks, and control hijacking attacks in IoT environments.

In recent years, machine learning-based detection methods have become increasingly popular, as they can automatically extract and mine traffic characteristics. For example, Okutan *et al.* [20] developed a Bayesian classifier-based network attack prediction system, CAPTURE. Huang *et al.* [21] put forward a risk assessment method based on Bayesian networks to quantify the impact of cyber attacks on Industrial cyber-physical systems (ICPS). Huang *et al.* [22] put forward a multi-stage Bayesian game framework to capture incomplete information about deceptive APTs and their multi-stage movements. Wang *et al.* [23] found that HTTP-based C&C is widely used in APT. Based on the fact that C&C domains are often accessed independently, they distinguished HTTP-based C&C communication from normal HTTP requests. The existing attack traffic detection method based on machine learning has achieved good results, but the sample size will affect the model performance. However, the available APT traffic samples in actual IoT scenarios are unbalanced and scarce.

3 Proposed Approach: DEML

This section introduces our proposed APT malicious traffic detection method in IoT: DEML. It is composed of three key components: Data Pre-processing, Data Augmentation and Malicious Traffic Detection. Figure 1 shows the framework of DEML.

3.1 Data Pre-processing

The Data Pre-processing phase consists of two parts: (1) feature extraction and processing, and (2) multi-task set construction.

In order to facilitate model training from effective features, we need to preprocess the captured network traffic. First, feature vectors are extracted from the original Pcap using CICflowmeter. Then, we use normalization and one-hot encoding to handle discrete and continuous feature values in the feature vector respectively, and finally obtain the processed feature vectors x. Among them, the normalized formula is given in Eq. (1). In particular, for the feature of the communication protocols "Protocol", we uniformly set the protocol value whose occurrence times are lower than the set threshold as other protocol, and its expression is "Others Protocol".

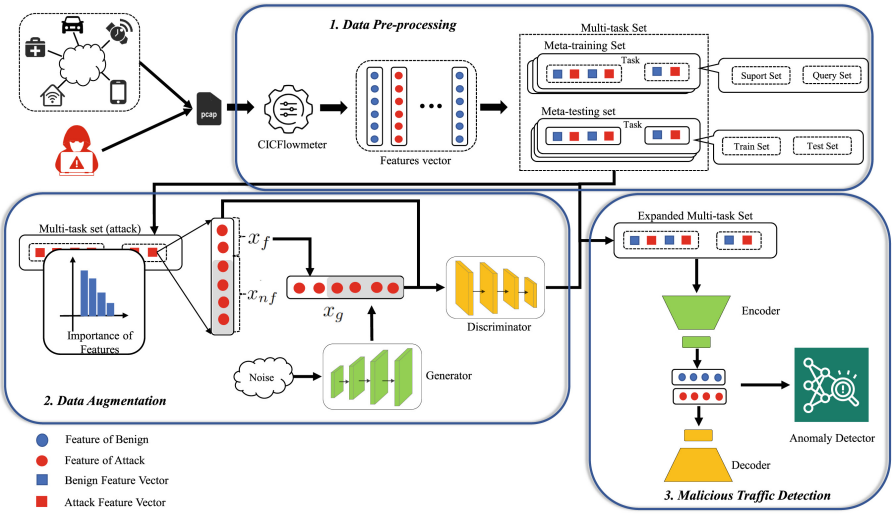


Fig. 1. The framework of DEML

$$x^{'} = \frac{x - min(x)}{max(x) - min(x)} \tag{1}$$

Multi-task set is built using x, each task containing benign traffic and a class of attack traffic. In addition, Multi-task set are categorized into meta-training set and meta-testing set. Among them, the former consists of the attack classes with the most attack traffic and benign traffic, while the latter includes all the attack classes and benign traffic. The meta-training set contains two subsets: the support set and the query set. The support set is used to train the initial model, while the query set is used to correct the model to prevent overfitting. The meta-testing set contains two subsets: the train set and the test set. The train set is used to fine-tune the model, while the test set is used to verify the model performance. In addition, another multi-task set consisting of attack samples is created in the same way to train the subsequent GAN.

3.2 Data Augmentation

In order to ensure that the generated attack instances retain the features of the original attack instances as much as possible, we retain the functional features of the original attack [24]. Thus, we divide the original attack feature vector x into functional features x_f and non-functional features x_{nf} through statistical analysis of the datasets. Specifically, we use gradient boosting decision tree (GBDT) to calculate the importance of each feature for each attack type [25]. Then, x is divided into x_f and x_{nf} according to the importance of these features.

Because x_f represents the functionality of the attack vector, if it is changed, it will significantly change the attack characteristics. Thus, GAN only need to focus on the generation of the non-functional part. Firstly, the x_f part of the

attack vector is obtained by the feature division. Then, a Gaussian noise of the same dimension as x_{nf} is randomly generated and fed into the generator G. The output features vector from G concated to x_f as the generated vector of attack instance, which is represented as x_g. x_g is calculated by Eq. (2), where F is a function for concating vectors. Lastly, x_g is sent to the discriminator Dis to verify its reality.

In the process of generating the above attack instances, G and Dis form a generative adversarial network and train against each other. On the one hand, G has to generate the feature vector x_g to bypass Dis. On the other hand, Dis should try to distinguish x_g from x. Therefore, the loss function of G is defined using the formula (3), where M represents the size of the attack set. The loss function of Dis is defined using Eq. (4). Specifically, the training process of the proposed NFGAN is shown in Algorithm 1. The first part of the algorithm (lines 2–14) is to train the initial NFGAN using the support set and to correct the NFGAN model with the query set. The second half of the algorithm (lines 14–20) fine-tunes the model with the training set to obtain the trained NFGAN. Finally, the trained NFGAN is used to expand the APT traffic data.

$$x_g = F(x_f, G(n)) \tag{2}$$

$$\mathcal{L}_G = \frac{1}{M}\sum_{i=1}^{M}\left\{log\left[1 - Dis(x_g^i)\right]\right\} \tag{3}$$

$$\mathcal{L}_{Dis} = \frac{1}{M}\sum_{i=1}^{M}\left\{log\left[Dis(x^i)\right] + log\left[1 - Dis(x_g^i)\right]\right\} \tag{4}$$

3.3 Malicious Traffic Detection

This section details the malicious traffic detection part of the DEML. It consists of an autoencoder and an Abnormal classifier (A). When new network traffic arrives, their feature vector x is extracted through data pre-processing. Then, x is mapped to the low-dimensional space by the autoencoder E to obtain the latent vector z. z input into the Abnormal classifier (A) to determine whether the newly arrived traffic is normal or malicious. Among them, A is a meta-learning-based classifier, whose loss function is shown in Eqs. (5), where M denotes the number of feature vectors in the set, and y indicates the label of table x, which takes the value 0 or 1, x^i denotes the i-th vector. The loss function of the autoencoder is defined in Eq. (6). The training process of the autoencoder and A is described in detail in Algorithm 2. First, the model parameters are randomly initialized as $\theta_E^{'}$, $\theta_D^{'}$, $\theta_A^{'}$. Then a branching task T_B is selected from the meta-trainning set for model training. $\mathcal{L}_{re}$ and $\mathcal{L}_A$ are calculated on each task support set in the branch and used to update the model parameters $\theta_E^{'}$, $\theta_D^{'}$, $\theta_A^{'}$ once (line 4–11). When the support set for all tasks in a branch has been trained, the model parameters $\theta_E^{'}$, $\theta_D^{'}$, $\theta_A^{'}$ are corrected once to take advantage of the average loss across the branch (line 12–16) . Finally, the optimized model parameters θ_E, θ_D, θ_A are obtained by fine-tuning using the train set (line 18–26).

Algorithm 1. Pseudocode of the NFGAN process

Input: the multi-task set of the attack, task batch size B, Number of training epochs (train_epoch)
Output: trained NFGAN

for *epoch* = 1 to *train_epoch* **do**
  for every batch T_B in meta-training set **do**
    for every task t_i in T_B **do**
      get x^i from the support set
      get x_g^i by Equation(2)
      get $\mathcal{L}_G^i, \mathcal{L}_{Dis}^i$ by Equation(3)(4)
      update G and Dis using the loss
    end for
    get x^i from the query set
    get x_g^i by Equation(2)
    using x^i and x_g^i test G and Dis for each task in T_B
    get the total loss function $\sum_{i=1}^{B} \mathcal{L}_G^i$, $\sum_{i=1}^{B} \mathcal{L}_{Dis}^i$
    update G and Dis using the loss $\frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_G^i$ and $\frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{Dis}^i$
  end for
  for every task in meta-testing set **do**
    get x from the support set
    get x_g by Equation(2)
    get $\mathcal{L}_G, \mathcal{L}_{Dis}$ by Equation(3)(4)
    update G and Dis using the loss
  end for
end for
return trained NFGAN

$$\mathcal{L}_A = \frac{1}{M}\sum_{i=1}^{M}\left\{y * log\left[A(E(x^i))\right] + (1-y) * log\left[1 - A(E(x^i))\right]\right\} \tag{5}$$

$$\mathcal{L}_{re} = \frac{1}{M}\sum_{i=1}^{M}\left\{D[E(x^i)] - x^i\right\} \tag{6}$$

4 Experimental Evaluation

4.1 Datasets

There are several public datasets for intrusion detection research in IoT, such as IoT-23 [26], Bot-IoT [27], CTU-13 [28] and N-BaIoT [29]. However, these datasets lack pure APT attack traffic. In order to obtain an usable APT dataset, Katharina *et al.* [30] created a hybrid dataset by merging a benign dataset with an APT dataset as the background. Thus, we constructed our experimental dataset by adopting this strategy, where the APT dataset was from Contagio malware database contributed by Mila Parkour [31] and the benign data was from the IoT-23 dataset created by Stratosphere Laboratory CTU University. The details of the constructed experimental dataset are shown in Table 1.

Algorithm 2. Parameter search algorithm

Input: the multi-task set, subset size M , task batch size B, learning rates η, λ, ξ;
Output: optimized θ_E, θ_D, θ_A

Randomly initialize $\theta_E^{'}$, $\theta_D^{'}$, $\theta_A^{'}$
for $epoch = 1$ to $train\_epoch$ **do**
  for every batch T_B in meta-training set **do**
    for every task t_i in T_B **do**
      $\theta_E, \theta_D, \theta_A, \leftarrow \theta_E^{'}, \theta_D^{'}, \theta_A^{'}$
      Get $(X, Y) = \{(x, y)^1, ..., (x, y)^M\}$ $y \in \{0, 1\}$ from the support set
      Get $\mathcal{L}_{re}, \mathcal{L}_A$ by Equation(6)(5)
      $\theta_{E,D}^i \leftarrow \theta_{E,D} - \lambda$
      $\theta_A^i \leftarrow \theta_A - \lambda \nabla_{\theta_A} \mathcal{L}_A(\theta_A)$
    end for
    Get $(X, Y) = \{(x, y)^1, ..., (x, y)^M\}$ $y \in \{0, 1\}$ from the query set
    Get $\mathcal{L}_{re}, \mathcal{L}_A$ by Equation
    Update $\theta_{E,D}^{'}$ by $\theta_{E,D}^{'} - \eta \frac{1}{B} \sum_{i=0}^{B} \nabla_{\theta_{E,D}^i} \mathcal{L}_{re}(\theta_{E,D}^i)$
    Update $\theta_A^{'}$ by $\theta_A^{'} - \eta \frac{1}{B} \sum_{i=0}^{B} \nabla_{\theta_A^i} \mathcal{L}_A(\theta_A^i)$
  end for
  for every task in meta-testing set **do**
    Get $(X, Y) = \{(x, y)^1, ..., (x, y)^M\}$ $y \in \{0, 1\}$ from the train set
    Get $\mathcal{L}_{re}, \mathcal{L}_A$ by Equation(6)(5)
    Update $\theta_{E,D}$ by $\theta_{E,D}^{'} - \xi \frac{1}{B} \sum_{i=0}^{B} \nabla_{\theta_{E,D}^i} \mathcal{L}_{re}(\theta_{E,D}^i)$
    Update θ_A by $\theta_A^{'} - \xi \frac{1}{B} \sum_{i=0}^{B} \nabla_{\theta_A^{'}} \mathcal{L}_A(\theta_A^{'})$
  end for
end for

Table 1. Details of the synthetic dataset

Type	Ratio(%)	Source	Type	Ratio(%)	Source
Benign	86.55	iot-23	Hupigon	0.46	Contagio
TrojanCookies	5.35	Contagio	Gh0st_variant	0.30	Contagio
Pingbed	1.73	Contagio	Taidoor	0.31	Contagio
LURK	1.42	Contagio	PlugX	0.22	Contagio
Mediana	1.34	Contagio	Nettravler	0.21	Contagio
PDF_CVE	0.69	Contagio	Sanny_Daws	0.21	Contagio
Xinmic	0.65	Contagio	RssFeeder	0.16	Contagio
8202	0.40	Contagio			

4.2 Experimental Setup

Evaluation Environment: DEML is evaluated on a 12-core Intel(R) Core(TM) i9-10920X CPU @ 3.50 GHz with 256 GB of RAM and the Ubuntu 20.04 LTS operating system with Linux kernel v.5.11.0. PyTorch v1.7.1 is chosen to implement related experiments with Jupyter notebook.

Evaluation Metrics: In this paper, we adopt the widely used metrics for evaluating the performance of DEML: Accuracy (Acc), Precision (Pre), Recall (Rec), F1-score (F1), True Positive Rate (TPR) and False Positive Rate (FPR). Accuracy (Acc) is the proportion of the number of correct detected APT traffic and benign traffic to the number of all the network traffic, which is calculated by Eq. (7). Precision (Pre) is the proportion of the number of correct detected APT traffic to the number of the detected APT traffic, whose calculation formula is shown in Eq. (8). Recall (Rec) is the proportion of the number of correct detected APT traffic to the number of the actual APT traffic, whose calculation formula is shown in Eq. (9). F1-score (F1) is a comprehensive evaluation metric using the weighted and averaged recall and accuracy, which is calculated through Eq. (10). TPR reflects the ratio of correctly detected APT traffic, and FPR shows the ratio of the benign traffic that are incorrectly classified as APT traffic. Receiver Operating Characteristic (ROC) curve is used to visualize the relationship between TPR and FPR, and its Area Under Curve (AUC) is used to measure the performance of the model. Precision Recall Curve (PRC) is used to reflect the relationship between Pre and Rec. Besides, we use Fréchet Inception Distance (FID) to measure the authenticity of the generated data. FID denotes the distance between the generated data and the real data in the feature space, which is calculated by Eq. 12, where μ represents the feature mean, Tr represents the trace of the matrix, and C represents the covariance matrix of the feature vector (x and x_g represent the real data and the generated data). The smaller the FID value of the generated data, the closer it is to the real data.

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \tag{7}$$

$$Pre = \frac{TP}{TP + FP} \tag{8}$$

$$Rec = TPR = \frac{TP}{TP + FN} \tag{9}$$

$$F1 = 2 * \frac{Rec * Pre}{Rec + Pre} \tag{10}$$

$$FPR = \frac{FP}{FP + FN} \tag{11}$$

$$FID(x, x_g) = \|\mu_x - \mu_g\|_2^2 + Tr(C_x + C_{x_g} - 2 * \sqrt{(C_x * C_{x_g})}) \tag{12}$$

Baseline Setting: To evaluate the effectiveness of DEML on data enhancement, we compared it with several existing data enhancement methods: G-IDS [32] solved the problem of data imbalance in the cyber-physical system by using basic GAN for traffic data generation to improve the performance of IDS. FCW-GAN [33] retained the important features of traffic data based on the feature

importance calculated by XGBoost. It then generated data on a few classes of attack traffic using CWGAN to reduce the impact of unbalanced data. SIGMA [24] used the same feature segmentation as DEML, and then adopted basic GAN to generate adversarial samples to enhance the detection performance of IDS. ML-CGAN [12] integrated a meta-learner structure into the conditional GAN (CGAN) backbone to improve the quality of the generated images when the training data was scarce. Moreover, we also selected several algorithms (SVM, RF, DNN, GDBT, AdaBoost) that perform well in classification tasks as baseline models to compare their performance in APT detection.

4.3 Evaluation Results

Comparative Experiments with Different Data Enhancement Algorithms: To evaluate the performance of different data enhancement methods, we use a meta-learning model trained with the original data to detect generated APT traffic. Besides, we also calculate the FID values of the generated data using five different data enhancement algorithms and count their total training time consumed after training 100 epochs. Table 2 shows the performance of the five methods. We can see that the detection model achieves only 85.12% Acc, 80.41% F1 for the generated data using traditional GAN (G-IDS) and G-IDS has the highest FID value, which indicates that the generated data has a large deviation from the real data. This is because the traditional GAN can only learn limited knowledge from a restricted number of samples in the training phase. Optimized GANs (including FCWGAN, SIGMA) outperform G-IDS, but still have a higher FID. This suggests that adding labels and retaining functional features when training the GAN can slightly improve performance. The meta-learning based GAN (ML-CGAN and NFGAN) have lower FID values and better performance compared to other types of GANs. This is because these models can use meta-knowledge learned from previous data to enhance the GAN's learning of features of new data. Moreover, our proposed method has a lower FID value, shorter training time and better accuacy compared to ML-GAN. This is because NFGAN not only ensures the authenticity of the generated APT samples by generating only the non-functional part of the samples, but also reduces the dimensionality of the data training. Therefore, meta-learning based NFGAN is more suitable for scenarios with sparse and unbalanced APT traffic samples.

Comparative Experiments with Different Classification Models: Under the same experimental conditions, the baseline algorithms are also trained on the NFGAN-enhanced dataset. The experimental results are shown in the Table 3. We can see that the proposed DEML achieves 99.35% Acc, 99.57%Pre, 98.78% Rec and 99.39% F1. In general, Combined with meta-learning-based DEML achieves competitive accuracy compared to traditional machine learning and deep learning algorithms such as SVM, RF, DNN. They also perform well compared with ensemble learning-based classification methods, such as GDBT and AdaBoost. This is because the meta-learning based DEML can use the meta-knowledge learned from previous data to enhance learning for the new data. In

Table 2. Comparison of data enhancement algorithms

Method	Acc	F1	FID	Trainning time
G-IDS [32]	85.12%	80.41%	283.42	91.8 s
FCWGAN [33]	89.87%	85.87%	245.56	54.82 s
SIGMA [24]	90.17%	86.99%	215.15	81.73 s
ML-CGAN [12]	93.84%	91.40%	64.63	61.76 s
Our NFGAN	**95.78%**	**93.12%**	**23.72**	**49.15 s**

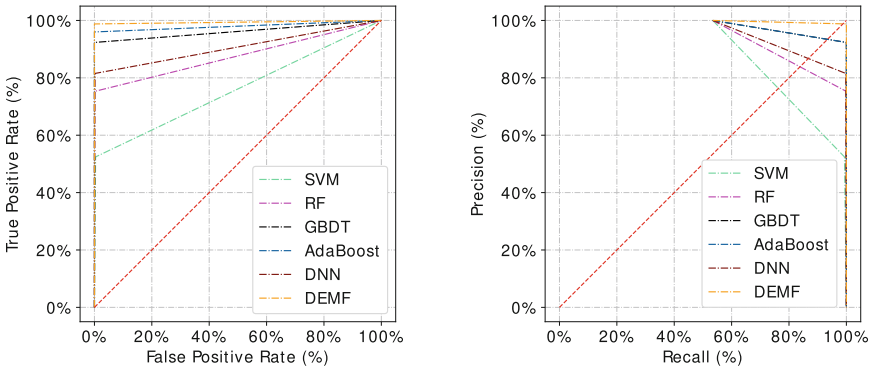
addition, Fig. 2 shows the ROC curves and PRC curves for different classification models. According to Fig. 2(a), we find that DEML has a higher false alarm rate compared to the baseline algorithm when achieving the same accuracy. The PRC curve of DEML is better than the other baseline algorithms as can be seen from Fig. 2(b). This indicates that DEML has higher detection accuracy when all methods achieve the same recall. These results shows that DEML is more suitable for IoT APT traffic detection.

Table 3. Abnormality detection performance of DEML and baseline methods (%)

Method	Acc	Pre	Rec	F1
SVM	74.39	99.24	52.44	74.7
RF	84.8	88.5	84.8	84.6
GBDT	95.9	99.14	93.34	96.01
AdaBoost	97.86	99.25	95.99	97.96
DNN	86.69	99.17	75.09	85.77
DEML	**99.35**	**99.57**	**98.78**	**99.39**

4.4 Ablation Study

Our DEML contains three main components: data pre-processing, data augmentation and malicious traffic detection. In the data augmentation part, the generative adversarial network is used to expand the sparse APT traffic data. In the malicious traffic detection part, the meta-learning based detection model can achieve fast learning for few samples through multi-task learning. To investigate the impact of meta-learning framework and feature partitioning in the model, we design three variants of DEML: DEML_1, DEML_2 and DEML_3. DEML_1: The feature vector of the traffic data is fitted directly using Gaussian noise without the feature partitioning on the data enhancement. DEML_2: Instead of using a meta-learning framework for data augmentation, a basic GAN is used to generate the non-functional part of the traffic feature vector. DEML_3: NFGAN is used for data enhancement. And traditional DNNs are used to detect malicious traffic.



(a) ROC of DEML and all baselines (b) PRC of DEML and baselines

Fig. 2. ROC and PRC of DEML and baselines

The experimental results of each variant on the hybrid dataset are shown in Table 4. It can be seen that DEML performs better than $DEML_1$. This suggests that NFGAN can improve the quality of the generated data by generating only the non-functional part of sample. $DEML_2$ and $DEML_3$ are both worse than DEML, which demonstrates that meta-learning framework can greatly improve model performance not only in data augmentation but also in detection model.

Table 4. Performance of DEML with different ablation settings (%)

Method	Acc	Pre	Rec	F1
$DEML_1$	94.75	95.71	93.64	94.82
$DEML_2$	89.54	91.48	89.52	89.57
$DEML_3$	86.69	99.17	75.09	85.77
DEML	**99.35**	**99.57**	**98.78**	**99.39**

5 Conclusion

In this paper, we proposed DEML, a Data-Enhanced Meta-Learning approach for detecting IoT APT traffic. DEML used NFGAN to expand IoT APT traffic samples to mitigate the impact of unbalanced data on the model. NFGAN only generated the non-functional part of the APT traffic samples and retained its functional part. In this way, the authenticity of the generated APT samples was guaranteed. In addition, DEML used meta-knowledge learned from the extended samples to enhance the model's ability to discover APT traffic in the absence and imbalance of original attack samples. We constructed a hybrid dataset to

validate the performance of DEML in terms of data enhancement and detection of APT traffic.

Extensive experimental results show that: 1) our deisgned NFGAN outperforms the existing traffic data enhancement methods; 2)Retaining functional features and generating only non-functional features can further improve the authenticity of generated APT traffic; 3) using meta-learning models can further improve the detection rate when APT traffic samples are scare and unbalanced.

Acknowledgements. This work was partially supported by the Opening Project of Intelligent Policing Key Laboratory of Sichuan Province (No. ZNJW2023KFQN003) and the Hennan Key Laboratory of Network Crytography Technology (No. LNCT2020-A02).

References

1. Al-Turjman, F., Nawaz, M.H., Ulusar, U.D.: Intelligence in the internet of medical things era: a systematic review of current and future trends. Comput. Commun. **150**, 644–660 (2020)
2. Sinha, A., Shrivastava, G., Kumar, P.: Architecting user-centric internet of things for smart agriculture. Sustain. Comput. Inform. Syst. **23**, 88–102 (2019)
3. Liu, K., Bi, Y.R., Liu, D.: Internet of things based acquisition system of industrial intelligent bar code for smart city applications. Comput. Commun. **150**, 325–333 (2020)
4. Babar, M., Arif, F.: Real-time data processing scheme using big data analytics in internet of things based smart transportation environment. J. Ambient Intell. Humaniz. Comput. **10**(10), 4167–4177 (2019)
5. Greenberg, A.: Sandworm: A New Era of Cyberwar and the Hunt for the Kremlin's Most Dangerous Hackers. Doubleday (2019)
6. National CyberSecurity Centre. Advisory: Apt29 targets COVID-19 vaccine development (2020). https://media.defense.gov/2020/Jul/16/2002457639/-1/-1/0/NCSC_APT29_ADVISORY-QUAD-OFFICIAL-20200709-1810.PDF
7. Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D.: A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. IEEE Commun. Surv. Tutor. **21**(2), 1851–1877 (2019)
8. Zhaoxue, J., Tong, L., Zhenguo, Z., Jingguo, G., Junling, Y., Liangxiong, L.: A survey on log research of aiops: methods and trends. Mob. Netw. Appl. **26**(6), 2353–2364 (2021)
9. Singh, P., et al.: Using log analytics and process mining to enable self-healing in the internet of things. Environ. Syst. Decis. **42**(2), 234–250 (2022)
10. Myneni, S., et al.: DAPT 2020 - constructing a benchmark dataset for advanced persistent threats. In: Wang, G., Ciptadi, A., Ahmadzadeh, A. (eds.) MLHat 2020. CCIS, vol. 1271, pp. 138–163. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59621-7_8
11. Alqudah, N., Yaseen, Q.: Machine learning for traffic analysis: a review. Procedia Comput. Sci. **170**, 911–916 (2020)
12. Ma, Y., Zhong, G., Liu, W., Wang, Y., Jiang, P., Zhang, R.: ML-CGAN: conditional generative adversarial network with a meta-learner structure for high-quality image generation with few training data. Cogn. Comput. **13**, 418–430 (2021)

13. Niu, W., Zhang, X., Yang, G., Zhu, J., Ren, Z.: Identifying apt malware domain based on mobile DNS logging. Math. Probl. Eng. **2017** (2017)
14. Li, X., Chen, P., Jing, L., He, Z., Yu, G.: Swisslog: robust and unified deep learning based log anomaly detection for diverse faults. In: 2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE), pp. 92–103. IEEE (2020)
15. Yang, L., et al.: Semi-supervised log-based anomaly detection via probabilistic label estimation. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE), pp. 1448–1460. IEEE (2021)
16. Cheng, X., Zhang, J., Chen, B.: Cyber situation comprehension for IoT systems based on apt alerts and logs correlation. Sensors **19**(18), 4045 (2019)
17. Li, Z., Cheng, X., Zhang, J., Chen, B.: Predicting advanced persistent threats for IoT systems based on federated learning. In: Wang, G., Chen, B., Li, W., Di Pietro, R., Yan, X., Han, H. (eds.) SpaCCS 2020. LNCS, vol. 12382, pp. 76–89. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-68851-6_5
18. Liu, Y., Ma, M., Liu, X., Xiong, N.N., Liu, A., Zhu, Y.: Design and analysis of probing route to defense sink-hole attacks for internet of things security. IEEE Trans. Netw. Sci. Eng. **7**(1), 356–372 (2018)
19. Lyu, C., Zhang, X., Liu, Z., Chi, C.-H.: Selective authentication based geographic opportunistic routing in wireless sensor networks for internet of things against dos attacks. IEEE Access **7**, 31068–31082 (2019)
20. Okutan, A., Werner, G., McConky, K., Yang, S.J.: POSTER: cyber attack prediction of threats from unconventional resources (CAPTURE). In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, pp. 2563–2565 (2017)
21. Huang, K., Zhou, C., Tian, Y.-C., Yang, S., Qin, Y.: Assessing the physical impact of cyberattacks on industrial cyber-physical systems. IEEE Trans. Ind. Electron. **65**(10), 8153–8162 (2018)
22. Huang, L., Zhu, Q.: Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks. ACM SIGMETRICS Perform. Eval. Rev. **46**(2), 52–56 (2019)
23. Wang, X., Zheng, K., Niu, X., Wu, B., Wu, C.: Detection of command and control in advanced persistent threat based on independent access. In: 2016 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2016)
24. Msika, S., Quintero, A., Khomh, F.: Sigma: strengthening ids with GAN and metaheuristics attacks. arXiv preprint arXiv:1912.09303 (2019)
25. Rao, H., et al.: Feature selection based on artificial bee colony and gradient boosting decision tree. Appl. Soft Comput. **74**, 634–642 (2019)
26. Erquiaga, M.J., Garcia, S., Parmisano, A.: IoT-23: a labeled dataset with malicious and benign IoT network traffic (2020). http://doi.org/10.5281/zenodo.4743746
27. Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B.: Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-iot dataset. Futur. Gener. Comput. Syst. **100**, 779–796 (2019)
28. Garcia, S., Grill, M., Stiborek, J., Zunino, A.: An empirical comparison of botnet detection methods. Comput. Secur. **45**, 100–123 (2014)
29. Meidan, Y., et al.: N-BaIoT-Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. **17**(3), 12–22 (2018)
30. Hofer-Schmitz, K., Kleb, U., Stojanović, B.: The influences of feature sets on the detection of advanced persistent threats. Electronics **10**(6), 704 (2021)
31. Mila. Collection of pcap files from malware analysis (2015). http://contagiodump.blogspot.com/2013/04/collection-of-pcap-files-from-malware.html

32. Shahriar, M.H., Haque, N.I., Rahman, M.A., Alonso, M.: G-IDS: generative adversarial networks assisted intrusion detection system. In: 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 376–385. IEEE (2020)
33. Wang, Y., Jiang, Y., Lan, J.: FCNN: an efficient intrusion detection method based on raw network traffic. Secur. Commun. Netw. **2021**, 1–13 (2021)