Detecting Internet-Scale NATs for IoT Devices Based on Tri-Net

Zhaoteng Yan[1,2], Nan Yu[2](✉), Hui Wen[2], Zhi Li[2], Hongsong Zhu[2], and Limin Sun[2]

[1] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[2] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{yanzhaoteng,yunan,wenhui,lizhi,zhuhongsong,sunlimin}@iie.ac.cn

Abstract. Due to the lack of available labeled Network Address Translation (NAT) samples, it is still difficult to actively detect the large-scale NATs on the Internet. In this paper, we propose an novel method to identify NATs for online Internet of Things (IoT) devices based on Tri-net (a semi-supervised deep neural network). By learning the features on three layers (network, transport and application layer) in the small labeled data set (with thousands of instances), the Tri-net can automatically identify millions of online NATs. We evaluate this approach on the real-world dataset with more than 8 million online IoT devices, and the performance shows the precision and recall can be both up to 92%. Moreover, we found 2,511,499 IoT devices connecting to the Internet via NAT, which account for one-third of the total. To our knowledge, this is the first successful attempt to automatically identify Internet-scale NATs.

Keywords: NAT detecting · IoT devices · Tri-net

1 Introduction

The number of online IoT devices (e.g., IP camera, wireless router, etc.) has grown explosively. Meanwhile, limited public IPv4 addresses can not meet the needs of those devices connecting on the Internet. NAT has become the ideal solution that allows multiple devices sharing one public IP address or providing Internet-wide services via port mapping. However, NAT also undoubtedly bring some security issues to the online IoT devices: (1) NAT prevents the empirical investigation into counting the actual number of vulnerable devices in cyberspace. (2) Unauthorized NAT devices provide the convenience for the malicious code infection. Therefore, it is necessary to actively detect NAT behaviours of IoT devices in the cyberspace.

Consequently, NAT detection has become a typical and valuable research issue. Most studies focused on passive approaches on traffic traces [2,8,11,13]. That is because the performance evaluation of active detecting NAT devices

D. Yu et al. (Eds.): WASA 2020, LNCS 12384, pp. 602–614, 2020.
https://doi.org/10.1007/978-3-030-59016-1_50

is more uncontrollable and uncertain than passive measurement. As a result, the current research on active NAT detection approaches have been slightly fewer [9,11]. Moreover, there are still no usable approaches to identify NAT for the large-scale targets, especially for the online IoT devices in the complex cyberspace.

Motivation: In this paper, we aim to quantify the above assumption, and conduct an comprehensive Internet-scale study on active NAT detection toward online IoT devices with multiple features on network, transport, and application layers based on Deep Neural Network (DNN). Specifically, we try to answer the following questions: (1) *How many NATs are used by online IoT devices across IPv4 address space?* (2) *How many IoT devices connected to the Internet via their public IP addresses?* (3) *What is the distribution of geography, type, and protocol of those NATs?*

Challenges: To achieve this end, we need to address the three main challenges. One the unstable features for active detecting NATs has its limitation. For instance, IP identification (IP ID) and time-to-live (TTL) can be modified or forged. More importantly, due to the large heterogeneity in IoT devices, not all features on each layer can be obtained on the Internet. Besides, inadequacy features cause the lackness of labeled NAT samples for training the applicable machine learning modules. These main reasons make the active NAT detection seems to be impossible in the real world.

Method: To address these challenges, we design and implement an approach to identify NATs for online IoT devices in IPv4 space automatically. The workflow of our approach consists of three steps. First, we pre-label a small part of the dataset and extract the relatively features on network, transport, and application layers. Second, we train the modules of Tri-net by automatically feature learning and three pseudo-labels on diverse layers predicting. Third, we determine whether it is a NAT or straight-connecting device based on the trained Tri-net and the enlarged labeled dataset.

Results: To evaluate the performance, we implement our approach to the real Internet-wide public data. The data set contains 8,644,288 online IoT devices with four categories (routing, monitor, printing, and industrial control). Among which, we identified 2,511,499 IoT devices using NAT for Internet access, which occupy 29.1% of the total experimental data set.

Contributions: Overall, we make the following desirable contributions:

- *First available online NAT detecting approach:* Combined five new features we introduced on the application layer and six typical features on the transport and network layer, our work is the first to conduct an automatic approach that can be used to identify Internet-wide NATs of the online IoT devices.
- *High precision and recall in the real experiment:* According to the real evaluation in the cyberspace, our novel approach present that the average precision and recall can both be up to 92%.

– *First investigation of NAT usage status which do not require cooperation:* We identified 2,511,499 Internet-wide IoT devices using NAT, which occupy one-third of the total number. Moreover, we analyzed the distribution from four dimensions: protocol, type, vendor, and geography.

The remainder of the paper is structured as follows. In Sect. 2, we discuss the related works. In Sect. 3 and 4, we present the features and detailed approach, followed by the experimental results in Sect. 5. We discuss the future work and conclude the paper in Sect. 6.

2 Related Work

Previous studies on of detecting NAT devices or identify behavior mainly focused on passive measurement [1,2,8,10,12–14,17]. Features (*e.g.*, IP ID [2], TTL [3], the HTTP user-agent strings [14], *etc.*) for passive NAT detection can be easily obtain from the collecting traffic traces of the internal network. Thus, machine learning (ML) algorithm for passive NAT detection has not been complex. Support Vector Machine (SVM), *C4.5* and Naive Bayes has been commonly used as the classifiers [1,13]. Khatouni *et al.* comprehensively employed ten kinds of ML-based classifiers to achieve higher accuracy [10]. Sun *et al.* proposed a novel density-based clustering algorithm (*DBSCAN*) and proved its efficiency [17].

However, these aforementioned ML algorithms made unavailable for active measurement. On one hand, it is hard to receive the traditional features from active probe data from the uncoordinated network. On the other hand, the target scale of active NAT detection is relatively larger than passive measurement. Murakami *et al.* and Ishikawa *et al.* both proposed the limited approach in their specific applications [9,14]. In summary, in order to actively identify the NAT-like IoT devices in the cyberspace, new features and method need be employed.

3 Preliminary Knowledge

3.1 Features of Network Layer

All packets traveling over the network layer contain the IP header with twenty fixed bytes. Among which, some classic fields are used as features for fingerprinting OS, device types [19], brands and models [18]. In order to actively identify NAT devices based on the <`request, response`> packets, we choose three features on the network layer as follows: *Time to Live (ttl)*, *IP identification (id)* and *ICMP type and code IDs (icmp)*.

The initial value TTL_{init} is set to range from 32 to 255, which is distinct from the diverse device type or OS. For instance, the TTL_{init} of `D-Link IP Camera` and `ZyXEL wireless router` are both 64 because they are using the same embedded Linux OS, and the TTL_{init} of `Siemens PLC` is 30 with its unique OS. Moreover, the TTL value is decremented by one as the IP datagram packet is transported at every hop. That means if the online device is directly connected to the Internet, the value of TTL is $TTL_{init} - 1$. For instance, Table 1 shows the TTL value of `Dell Printer` is 254, which means it can be concluded a non-NAT device. However, there are still some limitations of the TTL as a feature [8].

IP ID field also is useful but limited as a feature of NAT detection because it can be modified by a gateway. The unreachable message (Type: 3, Code: 3) or host unreachable message (Type: 3, Code: 1) can be used as a feature vector for NAT detection. And the feature has been proved effective for Internet-scale scan on unreachable hosts [16].

Table 1. Ten features of three layer among IoT devices.

Feature	Example 1	Example 2	Example 3	Example 4
ttl	60	47	254	30
id	61275	61930	0	730
icmp	{3, 1}	{0, 0}	{3 ,1}	{3 ,3}
sn & an	-	-	-	-
ws	4128	5840	2920	2048
protocol	HTTP	RTSP	PJL Raw	Ethernet/IP
port	80	554	9100	44818
banner	Non-null	Non-null	Non-null	Non-null
category	Routing	Monitor	Printing	Industial
device-type	Router	IPcam	Printer	PLC
vendor	ZyXEL	D-Link	Dell	Siemens
product	P-330W	DCS-930	b2360dn	S7-200

3.2 Features of Transport Layer

To ensure monotonically-increasing, a NAT may rewrite the sequence number, acknowledgement number, and window size of TCP packets being forwarded into the internal realm. *TCP sequence number (sn)*, *TCP acknowledgment number (an)* and *TCP window size (ws)*. As shown in Table 1, the window size value of `D-Link IP camera` is 5840 and the different value of `Dell Printer` is 2920. But only a part of features of transport layer can be extracted due to the inconspicuous differences between various device types. Thus, *sn, an* and *ws* can not be independently used for NAT detection.

3.3 Features of Application Layer

Researchers proposed Internet-wide fingerprinting device type, vendor, and product based on the banners of application protocols of IoT devices such as firewalls, SCADA and webcam [7,19], which improved the basic conditions for fingerprinting NAT devices on the Internet. Thus, we employ the following features of banners of eight general protocols, and ten particular protocols which including two monitor protocols (MP), three printing protocols (PP) and five industrial control protocols (ICP):

- *Diverse protocol categories:* Once more than two different types of particular protocols are opened on one host, we can determine that these various protocol services sharing one IP address by NAT device. For instance, Modbus and RTSP respectively belong to ICP and MP, which can not be simultaneously implemented on one network device. In this case, this feature is not suitable between general protocols and particular protocols because most IoT device products support multiple general protocols.
- *Standard service on well-known ports:* Some well-known ports have the official numbers for their corresponding protocol services that are assigned by IANA. However, there are still quite a number of online hosts opening their services on these well-known ports. For instance, there 660 K hosts that open HTTP services on port 23 (the official port for Telnet) on the Internet according to our statistics. That is one kind of typical NAT behavior that is supported by port mapping. Similarly, a part of unofficial ports are generally designed for designative services by the IoT Manufactures.
- *Multiple standard services on unknown ports:* Part of IoT devices allow the users to open various services on any ports with one IP address by port forwarding. For instance, Network Video Recorder (NVR) can be configured to simultaneously open RTSP services for several IP cameras on 515 (designated for a Line Printer Daemon (LPD) official port) or 1515 of an unknown port.
- *NAT Devices:* Many IoT devices have all series or parts of products offering native NAT functions, including wireless routers, Digital Video Recorder(DVR), etc. Their dimensional features can be learnt by the neural network.
- *Multiple different device types, vendors or products:* Generally, an independent device connects to the Internet and exposes its only TVP (type, vendor, product). Hence, if there are two or more device types, vendors or products are identified simultaneously on one IP address, it can be detected as a NAT device sharing the IP by multiple devices. Undoubtedly, this feature depends on the high accuracy of TVP fingerprinting identification.

Although features on the application layer behave more obvious than other layers, there is still a key limitation: features on the application layer mainly depend on multiple dimensional fields on two ports or more. However, these are about two-thirds of online devices in our experimental data set, that have the only open one port on the Internet, features on the application layer can not be effective in the single banner. Hence, we still need to employ other features on the network or transport layer to detect these single-port devices are whether NAT or not.

4 Methodology

In this section, we introduce our architecture and the deep learning algorithm for detecting NAT devices base on Tri-net.

4.1 Overview

As illustrated in Fig. 1, the workflow of our approach has three steps: pre-labeling, label learning, and detection. After labeled a part of instances, the training data set can be extracted features on three layers. Along with the label learning process, training data set can be enlarged and the modules of Tri-net can be trained for NAT detection as the last step. First of all, the features which are depicted in detail as follows.

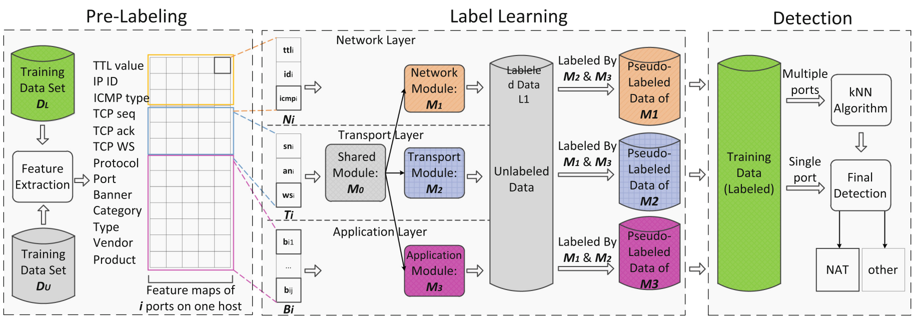

Fig. 1. The main system architecture.

4.2 Pre-labeling

As the first step of our approach, we need to conduct a part of the original data set as the training data $D_L = \{(H_{li}, y_l)|l = 1, 2, ..., L\}$ with L labeled instances in advance.

Features Extraction. An active IoT device H_{li} $(l = 1, 2, \ldots, L)$ may open i $(1 \leq i)$ ports on the Internet, we define a group of feature values on one port of H_{li} as an instance. Among which, the values of the referring features are divided into $N_i = [ttl_i, id_i, icmp_i]$ on network layer, $T_i = [sn_i, an_i, ws_i]$ on transport layer, and $B_i = [b_{i1}, b_{i2}, b_{i3}, ..., b_{ij}]$ where $b_{ij}(1 \leq j \leq 7)$ on application layer.

And y_i is defined for the detection result where $y_i = 1$ if the current host is connecting on the Internet via NAT, otherwise $y_l = 0$ means it is directly connecting on the Internet via the real public IP address or unable to determine. Based on the limit and effective features N_i, T_i or B_i on the three layers that we have discussed in Sect. 3, we can respectively label three small data set D_N, D_T, and D_B. Then the data set $D_L = D_N \bigcup D_T \bigcup D_B$ can be the input of the

NAT detection training neural network which automatically learns the respective features from each other layers to label the unlabeled instances. We denote the remaining unlabeled data set $D_U = ((H_{ui})|u = 1, 2, ..., U$ with U unlabeled online IoT devices.

4.3 Label Learning

In the second step, the unlabeled instances in D_U need be automatically labeled as NAT or *other* and added into D_L by synchronized training Tri-net through learning features.

Modules. Considering the faults of traditional NAT detecting algorithm and the weak crossing of the features on three independent layers, we employ the Tri-net as our semi-supervised approach [4]. Since the feature map has been allocated as the input of a shared initial modules M_0 to generate three diverse modules M_1, M_2, and M_3 for each layer, we can simultaneously train M_0, M_1, M_2, and M_3 modules. Then, we explore to use Tri-net for learning their particular features and training these modules.

Each module is an independent multi-layer feedforward neural network that can be trained to classify a device with one port. The input layer receives a vector (x_1 to x_i) and the values of i feature where x belongs to N_i, T_i or B_i on the three layers. And the output layer refers to the classified result where 1 denotes NAT device and 0 denotes *other* (non-NAT or undetermined). With regard to the number of the hidden layers, we chose different optimum nodes in the i hidden layers according to more than twice the largest number of neurons in the input layer. Then, the neural network can be initialized for simultaneously training each module.

With the training process in multiple rounds with the increasing labeled data set, three modules M_1, M_2, and M_3 become more and more similar. Considering the logical regression two-category classifier of our neural network, we define the probability $p_j \in [0, 1]$ calculating function of the three modules M_v $(v = 1, 2, 3)$ as follows:

$$p_j = \frac{e^{z_j}}{\sum_{k=1}^{2} e^{z_k}}, \tag{1}$$

where e^{z_j} is an exponent of network output for category j. And the cross-entropy loss function $Loss$ is:

$$Loss = \frac{1}{L} \sum_{1}^{L} \sum_{i=1}^{2} y_j \cdot \log(p_j), \tag{2}$$

where L denotes the total number of samples in training dataset D_L and $y_j \in \{0, 1\}$ denotes the labeling result.

Pseudo-label Editing. The three modules M_v $(v = 1, 2, 3)$ may generate different labeling results of one instance H_{li} based on their local features. For solving the divergence, we introduced the $pseudo-label$ strategy to denote the intermediate results which are produced by the independent modules. If one instance x is simultaneously predicted as the same pseudo-labels of the other two modules, that the same result of x will be voted as the final label. If x is differently predicted as NAT and *other* by the other two modules, it will not be labeled for continuous training in the next ground. For one instance x in M_1, if it has been both determined as a NAT device by M_2 and M_3, that x will be as NAT by our algorithm. Then the new labeled instance will be added into the training data set for the renewed training of M_1. With regard to the unseen instance M_1 which has been denoted as *other* by both M_2 and M_3 after three rounds, it will be dropped from the training data set to avoid degenerating the performance.

For determining the final label for H_i with its single port with three diverse pseudo-labels $M_v(M_0(H_i))$ and posterior probability p_j, we employ the Maximum posterior probability (MAP) as follows:

$$\begin{aligned} y_j = arg \max_{y_j \in \{0,1\}} \{ & p(M_1(M_0(H_i)) = y_j | H_i) \\ & + p(M_2(M_0(H_i)) = y_j | H_i) + p(M_3(M_0(H_i)) = y_j | H_i)\} \end{aligned} \tag{3}$$

4.4 Detection

After the training process, we got the steady modules M_v $(v = 1, 2, 3)$ and the stable labeled training dataset D_L. With regard to the host H_l in D_L may open $i(1 \leq i \leq 65{,}535)$ ports on the Internet. That means i multiple diverse labels can be produced in the training process. Hence, we employ k-Nearest Neighbors (kNN) to identify the final result infers the brand and model of a host H_l. In detail, we first calculate the distance between H_{li} its labels y_{ij} on each port every port H_{lj}. The final NAT detection result is determined by the closest distance in the trained dataset D_L.

5 Experiments and Result

In this section, we implemented the experiment to perform the actual validity and accuracy of our approach with real data on the Internet.

5.1 Experimental Data

Original Data Set. Considering the adversarial behavior of active port scan on the Internet and ethical guidelines which we must be followed [5], we employed three open data sets [6,15,16]. While there are no existing NAT labels in the three original data set. Thus, we need to label parts of the hosts as NAT or not by our pre-labeling algorithm and the above-mentioned feature fields.

With regard to the banners in three data set, device type, vendor and product can be identified by using the fingerprinting method [7,19]. After data processing and device fingerprinting, we take advantage of 7,197,713 public IP addresses with the identified IoT devices as our experimental data set. Due to multiple devices may share the same IP by NAT or a device may simultaneously open multiple ports on one IP address, we denote one port on one IP as one instance. As illustrated in Table 2, 8,644,288 instances have been classified 4 device categories, including 41 device types, 1,598 vendors, and 11,253 products. Among which, the routing or switching category devices are only identified based on the general protocol banners. And the other category devices may be identified based on the general or special protocols. For one instance, a `Dell Printer` can be identified as its device type and vendor on both HTTP and PJL Raw services.

Table 2. Experimental data set and identification result.

Category	Protocol	Port	IoTs	NATs	Type	IoTs	NATs
General	FTP	21	300,858	156,974	Router	1,193,270	238,252
	SSH	22	310,538	35,649	Gateway	502,021	312,893
	Telnet	23	468,732	84,522	UTM	431,806	33,218
	HTTP	80	3,295,851	916,489	Switch	153,539	18,647
	SNMP	161	520,363	175,276	Modem	136,479	64,864
	HTTPs	443	2,780,104	860,725	VPN	37,332	12,509
	UPnP	1900	89,453	16,978	Firewall	16,122	11,550
Monitor	RTSP	554	498,917	146,956	NVR	802,201	260,146
	ONVIF	3702	308,249	70,413	DVR	764,100	156,875
Printing	LPD	515	63,721	11,718	IPcam	763,963	172,482
	IPP	631	17,177	4,936	Printer	438,421	156,449
	PJL Raw	9100	26,233	10,560	Scanner	4,675	1,633
Industrial control	Siemens S7	102	826	411	PLC	11,742	6,754
	Modbus	502	19,730	10,945	HMI	1,799	778
	PCworx	1962	676	293	RTU	647	432
	Ethernet/IP	44818	6,720	3,243	SCADA	442	262
	BACnet	47808	9,847	5,431	DCS	364	243
Total	-	–	8,644,288	2,511,499	Other	3,385,980	1,063,512

Pre-labeling. Based on the features on the network, transport, application layer which has been discussed above, we separately labeled three labels N_1, N_2, and N_3 of the devices. Due to not all features on the three layers that can be covered for some devices, only part of the devices can be labeled, and three labels N_1, N_2 and N_3 may can not be all labeled for these devices. In essence, there also exist conflicts among three labels of N_1, N_2, and N_3 on one port or multiple ports of some hosts. This condition does not affect the training process because these labels are only pseudo-labels for the neural network model to determine the devices are NAT or not finally. After the pre-labeling process, we got a D_L labeled data set with $50k$ instances and the remaining unlabeled data set D_U with more than 8 million instances. In this case, the labeled instances only occupy the low rate of 6.2% of the total instances.

Training Process. In order to label the unlabeled instances in D_U to be labeled and added into D_L, we have programmed our aforementioned Tri-net method of the training neural network in Python. For semi-supervised training the shared initial modules M_0 to generate three diverse modules M_1, M_2, and M_3 for each layer, we fine-tuned different structures and depths for M_1, M_2, and M_3 to adjust their diverse input feature maps. Then, we denote N_4 for the synthesize labeling result which is determined by the stable pseudo-labels N_1, N_2, and N_3 of three layers.

We implemented the training program on a Lenovo server with two Intel Xeon CPU E5-2650 v4, 256 GB 2400 Mhz memory chips and two NVIDIA Tesla K80 graphics cards. It has taken about 49 h until the training process ended and the stable data set D_L with 6,318,165 instances were labeled. Eventually, 1,704,833 instances are dropped out because they can not be judged whether NAT or not.

Table 3. Experimental result of detecting NATs performance.

Category	Precision	Recall	F1 score
Routing	0.924	0.911	0.917
Monitor	0.943	0.937	0.939
Printing	0.953	0.968	0.960
Industrial control	0.967	0.944	0.955
Average	0.930	0.922	0.926

Identification. As shown in Table 2, 2,511,499 instances are determined as NATs, that account for 29.1% percent of total 8,644,288 instances. With regard to the IoT devices which only open one port on the Internet, 5,845,382 instances in our experimental data belong to the single-port class. Among which, $1,692,384$ instances are identified as NATs.

According to our statistics, 1,352,331 IP addresses were shared by more than two ports. Based on the kNN identification algorithm, 538,596 IPs are recognized as NATs. Overall, 2,130,980 IPs can be detected as NAT, which nearly rating one-third of the total 7,197,713 IPs of experimental data set.

5.2 Evaluation

Measurement. To evaluate the performance of our approach, we introduce two evaluation indexes: `precision` and `recall`. Precision reflects the rating of NAT devices correctly classified, which is calculated using Eq. (4):

$$Precision = \frac{TP}{TP \ + \ FP} \tag{4}$$

where True Positive (TP) denotes the number of NAT devices correctly classified as `NAT`, False Positive denotes the number of non-NAT devices incorrectly classified as `NAT` and False Negative (FN) reflects the number of NAT devices incorrectly classified as `Other` (i.e. non-NAT devices). On the other hand, recall reflects the number of `Other` devices incorrectly classified as NATs using Eq. (5):

$$Recall = \frac{TP}{TP \ + \ FN} \tag{5}$$

Naturally, high precision and recall are the desirable outcomes. And the harmonic means of precision and recall `F1` is calculated using Eq. (6):

$$F1 = \frac{2 * Precision * Recall}{Precision \ + \ Recall} \tag{6}$$

Verification Result. The detailed performance is revealed in Table 3. We observe that the precision of monitor devices is lower than the other categories because of the NAT functional components of themselves. Experimental results show that it is efficient and robust for detecting NATs of online IoT devices based on the features of three layers and the Tri-net method with high precision and recall rate.

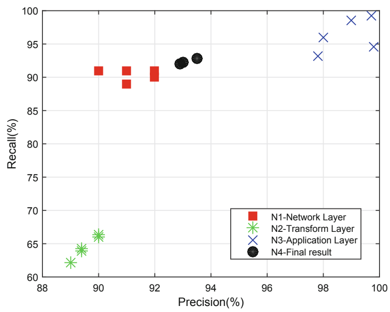

Fig. 2. The precision and recall of features on three layers.

Feature Effectiveness on Three Layers. As shown in Fig. 2, X-axis and Y-axis separately indicate the precision and recall of the diverse features on network, transport and application layers. That obviously indicates that features on the application layer have the highest precision and recall, and the transport layer has the lowest performance. The difference depends on the feature effectiveness of three layers. For example, the variation of TCP sequence and acknowledgment number between NAT and non-NAT is not obvious.

6 Conclusion

In this paper, we presented a new approach based on Tri-net for active NAT detecting with eight features on three layers. Through experimenting on real Internet open data, we found almost one-third of online IoT devices using NATs to connect to the IPv4 space. Among the four IoT categories, industrial control devices are more using NAT than routing, printing, and monitor. The final evaluation of our measurement revealed the precision and recall both can be up to 92%. With the wide deployment of IPv6, a further test would be to identify NAT on intelligent devices which are connected to Internet via IPv6.

Acknowledgments. This work was supported by National Key R&D Program of China (Grant 2017YFC0820701), National Natural Science Foundation of China (Grant U1766215, 61702504).

References

1. Abt, S., Dietz, C., Baier, H., Petrović, S.: Passive remote source NAT detection using behavior statistics derived from NetFlow. In: Doyen, G., Waldburger, M., Čeleda, P., Sperotto, A., Stiller, B. (eds.) AIMS 2013. LNCS, vol. 7943, pp. 148–159. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38998-6_18
2. Bellovin, S.M.: A technique for counting Natted hosts. In: Proceedings of ACM SIGCOMM Workshop on Internet Measurment (2002)
3. Beverly, R.: A robust classifier for passive TCP/IP fingerprinting. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 158–167. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24668-8_16
4. Chen, D., Wang, W., Gao, W., Zhou, Z.: Tri-net for semi-supervised deep learning. In: Proceedings of International Joint Conference on Artificial Intelligence (2018)
5. Dittrich, D., Kenneally, E.: The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. Technical report, U.S. Department of Homeland Security (2012)
6. Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., Halderman, J.A.: A search engine backed by Internet-wide scanning. In: Proceedings of 22nd Computer and Communications Security (2015)
7. Feng, X., Li, Q., Wang, H., Sun, L.: Acquisitional rule-based engine for discovering Internet-of-Thing devices. In: Proceedings of 27th USENIX Security Symposium (2018)
8. Gokcen, Y., Foroushani, V.A., Heywood, A.N.Z.: Can we identify NAT behavior by analyzing traffic flows. In: Proceedings of IEEE Symposium on Security and Privacy (2014)
9. Ishikawa, Y., Yamai, N., Okayama, K., Nakamura, M.: An identification method of PCs behind NAT router with proxy authentication on HTTP communication. In: Proceedings of Symposium on Applications and the Internet (2011)

10. Khatouni, A.S., Zhang, L., Aziz, K., Zincir, I., Zincirheywood, N.: Exploring NAT detection and host identification using machine learning. In: Proceedings of Conference on Network and Service Management (2019)
11. Kohno, T., Broido, A., Claffy, K.C.: Remote physical device fingerprinting. IEEE Trans. Dependable Secur. Comput. **2**(2), 93–108 (2005)
12. Komarek, T., Grill, M., Pevny, T.: Passive NAT detection using HTTP access logs. In: Proceedings of International Workshop on Information Forensics and Security (2016)
13. Li, R., Zhu, H., Xin, Y., Yang, Y., Wang, C.: Remote NAT detect algorithm based on support vector machine. In: Proceedings of Information Engineering and Computer Science (2009)
14. Maier, G., Schneider, F., Feldmann, A.: NAT usage in residential Broadband networks. In: Spring, N., Riley, G.F. (eds.) PAM 2011. LNCS, vol. 6579, pp. 32–41. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19260-9_4
15. Rapid7: Open data. https://opendata.rapid7.com/
16. Rüth, J., Zimmermann, T., Hohlfeld, O.: Hidden treasures – recycling large-scale Internet measurements to study the Internet's control plane. In: Choffnes, D., Barcellos, M. (eds.) PAM 2019. LNCS, vol. 11419, pp. 51–67. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-15986-3_4
17. Sun, W., Zhang, H., Cai, L., Yu, A., Shi, J., Jiang, J.: A novel device identification method based on passive measurement. Secur. Commun. Netw. 1–11 (2019)
18. Yan, Z., Lv, S., Zhang, Y., Zhu, H., Sun, L.: Remote fingerprinting on Internet-Wide printers based on neural network (2019)
19. Yang, K., Li, Q., Sun, L.: Towards automatic fingerprinting of IoT devices in the cyberspace. Comput. Netw. **148**, 318–327 (2019)