Poster: Enhanced Device Identification in Cellular IoT Using IPFIX Records

Ryuta Ohishi*, Norihiro Okui†, Masataka Nakahara†, Masakatsu Nishigaki*, Tetsushi Ohki*‡
*Shizuoka University, Hamamatsu, Shizuoka 432-8011, Japan
†KDDI Research,Inc, 2-1-15 Ohara, Fujimino, Saitama 356-8502, Japan
‡RIKEN AIP, Chuo, Tokyo 103-0027, Japan
Email: ohishi@sec.inf.shizuoka.ac.jp

***Abstract*—With the rapid growth of IoT devices, firmware vulnerabilities and malware infections have become critical concerns, potentially leading to unauthorized access and network breaches. To address this, accurate device identification and authentication are essential. This study proposes a method using payload-free IPFIX records to preserve privacy, combined with deep metric learning for device identification. The approach is validated in a cellular IoT environment with a dataset of 72 device models, leveraging angle-based deep metric learning algorithms.**

I. Introduction

The rapid growth of IoT devices has introduced critical security risks, including firmware exploits and malware infections. Attacks like Mirai's DDoS campaigns demonstrate these threats, with LTE router compromises [1] underscoring the need for stronger security in cellular IoT networks.

To facilitate the early detection of cyberattacks and to issue timely alerts, it is essential to accurately identify the types and characteristics of devices connected to the network. Internet service providers (ISPs) can estimate the devices connected in advance, enabling early warnings targeted at specific devices and the detection of anomalies on a per-device basis. However, most existing methods rely on payload-based traffic analysis [2], [3], raising privacy concerns and exhibiting limited adaptability to newly introduced devices. Furthermore, these approaches often fail to scale effectively in real-world environments with diverse and growing device populations.

To address these challenges, we propose a novel IoT device estimation method leveraging IPFIX(IP Flow Information Export) records and deep metric learning. By eliminating dependency on payload data, our approach ensures privacy and enhances scalability. The method is specifically designed to identify both known and unknown devices in cellular IoT environments, demonstrating superior adaptability and accuracy. The contributions of this paper can be summarized as following three items:

- Proposes a privacy-aware IoT device estimation method using IPFIX records, eliminating the need for payload data.
- Develops a scalable framework for identifying both known and unknown devices via deep metric learning.
- Demonstrates superior performance through experiments on a dataset of 72 IoT device models.

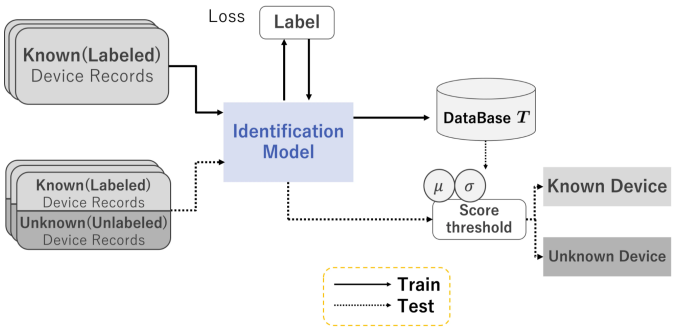


Fig. 1. Overview of our device estimation framework

II. Proposed Method

A. Overview

We propose a privacy-aware IoT device estimation method using IPFIX records and deep metric learning. This approach employs a one-dimensional convolutional neural network model and advanced loss functions, ArcFace and AdaCos, to improve classification performance for both known and unknown devices.

B. Dataset

We utilized the dataset obtained by organization affiliated with the authors (KDDI Research, Inc). It comprises IPFIX records collected over four years from 72 IoT devices. Each record represents a single communication session.On average, each device has 396,836 records, ranging from 490 to 4,616,185 records.

C. Feature Extraction

Feature vectors are extracted from IPFIX records using a 1D-CNN model. The input IPFIX flow data X is processed to generate a feature vector z as follows:

$$z = f_\theta(X) \tag{1}$$

Here, $f_\theta(\cdot)$ is the feature extractor trained to maximize inter-class separation and minimize intra-class variance.

TABLE I
METRICS AND VALUES FOR USE CASE 1

	Recall					
	R@1	R@2	R@4	R@8	AUC	EER
ArcFace	0.832	0.842	0.882	0.909	0.873	0.187
AdaCos	0.831	0.871	0.911	**0.954**	**0.904**	**0.154**

TABLE II
METRICS AND VALUES FOR USE CASE 2

		Detection of Unknown Devices					Identification of Known Devices			
	n	Accuracy	Precision	TPR	FPR	AUC	R@1	R@2	R@4	R@8
ArcFace	2	0.854	0.817	0.782	0.123	0.849	0.882	0.899	0.922	0.945
	3	0.778	0.762	0.695	0.184	0.821	0.852	0.867	0.892	0.911
AdaCos	2	**0.891**	**0.889**	**0.851**	**0.097**	**0.881**	**0.903**	**0.917**	**0.941**	**0.958**
	3	0.849	0.847	0.779	0.149	0.843	0.869	0.893	0.910	0.928

D. Model Architecture

Proposed method uses a 1D-CNN for feature extraction, with ArcFace enhancing inter-class separability via angular margins and AdaCos dynamically scaling to reduce intra-class similarity. These optimizations enable robust classification, even on imbalanced datasets.

E. Classification and Similarity Calculation

In the classification phase, the feature vector z of the target device is extracted using the trained 1D-CNN model. This vector is then compared with pre-registered templates $T = \{t_1, t_2, \cdots, t_k\}$, representing k known device templates. The device is classified based on the highest similarity score $\hat{k}$:

$$\hat{k} = \arg\max_{k} \text{clip}(\text{cossim}(z, t_k)). \tag{2}$$

Here, $\text{cossim}(\cdot, \cdot)$ calculates the cosine similarity between two vectors and $\text{clip}(\cdot)$ clips the output between [-1, 1]. Unknown devices are detect based on similarity threshold (μ, σ) using a normal distribution.

F. Threshold-based unknown device detection

A device is classified as unknown if its similarity $\hat{k}$ falls outside the range $[\mu - n\sigma, \mu + n\sigma]$. The parameter μ denotes the mean cosine similarity among the registered data, while σ represents the variance of cosine similarity among the registered data. The value n is an arbitrary non-negative number, and in this study, $n = 2$ or $n = 3$ is used to detect unknown devices effectively. This configuration ensures coverage probabilities of approximately 95% and 99.7%, respectively.

III. EVALUATION

In this study, we designed two use cases to assess the performance of our IoT device estimation method.

Use case 1 (Known Devices Only) This case focuses exclusively on known devices, simulating an environment where device templates are pre-registered.

Use case 2 (Including Unknown Devices) This case incorporates both known and unknown devices, representing a more realistic scenario where new, previously unseen devices may appear in the network.

These scenarios were designed to test the method's accuracy, scalability, and ability to handle unknown devices.

In use case 1, a dataset of 72 IoT device models was utilized for evaluation. The experiments employed 5-fold cross-validation, dividing the data into 80% for training and 20% for testing. Key metrics, including Recall@k, Area Under the Curve (AUC), and Equal Error Rate (EER), were used to evaluate performance. As shown in TABLE I, AdaCos outperformed ArcFace in all metrics, achieving a Recall@1 of 0.831 and a Recall@8 of 0.954. The AUC reached 0.904, and the EER was reduced to 0.154, showcasing AdaCos's superior classification capability.

In use case 2, it was designed to include 58 known device models and 14 unknown models, simulating a practical environment. Threshold settings of $n = 2$ and $n = 3$ were tested to classify unknown devices based on cosine similarity. As show in TABLE II, at $n = 2$, AdaCos achieved an accuracy of 0.89 for unknown device detection and a Recall@1 of 0.90 for known device identification. When the threshold was relaxed to $n = 3$, detection accuracy for unknown devices decreased slightly to 0.85, while Recall@8 for known devices remained robust at 0.93.

Across both use cases, the proposed method demonstrated high accuracy and adaptability. AdaCos consistently provided better results than ArcFace, particulary in environments with a mix of known and unknown devices. The ability to dynamically adjust thresholds allowed for flexibility in balancing detection sensitivity and accuracy.

IV. CONCLUSION

This study proposed a device estimation method for IoT environments using deep metric learning on 72 devices in a cellular IoT setting. The results showed high accuracy in known device estimation and effective detection of unknown devices. This method provides a foundation for anomaly detection and vulnerability reporting, with potential to quickly identify unknown threats. Future work should test its application in real-world environments.

ACKNOWLEDGEMENT

This study was supported in part by JST Moonshot JPMJMS2215.

REFERENCES

[1] B. N. (2025, 1) Mirai botnet exploiting routers 0-day vulnerabilities to launch ddos attack. [Online; accessed 2025-02-04]. [Online]. Available: https://cybersecuritynews.com/mirai-botnet-exploiting-routers-0-day-vulnerabilities/

[2] R. Bikmukhamedov and A. Nadeev, "Lightweight machine learning classifiers of iot traffic flows," in *2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, 2019, pp. 1–5.

[3] A. Sivanathan, H. H. Gharakheili *et al.*, "Classifying IoT devices in smart environments using network traffic characteristics," vol. 18, no. 8, pp. 1745–1759. [Online]. Available: https://ieeexplore.ieee.org/document/8440758?denied=

Enhanced Device Identification in Cellular IoT Using IPFIX Records

Ryuta Ohishi [1] Norihiro Okui [2] Masataka Nakahara [2]
Masakatsu Nishigaki [1] Tetsushi Ohki [1,3]
[1] Shizuoka University [2] KDDI Research,Inc [3] RIKEN AIP

Introduction

By identifying the types and charact eristics of devices connected to the network, it becomes possible to detect cyberattacks early and perform anomaly detection for each device.

- Payload-inclusive network traffic data is often used in research
  - **Privacy concerns** create resistance to providing such information
  - ✔ Proposes a privacy-aware IoT device estimation method using **IPFIX records** without payload data


```
{"flows":{"flowStart":"2019-06-21 02:31:19.799","flowEnd":"2019-06-25
08:36:51.236","srcIP":"xxx.xxx.xxx.xxx","srcPort":yyy,"dstIP":"zzz.zzz.zzz.zzz",
"pktCount":4,"byteCount":810,"revPktCount":4,"revByteCount":229,"TCPFlags":"APF",...}}
```


Example of an IPFIX Flow Record(1 session)

- Many studies are based on **closed-set** assumptions
  - ✔ Identify devices and accurately **detect unknowns** in multi-device environments
- Few studies consider environments with numerous connected devices
  - ✔ Shows promising results through experiments on a dataset of **72 IoT devices**

Method

- Input data: IPFIX (IP Flow Information Export) record data
- Identification using deep metric learning (loss functions: ArcFace, AdaCos)

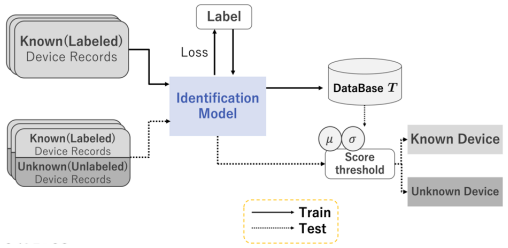
1 Registration Process

- Extract the feature vector of the target device
- Save the extracted data in the database as template data T

2 Verification Process

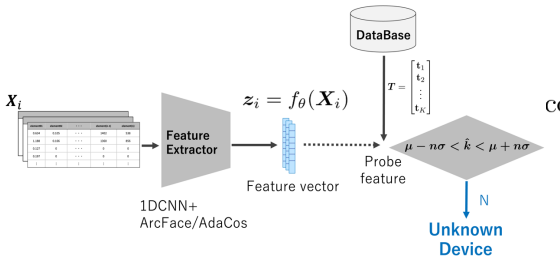
- Compare features with T using cosine similarity to identify known or unknown devices

Overview



After training, use the score threshold values $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$ to detect unknown devices

Detection of Unknown Device



$$\hat{k} = \arg\max_k \mathrm{clip}(\mathrm{cossim}(z, t_k))$$

$\mathrm{cossim}(z, t_k)$ Cosine similarity between the pre-registered template t_k and the input feature z_j

μ σ Mean and standard deviation of cosine similarity among registered data

Evaluation

Known Devices Only

- All devices within a specific network are pre-identified
- Record data obtained from a total of 72 types of IoT devices is used

▶ **Verification of Identification Performance with Multiple Devices**

- For Recall@8, AdaCos achieves 0.954, surpassing ArcFace's 0.909, **demonstrating accurate device identification without relying on payload data**
- Regarding AUC and EER, AdaCos demonstrates higher precision with fewer misclassifications.

Including Unknown Devices

- Devices connected to the system are not all pre-trained
- Known devices (trained): 58 types
- Unknown devices (untrained): 14 types
- n=2, 3 thresholds

▶ **Verification of Unknown Device Identification Performance**

- For unknown device detection, AdaCos achieves higher precision compared to ArcFace, **validating its effectiveness in open-set scenarios**
- For known device identification, AdaCos also demonstrates superior precision

	Recall					
	R@1	R@2	R@4	R@8	AUC	EER
ArcFace	0.832	0.842	0.882	0.909	0.873	0.187
AdaCos	0.831	0.871	0.911	0.954	0.904	0.154

Metrics and Values for Use Case 1

		Detection of Unknown Devices					Identification of Known Devices			
	n	Accuracy	Precision	TPR	FPR	AUC	R@1	R@2	R@4	R@8
ArcFace	2	0.854	0.817	0.782	0.123	0.849	0.882	0.899	0.922	0.945
	3	0.778	0.762	0.695	0.184	0.821	0.852	0.867	0.892	0.911
AdaCos	2	0.891	0.889	0.851	0.097	0.881	0.903	0.917	0.941	0.958
	3	0.849	0.847	0.779	0.149	0.843	0.869	0.893	0.910	0.928

Metrics and Values for Use Case 2

Conclusion

- Proposed an IoT device identification method using deep metric learning with IPFIX records
- Achieves high accuracy for known and unknown devices
  - Specifically, using AdaCos achieved better precision compared to ArcFace
- For unknown device detection, a stricter threshold setting (n=2) yielded higherprecision

Future Work

Examination of anomaly detection scenarios
Investigating anomaly detection methods using device estimation and feature extraction

Verification based on unidirectional flow records
Using unidirectional flow data only for device estimation/anomaly detection

Acknowledgement: This study was supported in part by JST Moonshot JPMJMS2215