Hindawi
Security and Communication Networks
Volume 2021, Article ID 1828182, 13 pages
https://doi.org/10.1155/2021/1828182

Research Article

IoT-IE: An Information-Entropy-Based Approach to Traffic Anomaly Detection in Internet of Things

Yizhen Sun,[1,2] Jianjiang Yu,[3] Jianwei Tian,[1,2] Zhongwei Chen,[1,2] Weiping Wang,[3] and Shigeng Zhang[3,4]

[1]*State Grid Information & Communication Company of Hunan Electric Power Corporation, Changsha, China*
[2]*Hunan Key Laboratory for Internet of Things in Electricity, Changsha 410004, China*
[3]*School Computer Science and Engineering, Central South University, Changsha 410012, China*
[4]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China*

Correspondence should be addressed to Shigeng Zhang; sgzhang@csu.edu.cn

Received 19 October 2021; Accepted 14 December 2021; Published 30 December 2021

Academic Editor: Jinwei Wang

Security issues related to the Internet of Things (IoTs) have attracted much attention in many fields in recent years. One important problem in IoT security is to recognize the type of IoT devices, according to which different strategies can be designed to enhance the security of IoT applications. However, existing IoT device recognition approaches rarely consider traffic attacks, which might change the pattern of traffic and consequently decrease the recognition accuracy of different IoT devices. In this work, we first validate by experiments that traffic attacks indeed decrease the recognition accuracy of existing IoT device recognition approaches; then, we propose an approach called IoT-IE that combines information entropy of different traffic features to detect traffic anomaly. We then enhance the robustness of IoT device recognition by detecting and ignoring the abnormal traffic detected by our approach. Experimental evaluations show that IoT-IE can effectively detect abnormal behaviors of IoT devices in the traffic under eight different types of attacks, achieving a high accuracy value of 0.977 and a low false positive rate of 0.011. It also achieves an accuracy of 0.969 in a multiclassification experiment with 7 different types of attacks.

1. Introduction

With the popularity of the Internet of Things (IoTs) [1], the number of devices connected to the Internet has grown rapidly. According to the report released by Ericsson in June 2020 [2], it is expected that the number of cellular IoT connections will exceed 5 billion by 2025, nearly three times of that in 2020. IoT applications have penetrated into many fields such as medical, agriculture, and logistics, providing efficient collaborative work for human production, life, and home travel [3, 4]. The IoT security has aroused extensive attention in academia and industry. From the industrial IoT, to the vehicle IoT, and then to the smart-home IoT, there are a large number of security requirements in any scenarios. However, due to the use of weak keys, the security problems caused by security flaws in the design, as well as the user's weak security awareness, etc., IoT devices are vulnerable to malicious intrusion by criminals [5], leading to the disclosure of user's private data, or even the breakdown of industrial control systems, resulting in a large amount of direct and indirect economic losses.

IoT devices' identification based on traffic is an important safety measure to maintain and control device assets. It has two advantages [6]: one is introducing machine learning technology into device identification task to achieve automated identification; the other is the difficulty of obtaining traffic data is much easier than other data. You only need to install the captured API on the router to capture the two-way communication traffic between the IoT devices and the remote servers in real time. In particular, compared with controlling devices to actively send request packets to obtain the desired information, it is almost zero cost to

monitor its communication in a passive manner, and no prior knowledge is required. Existing IoT device identification methods can be roughly divided into encrypted traffic identification and unencrypted traffic identification. However, according to Garnter's report, more than 80% of the enterprise network traffic has been encrypted by 2019, and this is an irreversible trend. Thus, in recent years, many researches have begun to focus on the IoT device identification based on encrypted traffic. Since the contents of the payload in application layer cannot be extracted because of the encrypted traffic, some statistical features would inevitably be utilized under normal circumstances, including (1) packet size; (2) packet arrival time interval (IAT) [7]; and (3) frequency domain features of periodic time series traffic. These selected features and machine learning algorithm based on statistical features have achieved good results.

However, we find that the existing researches [8–10] are focused on how to improve the identification accuracy, without considering the situation of IoT devices being attacked by malware. In fact, the two tasks of IoT device identification and anomaly detection are complementary. The malicious traffic is mixed with the benign traffic, and if it is not detected and filtered, it will inevitably affect the development of the identification work. This also leads to the lack of robustness of most current identification methods. As long as a few malicious traffic is added, it will greatly reduce the identification accuracy. Our experiment found that only 10% of the malicious traffic was added to the 9 types of IoT device traffic collected in the laboratory, and the identification accuracy dropped from 99% to 75%. Thus, how to design an anomaly detection model of IoT devices based on encrypted traffic is of critical significance.

Our goal is not to detect malware [11], but to detect whether IoT devices are under attack. Existing solutions for abnormal traffic detection of IoT devices fall into two types according to the granularity of division. The detection granularity of the first solutions is accurate to each packet [12, 13]. Its disadvantage is that the identification accuracy is often very low and needs to extract a large number of attributes for each packet to form a high-dimensional feature vector. The second type of solutions detects whether the device has been attacked within a period of time [14–16]. It usually uses statistical features over a period of time to achieve anomaly detection. Although the real-time performance is not as good as the first type of solutions, the selection of features is more reasonable and the accuracy rate is higher.

To improve the detection accuracy as much as possible on the basis of ensuring real-time performance, we propose an anomaly detection model IoT-IE based on information entropy and sliding window. Obviously, it is an improvement on the second type of solutions. Information entropy is a statistic that describes the value distribution of a variable. Here its most critical feature is that the changes in the number of values that rarely occur have a greater impact on the entropy than the changes in the number of values that frequently occur; that is, values that rarely occur play a significant role in IoT-IE, which matches the phenomenon that the attribute value of abnormal packet is different from the same attribute value of normal packet exactly. Hence, we can effectively filter the window of abnormal traffic before the task of IoT devices' type identification and evaluate the retention rate of abnormal packets as well as the loss rate of normal packets through the anomaly detection model.

Contribution Summary. Our aim is to design an IoT devices' anomaly detection model based on encrypted traffic, IoT-IE. The main contributions are as follows:

(1) Aiming at the real 7 types of attacks on IoT devices, verify the impact of the traffic attack model for IoT devices on the accuracy of device identification.

(2) Comparing the normal and abnormal behavior patterns of traffic from vulnerable devices, we proposed a traffic monitoring method based on information entropy and sliding window for specific device types.

On the public IoT device malicious traffic dataset, our anomaly detection method can distinguish benign traffic and abnormal traffic with an accuracy of 97.73%, which is better than the current baseline method, and still achieves a good result in identifying the attack types.

2. Related Works

In this section, we first discuss the IoT device identification for unencrypted traffic and encrypted traffic based on whether the IoT device traffic is encrypted or not. After that, we will introduce some real-world attacks and threats against IoT devices, as well as the existing anomaly detection researches for IoT device traffic

2.1. IoT Device Identification

2.1.1. Unencrypted Traffic. The main characteristic of IoT plaintext traffic is that it can extract high-level payload, which makes a huge contribution for device type identification.

The paper [17] proposes a fingerprinting generation method for discovering IoT devices in the cyberspace. It selects TCP/IP/UDP header field values and words extracted from application layer data as features to generate fingerprinting of IoT devices. Feng et al. design an Acquisitional Rule-based Engine (ARE) [18]; it extracts the key fields in the application layer in response packets as the search query of the crawler website and utilizes the Named-Entity Recognition (NER) to extract the device labels from the selected web pages, and finally the association algorithm is utilized to generate the annotation rules for the IoT devices and to discover IoT devices in the cyberspace through these rules.

2.1.2. Encrypted Traffic. Although the payload of the traffic is an intuitive and efficient feature for device type identification, it will cause an infringement of user privacy and a higher cost in feature extraction. In addition, identification methods based on payload characteristics have become

infeasible in the research of IoT devices identification for encrypted traffic.

Radhakrishnan et al. [19] observe the heterogeneity between devices. They first propose utilizing IAT to identify IoT devices by capturing device traffic and extracting the statistical characteristics of IAT to create the unique signature of the device type; then through Artificial Neural Network (ANN), they identify the extracted device fingerprints to realize accurate classification of the device type. Aneja et al. [20] extract this feature to draw IAT diagrams for every 100 packets as well and utilized Convolutional Neural Networks (CNN) to process the generated IAT diagrams.

Msadek et al. [21] utilize dynamic segmentation technology on encrypted flow, extract relevant statistical distribution such as protocol type, packet size, and number of packets as features and then compare and evaluate five types of machine learning algorithms: KNN, Support Vector Machine, Random Forest, AdaBoost, and Extra-Tree. The work of Pinheiro et al. [22] is similar, but they only utilize three characteristics of the mean and standard deviation of packet length generated by IoT devices in one second, and the total number of bytes sent in this second. However, it is only for the known device identification, the method seems powerless for the new IoT devices that is increasing in number at this stage.

2.2. IoT Device Attack and Anomaly Detection. The huge heterogeneity and scale of IoT devices brings severe challenges to device assets management and security protection [23, 24]. At the end of September 2016, the website KrebsOnSecurity.com was hit by a large-scale DDoS attack launched by Mirai. Mirai malware scans services such as Telnet on the network to spread and then uploads its own binary files on the device through the load service to realize infection. The infected devices continue to scan for other vulnerable devices. Finally, the intruder sends control instructions through the C&C server to attack the target [25, 26].

In order to deal with the increasingly severe IoT security issues, especially large-scale DDoS attacks, some related work has been carried out in recent years. In [27], Nguyen et al. implemented an autonomous and self-learning distributed IoT device detection system. They built a device-specific normal behavior model and through the GRU neural network model to detect the deviation of benign flow and malicious flow then isolated Infected devices. For solving the detection of unknown suspicious activities or zero-day attacks, the paper [16] designs a two-stage anomaly detection method based on machine learning. In the first stage, a supervised ML algorithm is used to identify known malicious behaviors. In the second stage, an unsupervised ML algorithm such as clustering is used to identify unknown malicious behaviors or zero-day attacks. This has achieved good results in detecting a wide range of IoT attacks.

Anthi et al. design a supervised IoT device anomaly detection model [12]. They extract the header field value of OSI each layer for each data packet, including a total of 121 features such as packet length, flag, port, and window size, test and evaluate the detection performance of Naive Bayes, J48, Logistic Regression, Random Forest, SVM, and Fully Connected Neural Network for scanning attacks, DoS attacks, MIMT attacks, replay attacks, and spoofing attacks against IoT devices. However, the disadvantage of this method is that it utilizes a large number of redundant features. On this basis, KS test and Pearson's correlation coefficient are utilized for dimensionality reduction in [13]; only 28 features are utilized to achieve high-precision detection of these attacks finally.

Yair et al. propose a network-based IoT anomaly detection method N-BaIoT [15], which extracts the statistical characteristics of packet length, IAT, and packet number of IoT benign flow in five time windows (the latest 100 ms, 500 ms, 1.5 s, 10 s, and 1 min), and train deep autoencoders (one for each device) to characterize the benign behavior of IoT devices. If the autoencoder is trained on benign samples only, it will successfully reconstruct the normal observations, and when a major reconstruction error is detected, it classifies the given observation as abnormal and finally has achieved good results in terms of accuracy and time. Wan et al. propose an anomaly detection method based on the minimum description length (MDL) principle [28]. They extract features such as flow duration, source and destination IP address, source and destination ports, protocol type, number of packets, number of bytes, and compressing and encoding and then take the encoded length as the abnormal score of the traffic to be measured. Finally, the score threshold is set to detect malicious traffic.

3. Preliminary Work

For a general IoT device-type identification system, if an intruder attacks various devices connected to the network before the IoT device traffic is entered into the identification system, it will inevitably lead to the destruction of the inherent pattern of IoT device traffic, consequently affecting the identification system ability to determine the type of the device under attack. The purpose of this section is to explore the impact of the type and volume of traffic attacks on identification performance.

3.1. Attack Models. IoT devices are mainly subject to two types of attacks. One is a port scanning attack, whose purpose is to discover the open ports of the device to achieve intrusion and then control the IoT devices; the other is a denial-of-service attack, which is mainly to cut off the communication between the device and the remote server and invalidate the functions provided.

Since our IoT device-type identification system and anomaly detection model is based on encrypted traffic, we refer to the threat model mentioned in [14] and consider the following 6 different types of attacks. These attacks do not require the intruder to have a wealth of prior knowledge, only a basic understanding of the attack command.

In order to enrich the abnormal traffic dataset, we consider different reflection/DDoS attacks and ensure the normal operation of the devices during the device identification tasks, we reduce the rate of attack traffic, aiming to

change the traffic pattern of the devices and make the devices classified incorrectly. See Section 4.2.1 for details on how to attack and the process of collecting benign and abnormal traffic.

(1) *ArpSpoof*. ARP is a protocol that converts IP addresses into physical addresses. ARP spoofing refers to the fact that the intruder sends a forged ARP reply message to the device so that it utilizes wrong information to update the ARP cache table, resulting in communication errors [29].

(2) *Ping of Death*. Ping of Death is a common denial of service attack. When the IP packet length exceeds the maximum size of the Ethernet frame, the packet will be fragmented and sent as multiple frames. The receiving end can only reassemble after receiving all the fragments. Normally, the reassembled IP packet will not exceed the specified maximum size. Ping of Death is to send a large number of IP packets that exceeding the maximum size, and the extra data in the packet will be written into other normal areas, which will cause buffer overflow.

(3) *TCP SYN*. TCP SYN flooding mainly occurs at the fourth layer of OSI. The principle of the attack is that after the intruder forges the connection request from the sender and the receiver returns the response packet of the first handshake, it cannot receive the feedback of the third handshake from the sender and finally consumes the server's memory and causes to crash.

(4) *SNMP Reflection*. SNMP is mainly used to manage the devices in the local area network and can obtain the basic configuration information and status information of the devices. The intruder forges the SNMP request packets of the source IP address, and the reflection servers send a large number of response packets to the victim devices after receiving them, which will cause network congestion [30].

(5) *SSDP Reflection*. SSDP is commonly applied to Universal Plug and Play (UPnP) devices for device discovery. In view of the limitation of SSDP is that it does not check whether the querier is in the same network as the device, similar to SNMP, it is also a UDP protocol that can easily be used for reflection attacks.

(6) *Smurf*. The ICMP Echo request packet is utilized to diagnose the network. The device will respond with an ICMP Echo Reply to the source address of the message after receiving it. Once the source address is set to the broadcast address, all devices on the local network must process these broadcast messages. A large number of Reply broadcast messages will cause the network to flood [31].

3.2. Traffic Attacks Decrease Device Identification Accuracy. After reading a large number of literatures on fingerprint identification schemes that are vulnerable to traffic attacks, it is found that packet size and IAT are the two most common features they used. We adopt the strategy of making things simple, utilizing the above two features and several general machine learning algorithms to identify, and consequently explore the impact of the traffic attack model for IoT devices on the performance of the conventional device identification method.

For a mixed traffic dataset containing benign traffic and malicious traffic, when extracting statistical features for classification tasks, malicious attacks will change the inherent pattern of traffic sent by IoT devices; the statistical features and time series features of traffic will definitely occur changes. After that, whether the traffic is divided according to the session as the sample, or divided by a time window cutting method as the sample, will lead to a decline in the identification rate.

In one-way ARP spoofing, the victim IoT device receives many response packets; the time interval between these response packets remains stable and differs significantly from the normal packets time interval, while the packet size is controlled by the attack script, which can be random or of a specified size but always differs significantly from the normal case. We refer to several traditional machine learning methods mentioned in [6, 21, 22, 32, 33]. Here, we attack devices using the ArpSpoof. Figure 1 shows that Random Forest has the best performance in identification accuracy on benign test dataset and identification robustness under attack, which is consistent with the conclusions of [22, 34]. Therefore, the subsequent verification experiments choose Random Forest as the machine learning identification model baseline.

Then, we consider all the attack types for IoT devices mentioned in the previous subsection and take the proportion of the malicious traffic to the total traffic as an independent variable to explore its impact on the identification accuracy of IoT devices.

As shown in Figure 2, it is obvious that as the proportion of the malicious traffic increases, that is, the longer the attack takes, the more obvious the decline in the identification accuracy of IoT devices. At the same time, we find that the top 10% of malicious traffic volume will lead to a sharp decline in the identification accuracy, and then the degree of decline is diminished. In other words, the intruder only needs to inject a small amount of malicious traffic before the device traffic is sent to the identification model to greatly reduce the classification accuracy of the model.

In addition, the type of attack seems to have nothing to do with the decline in the classification accuracy. The reason is that no matter what kind of attack, their traffic patterns are different from the normal traffic. This deviation is the main reason for the decline in classification accuracy.

3.3. Some Observations on IoT Traffic Patterns. Compared with traditional connected devices (PCs and mobile phones), IoT devices are simple in structure. At the beginning of their design, they usually only make use of running a single task or perform a single function. Thus, their traffic always shows a repeated communication mode and generates the same volume of data periodically. Figure 3 shows the distribution

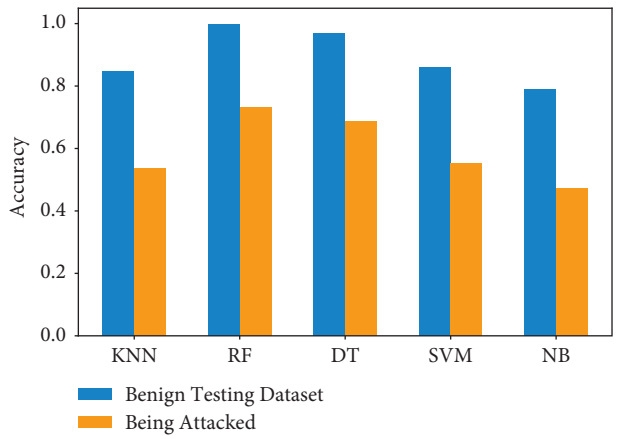

Figure 1: The impact of traffic attacks on the identification accuracy.

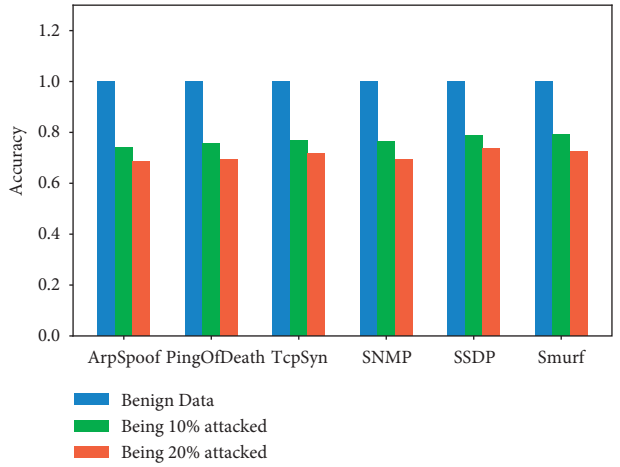

Figure 2: The impact of attack type and volume on identification accuracy.

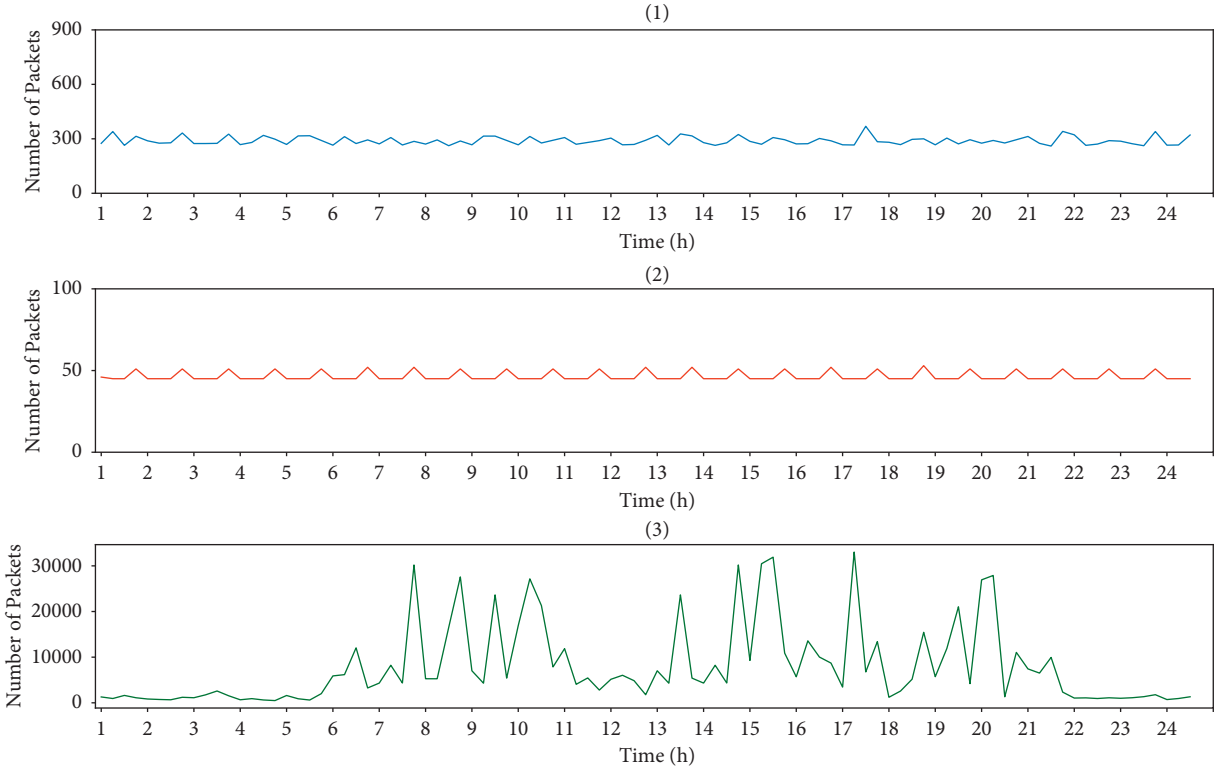

Figure 3: IoT/non-IoT device communication volume for one day. (1) Amazon Echo. (2) iHome Smart Plug. (3) Laptop.

of communication volume for two IoT devices and one non-IoT device for one day. Among them, the IoT devices are Amazon Echo and iHome Smart Plug, and the non-IoT device is laptop. It is obvious from the figure that the communication volume of non-IoT device is more irregular than that of IoT devices. This is because all the traffic of laptop comes from the usage records of users, and its communication volume conforms to people's routines. The communication volume in the daytime is much higher than at night.

The packets sending rate of these two IoT devices are in a stable state throughout the whole process for one day, and the number of packets sent within the same size time window tends to be close; here, we assume that IoT devices are in an idle state. And for different types of IoT devices, their packets' sending rates are different. The number of packets sent by Amazon Echo per hour is between 270 and 330, with a fluctuation of about 10%, while the number of packets sent per hour by iHome is less than 60 basically, and it sends 45 packets or 51 packets in most windows, which is related to the specific functions they execute. The functions of smart speakers are far richer than those of smart plugs, and the number of protocols used by smart speakers is also greater than that of smart plugs. So, the average packets sending rate of IoT devices can be used as an important feature for device type identification.

It is known that different destination ports of IoT devices traffic correspond to different protocols/services. Thus, in order to further mine the communication patterns of IoT devices, we continue to find the potential regulations of IoT devices traffic after classifying traffic according to destination port/service type. Figure 4 shows the corresponding data flow after Amazon Echo classified by the most frequently appearing destination ports; the destination ports shown in the figure account for 83% of the total traffic for one day. It can be seen from the figure that the packets' sending frequency belonging to the same service has obvious periodicity: 12 http packets from the device are sent out every 300 seconds, https packets are sent out every 30 seconds basically, and the packets to destination port 33434 is sent every 27 seconds, which is the same as the packets to destination port 49317. The phenomenon is caused by the unique characteristics of IoT devices; they will send packets to their respective servers periodically to remain connected. And it is predictable that the contents of these packets payload are the same basically. From the results, this phenomenon provides a basic follow for the slight statistical differences in each feature between each window after the flow is divided by the sliding window, and the constant traffic pattern can be used as the precondition of anomaly detection.

In addition to describing the IoT devices traffic from the time perspective, we also extract important and commonly used header field values from each packet as spatial information to describe the characteristics of IoT devices traffic. So as to know that the number of attribute values is limited, we draw the Sankey graph to observe. Figure 5 is a Sankey

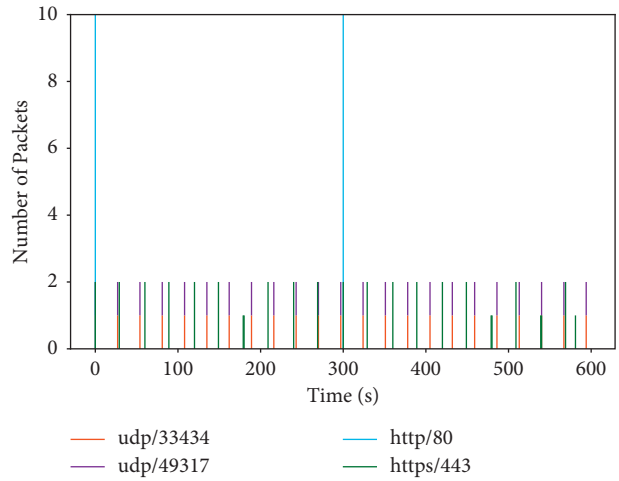

Figure 4: Periodic traffic volume of Amazon Echo over a period of time.

diagram of the Amazon Echo traffic (24,609 packets collected over a 24-hour period). The figure shows statistical information, such as the protocol used by the device, the IP address of the remote server it communicates with, and the destination port. We see that Amazon Echo not only involves the necessary 80/443 ports, but also communicates with diverse protocols, a large number of local/external server IP addresses, and different ports that provide various services. It is worth noting that the remote IP associated with port 80 is only 93.184.216.34, while port 443 has several remote IP addresses. Besides, in addition to the common high-level protocol based on UDP, Amazon Echo makes connections to ports 33434 and 49317 of the remote server for maintenance of device-specific services and points to the same remote IP.

4. IoT-IE Overview

In this section, to start with, we introduce the overview and architecture of IoT-IE. Subsequently, we discussed each module of IoT-IE, which work together to monitor IoT device communication.

4.1. System Structure. In the experimental environment we deployed, the gateway acts as a bridge between IoT devices and remote servers, and local IoT devices connect to network via WiFi or Ethernet. Thus, in order to collect IoT devices traffic more conveniently, as well as can be processed and detected in a timely manner at the same time, the IoT gateway also needs to be responsible for hosting our anomaly detection system IoT-IE.

The system architecture of IoT-IE is mainly composed of four key modules: traffic capture, IoT communication data characterization, malicious traffic detection based on information entropy, and alarm/isolation. The traffic capture module is responsible for running the traffic capture commands of IoT devices on the IoT gateway and collects the

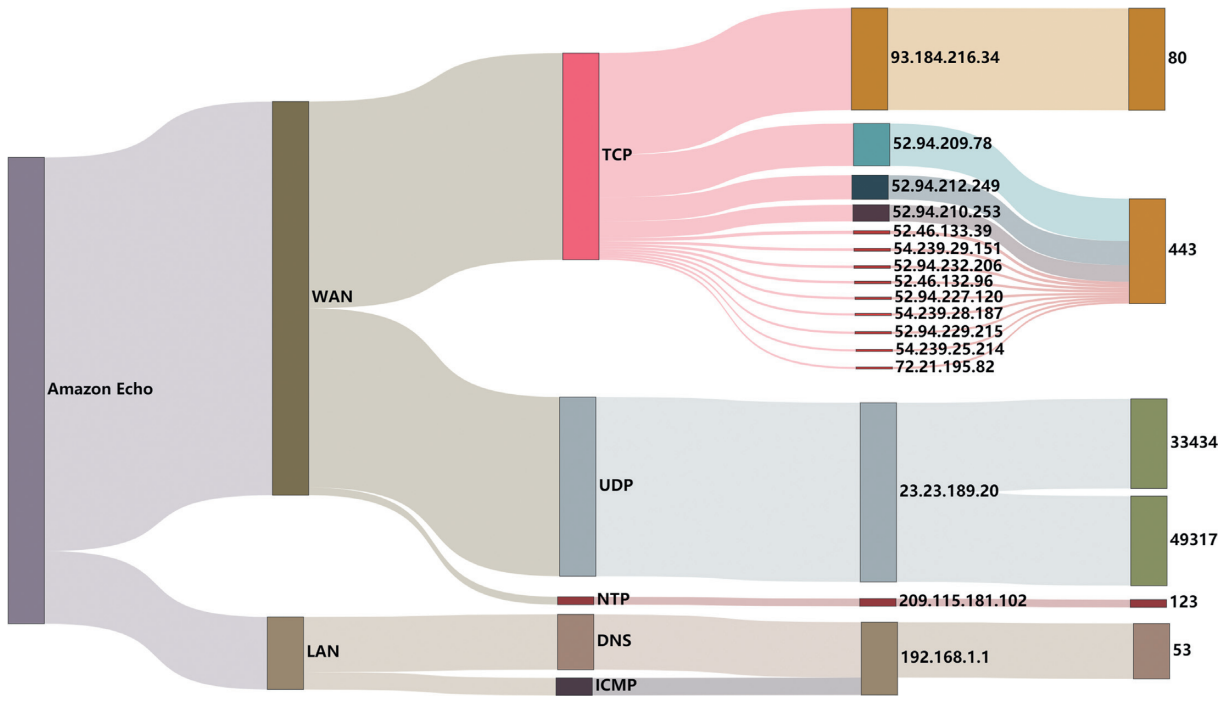

Figure 5: Sankey diagram of Amazon echo traffic. Bars from left to right represent device name, WAN or LAN, protocol, server IP address, and destination port. The protocol ICMP has no port.

traffic of the bidirectional communication between the IoT devices and the external cloud servers of IoT providers or other remote servers on the Internet. The main work of the IoT communication data characterization module is to mine the inherent traffic communication patterns of IoT devices in a normal state (without malicious attacks) and build a baseline of IoT devices' benign traffic behaviors, providing basic guidance for subsequent malicious traffic detection work. The task of the malicious traffic detection module based on information entropy is to utilize information entropy as a quantitative metric to measure the statistical differences between benign traffic and malicious traffic so that benign traffic and malicious traffic can be well distinguished, and then machine learning algorithms are used for normal/abnormal binary classification. Finally, the alarm/isolation module isolates the IoT devices that are confirmed to have been attacked so that they cannot communicate with other IoT devices in the experimental environment and prohibit communication with remote servers and alarm at the same time.

4.2. System Models

4.2.1. Traffic capture. As is known to all, the main advantage of an IoT Gateway centric security monitoring system consists in its flexibility to collect all IoT device traffic in a centralized location [16]. Our IoT device traffic collection setup in the laboratory is shown in Figure 6. Use hostapd command on a laptop with Ubuntu Linux operating system to create an IoT gateway, which serves as the access point for the WiFi or Ethernet interfaces of all IoT devices, and then use the traffic capture commands or tools such as TCPdump and Wireshark to capture the traffic data of the bidirectional communication between IoT devices and IoT provider's cloud servers via the IoT gateway. After that, we refer to the attack scripts published in [14] and use a computer under the local area network to run these scripts to attack the target IoT devices.

4.2.2. Some Observations on IoT Traffic Patterns. The IoT devices that we adopted in the experiment are all consumer IoT devices, which simulate the smart homes' IoT environment. In this module, we mine traffic patterns to prove the difference between IoT devices and traditional connected devices. The predictability of IoT device traffic patterns can make it possible to utilize machine learning to realize type identification and anomaly detection. In general, IoT devices will not be attacked immediately after it is connected to the network, so we consider analyzing the normal communication patterns of IoT devices during the period before they are attacked. This benign traffic can be used for building training set, which can be used for IoT device identification and as negative samples for anomaly detection. Nowadays, personal user privacy is getting more and more attention; the traffic of IoT devices produced by many vendors on the market is encrypted. Thus, we consider the metafeatures of IoT device traffic without any information extracted from the payload of packets, so it does not need rich prior knowledge and feature extraction cost.

Specifically, the traffic features used by IoT-IE can be packet size, IAT, number of bytes, source port, destination port, source IP address, destination IP address, protocol, flow duration, flow average rate, etc. Here, we do not utilize all features but try to explore which features may cause obvious changes after malicious attack to filter out the best

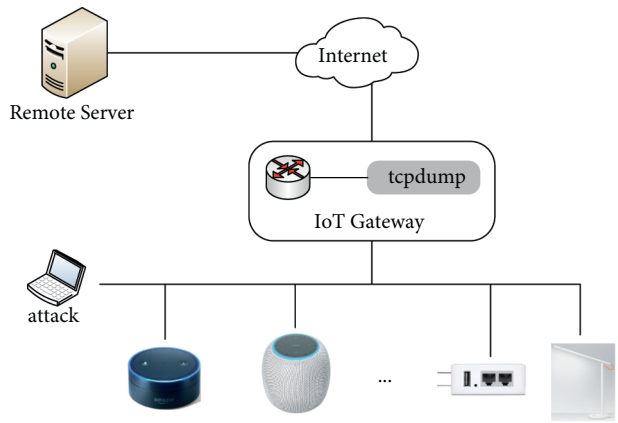

Figure 6: IoT device traffic collection platform.

features that can effectively distinguish between benign and malicious traffic. In addition, selecting appropriate features can also reduce the burden of the anomaly detection module and improve the detection efficiency.

4.2.3. Anomaly Detection Based on Information Entropy. In this module, we propose a malicious traffic detection method based on information entropy and sliding window, as shown in Figure 7. Information entropy is used as a metrics to describe the value distribution of features in a period of time. Different from human-centered Internet traffic, in general, IoT devices repeat the same operations throughout the entire capture process and generate the same amount of data regularly, and each IoT device has its own unique normal communication pattern, so the value of information entropy only fluctuates in a small range over time. However, once malicious traffic appears, some of its attributes will change in the value distribution. For instance, some values that have never appeared before will appear, which is very sensitive to the entropy; it is because the change of information entropy is more sensitive to values with a small probability of appearance than values with a large probability of appearance, resulting in a large difference between the entropy of benign traffic and that of malicious traffic. Then, the inherent feature measurement is sent to the machine learning model for training, and by judging the statistical difference between benign and malicious traffic, whether an attack occurs can be detected.

Alarm/Isolation. Once IoT-IE determines which IoT devices in the experimental environment are under attack, the module will immediately cut off the communication between the infected devices and all other devices in the experimental environment as well as remote servers to take isolation measures and notify people which devices have been attacked by means of alerts.

5. Anomaly Detection Method

Our purpose is to find suitable features or indices to detect whether there is malicious traffic in the IoT network. From the perspective of features value distribution, we can find differences between the header field values distribution of benign packets and that of malicious packets. Information entropy is a metric that describes the occurrence probability of attribute various possible values, so it can describe the values distribution of each attribute in packets well.

Entropy is a concept in thermodynamics originally, and it is utilized to measure the uncertainty of an attribute in information theory [35].

Firstly, the concept of information quantity is introduced as a measure of "how much" information. The amount of information of a specific event should decrease with its occurrence probability and cannot be negative, so it can be represented by a logarithm, as shown in formula (1). Information entropy is actually the expectation of the amount of information that may be generated before the result comes out. Considering all possible values of the random variable, the expectation of the amount of information that can be brought by all possible events.

$$h(x) = -\log\, p(x), \tag{1}$$

$$H(X) = \sum_{i=1}^{n} p(x_i) \cdot h(x_i) = -\sum_{i=1}^{n} p(x_i) \log\left(p(x_i)\right). \tag{2}$$

$p(x)$ represents the probability that the attribute X takes the value x, and there are a total of n possible values for the attribute X.

It can be seen from the previous section that IoT devices have a fixed traffic pattern. If the value of a certain attribute is different from the previous pattern or completely deviates from the pattern, it will cause a huge change in the entropy value. Furthermore, a significant advantage that distinguishes entropy from other statistical features is that it can calculate string-type values, such as IP address. As long as the value distribution is stable, entropy can be effectively used. The entropy feature extraction process is shown in Algorithm 1. For a given window size M and step size T, S samples are generated by cutting flow through a sliding window. For each sample S_i, we extract the appropriate attributes and add them to the corresponding attribute lists, use these lists as input to calculate the entropy value, and finally form the entropy feature vector.

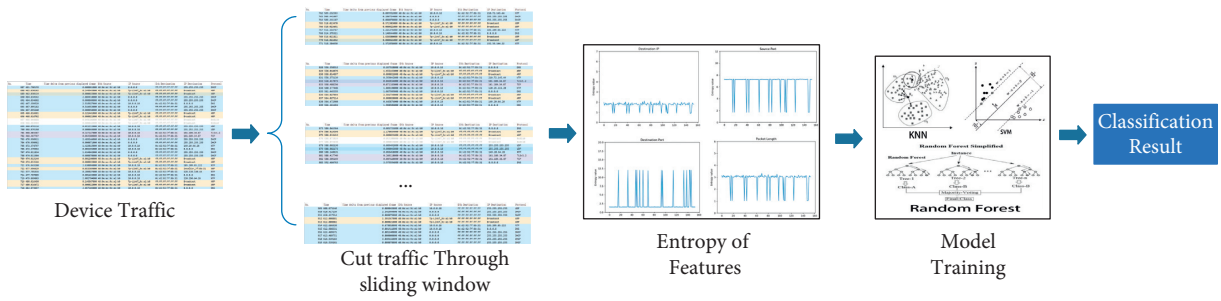

Figure 7: Anomaly detection process based on entropy and sliding window.

Input: Raw pcap file, sliding window size M set in advance, sliding step T set in advance, number of attributes to be extracted N.
Output: Feature vector

(1) *Feature vector* $\Leftarrow\varnothing$
(2) Take the first M package according to the window size, move backward T steps each time to form $F_1, F_2, F_3, \ldots, F_s$, s represents the number of samples.
(3) **for** each F_i **do**
(4) *attribute*1*_list*, . . ., *attributeN_list* $\Leftarrow$[ ], K, . . . [ ]
(5) **for** each packet P in F_i do
(6) $attribute_1, \ldots, attribute_N \Leftarrow$ Extract attributes, such as packet size, source port, destination port, destination IP, etc. in P
(7) **for** each *attribute A in attribute*$_i$ **do**
(8) *attribute*1*_list*, . . ., *attributeN_list* $\Leftarrow \cup A$
(9) **end for**
(10) **end for**
(11) **for** each *attribute_list* in *attributeI_list* **do**
(12) *Entropy* $\Leftarrow$ *F*2 (*attribute*1*_list*), which is mentioned in Section 5
(13) *Feature_vector* $\Leftarrow$ *Feature_vector* $\cup$ *Entropy*
(14) **end for**
(15) **end for**
(16) return *Feature_vector*

Algorithm 1: Entropy feature extraction.

We take part of the IoT devices in the experiment to calculate the entropy value of common features, including the normal communication traffic of IoT devices and the abnormal communication traffic represented by Smurf attacks; the benign communication traffic of IoT devices comes from Amazon Echo. The initial value of the sliding window is set to 300 seconds. As shown in Figure 8, the information entropy in the sliding window for Smurf attack is quite different from that in the same size window for benign traffic. For instance, the entropy value of the packet length basically fluctuates between 3 and 4, and the waveforms of each attribute are relatively similar, indicating that each attribute value of the IoT device corresponds to each other. The entropy values of the destination IP address, source port, destination port, and packet length under Smurf attack are much smaller than the entropy values of the corresponding feature in benign traffic. We delineate a dashed line as a threshold to isolate the benign traffic window and the malicious traffic window. Thus, we can realize malicious traffic monitoring by calculating the deviation degree between the entropy values of the malicious packets attribute field and the corresponding attribute entropy values of the benign packets.

6. Experiment

6.1. Datasets Description. We experimented on the public dataset UNSW-2018 [14]. The UNSW-2018 dataset contains benign traffic and malicious traffic of IoT devices; these two types of data are unbalanced, while the benign traffic is too much. In order to eliminate the impact of unbalanced dataset during training and testing, we take a part of benign dataset only. The authors designed two attack modes: direct attacks (e.g., ARP spoofing, TCP SYN flooding, Fraggle (UDP flooding), and Ping of Death) and reflection attacks (e.g., SNMP, SSDP, TCP SYN, and Smurf), which involve some protocols such as ARP, TCP, UDP, ICMP, DNS, and so on. In order to ensure that the devices remain functional during attack and reflect the attack traffic to the infected devices, a total of 200 attacks were launched at different rates, each attack lasting 10 minutes.

6.2. Evaluation Metrics. We consider the accuracy, precision, TPR (True Positive Rate), FPR (False Positive Rate), FNR (False Negative Rate), and $F1$-score as the evaluation metrics and adaptability measurement of the test result.

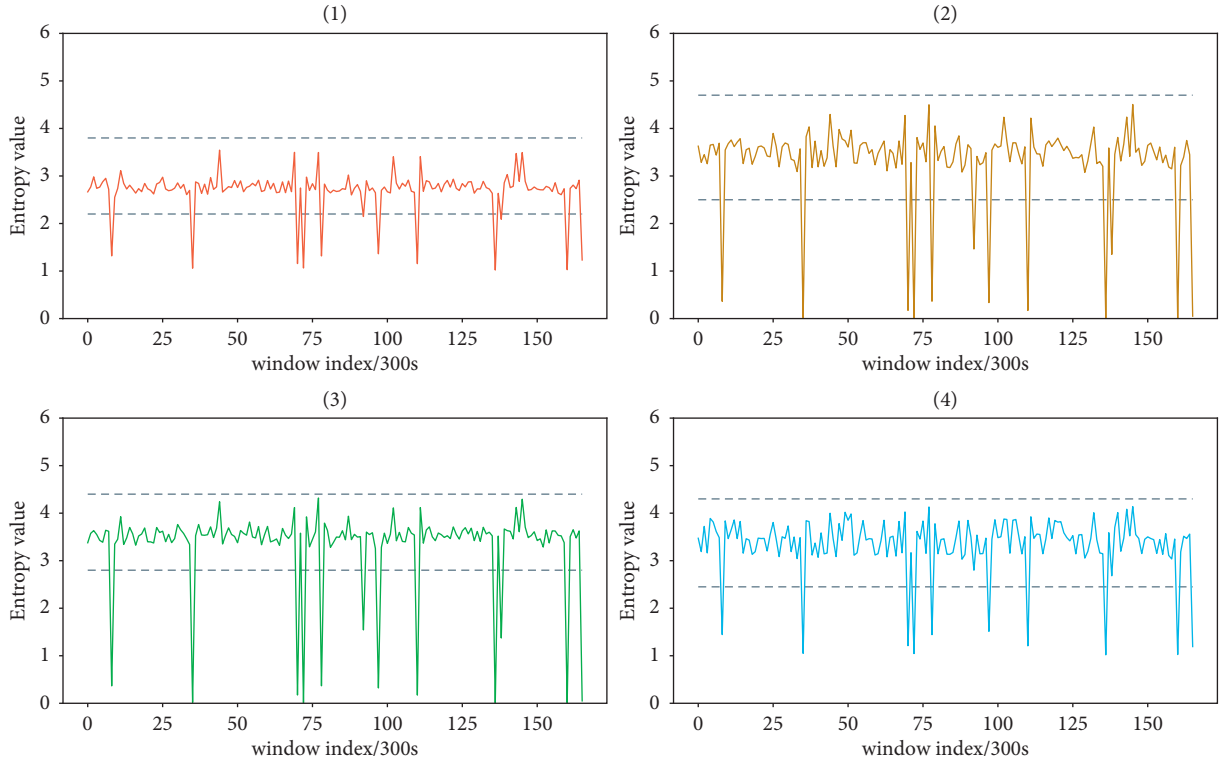

Figure 8: Comparison of the benign traffic window entropy and the Smurf window entropy from Amazon Echo in (1) destination IP, (2) source port, (3) destination port, and (4) packet length.

$$\begin{aligned} \text{Acc} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\ \text{TPR} &= \frac{\text{TP}}{\text{TP} + \text{FN}}, \\ \text{FPR} &= \frac{\text{FP}}{\text{FP} + \text{TN}}. \end{aligned} \tag{3}$$

Among them, TP represents the number of abnormal flow samples that are correctly classified as abnormal type of flow (True Positives), FN is the number of abnormal flow samples that are incorrectly classified as benign type of flow (False Negatives), and TN represents the number of benign flow samples that are correctly classified as benign type of flow (True Negatives); FP represents the number of benign flow samples that are incorrectly classified as abnormal type of flow (False Positives).

FPR measures the rate that incorrectly classified a sample of benign flow as abnormal flow, which can raise false alarms. TPR represents the percentage of abnormal flow samples correctly classified to abnormal type. Thus, the designed abnormal traffic detection system needs to maximize TPR when FPR is as low as possible, so as to prevent users from being overwhelmed by a large number of false alarms and failing to effectively perform the alarm function of the detection system. On this basis, it is also necessary to implement accurate identification of abnormal traffic in order to satisfy the basic requirements of a good abnormal traffic detection system.

7. Results

We choose IoTArgos [16] as the comparison work of our proposed anomaly detection methods because the difference between them is that the extracted features are completely different, even if they are statistical features over a period of time, IoTArgos mostly utilizes average IAT, average packet size and flow volume, etc., while we utilize entropy features, the subsequent detection algorithms are the same.

As shown in Figure 9, we set the sliding window to 200 in advance, comparing the detection metrics of the two. Compared with IoTArgos, our proposed method IoT-IE has a 1% to 2% improvement in accuracy, precision, and TPR, although the IoTArgos method has more than 95% data on various metrics already. Besides, the result of IoT-IE on FPR is much better than IoTArgos, from 0.025 to 0.007, which is essential for the normal operation of a detection system.

Secondly, we evaluated the performance of our detection method when considering various IoT devices individually, and assume that the window size is 200. The result is illustrated in Table 1 from part of devices. Different algorithms show different detection performance on various types of devices. On the whole, Naive Bayes algorithm is overall inferior to the other four algorithms, and the performance gap of the other four algorithms is very small. This shows that the entropy feature can represent the differentiator between benign traffic and malicious traffic, which has good adaptability to most machine learning algorithms.

For LiFX, all algorithms tested can approach almost 100% detection accuracy, as did for the TP-Link Plug. This is

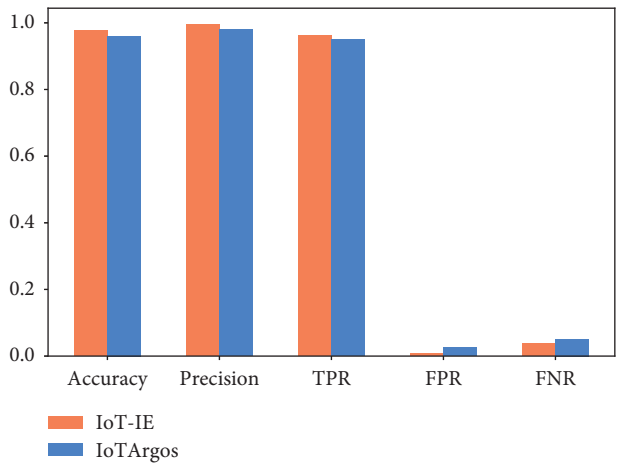

Figure 9: Performance comparison of IoT-IE and IoTArgos under 200 size window.

Table 1: Detection performance for individual devices.

Algorithm	AE			TP			NC			CU			LX		
	Acc (%)	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)	Acc (%)	TPR (%)	FPR (%)
KNN	98.5	97.0	0.0	98.9	98.7	0.4	96.8	95.3	1.6	94.6	91.0	1.7	99.6	99.2	0.0
DT	98.5	100.0	3.1	98.2	99.1	4.5	95.2	94.6	4.2	93.3	91.7	5.0	99.4	98.9	0.0
RF	99.5	99.0	0.0	99.4	99.4	0.4	96.5	95.1	2.1	94.2	92.0	3.6	99.4	100.0	1.2
SVM	100.0	100.0	0.0	99.1	99.0	0.4	97.5	96.5	1.6	94.3	91.1	2.5	99.4	99.2	0.4
NB	99.5	99.0	0.0	98.9	99.0	1.2	85.9	88.8	17.0	86.1	89.3	53.2	99.2	98.9	0.4

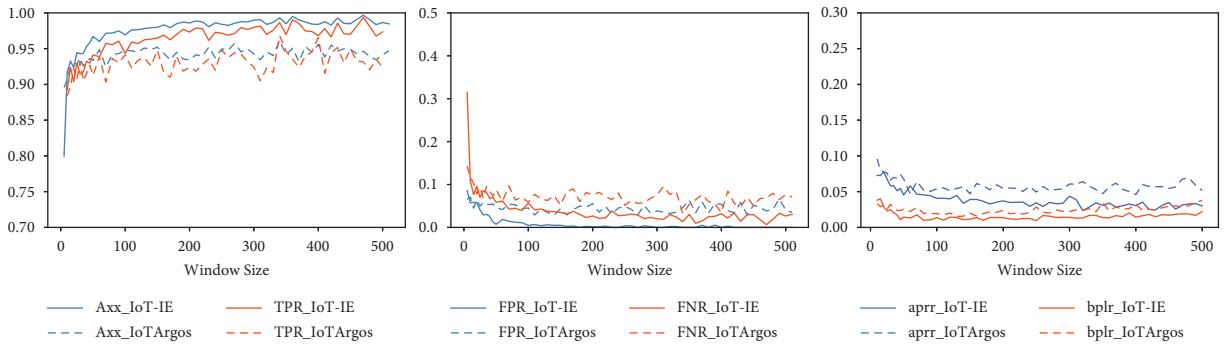

Figure 10: Detection performance varying with window size.

related to the function they have. The bulb only has two operations of light on and light off, while the plug only has power off and power on. In contrast, IoT devices such as camera or media player have more functions. As a result, the former has a much smaller change in the benign traffic pattern than the latter.

Thirdly, we explore the influence of sliding window size on detection performance. With the increase of window size, the evaluation metrics of detection performance tend to be stable quickly. When the window size exceeds 40, the accuracy can also rise to more than 95%. Moreover, the method we proposed is always better than IoTArgos obviously after curve tends to be stable.

The two metrics we first proposed are abnormal packets retention rate and benign packets loss rate. The former means the ratio of abnormal packets in windows which are misclassified as benign traffic windows, and the latter refers to the ratio of benign packets in benign traffic windows which are misclassified to be malicious traffic windows. As shown in Figure 10, its curve with the window size is closely related to the detection accuracy; when the detection is not accurate, there will be more abnormal packets retained and normal packets discarded due to misclassification. In addition, we find that the curve rises slightly when the window size is larger than 400; more packets are retained and discarded due to misclassification in a window, while the total number of packets remains unchanged, resulting in a larger proportion.

Finally, we utilize entropy features to classify the attack types, the classification result is 96.9%, and confusion matrix

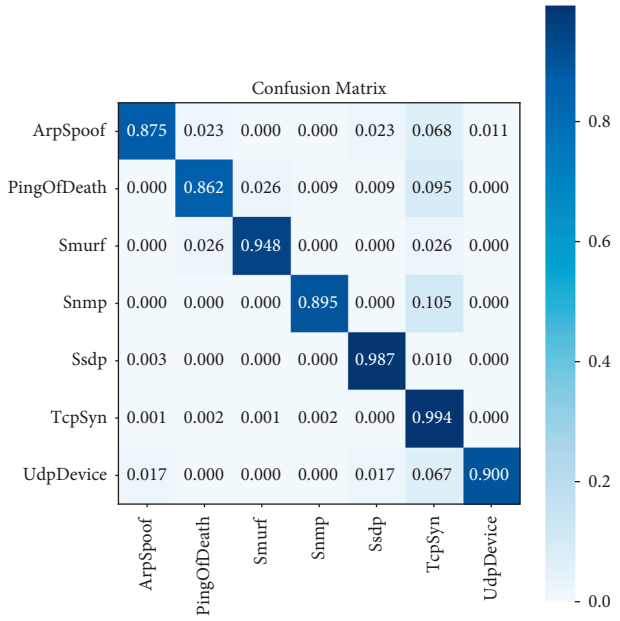

FIGURE 11: Confusion matrix of attack type classification (*x*-axis: predicted labels, *y*-axis: true labels).

is shown in Figure 11. Among them, the results of ArpSpoof and Ping of Death are not satisfactory. Through the analysis of the original malicious traffic dataset and its annotation documents, ArpSpoof will affect all features, and zero is added to the missing value in the preprocessing. While it is learned from the annotation document that TCP SYN attack in the dataset affects a specific port of a specific IP, their entropy on IP and port will be very close, which is the same for the Ping of Death.

8. Conclusion

This article introduces a method for detecting anomaly traffic of IoT devices based on information entropy. Firstly, we start from the traffic characteristics of IoT devices and compare with non-IoT devices to highlight the unity and distinguishability of IoT devices in communication patterns. Then, we propose to utilize information entropy and sliding window to detect and locate the malicious traffic of IoT devices, utilizing information entropy to describe the statistical differences of packet attributes and seeking the best classification performance by constantly changing the size of window. Experiments show that our method can still reach an accuracy of 97.73% in response to various types of IoT attacks and has good real-time performance. Even if the window size is compressed to about 40, the detection accuracy can also reach 95%.

Since our method is deployed in a smart home IoT environment currently, the focus of future work is to deploy IoT-IE in power IoT scenarios to evaluate its detection efficiency in different scenarios, mining the inherent communication patterns of IoT device traffic in different scenarios, achieving differentiation in the feature selection, and improving the robustness of anomaly detection.

Data Availability

The data used to support the findings of this study are available from the corresponding author (Shigeng Zhang) upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants nos. 61772559, 61902434, and 62172154, and the Natural Science Foundation of Hunan Province, China, under Grant no. 2019JJ50826.

References

[1] X. Liu, J. Yin, S. Zhang, B. Xiao, and B. Ou, "Time-efficient target tags information collection in large-scale RFID systems," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2891–2905, 2021.

[2] V. Hemalatha, K. A. Kumar, M. S. D. P. Gomathi, and Dr. P. Gomathi, "CAMPRO-G: an autonomous mobile robot guide for campus using IoT," *International Journal of Trend in Scientific Research and Development*, vol. 2, no. 3, pp. 896–901, 2018.

[3] X. Liu, J. Yin, L. Jia et al., "Time-efficient tag searching in large-scale RFID systems: a compact exclusive validation method," *IEEE Transactions on Mobile Computing*, vol. 21, no. 4, pp. 2891–2905, 2022.

[4] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in internet-of-things," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1250–1258, 2017.

[5] T. A. Ahanger and A. Aljumah, "Internet of things: a comprehensive study of security issues and defense mechanisms," *IEEE Access*, vol. 7, pp. 11020–11028, 2019.

[6] S. Abdalla Hamad, W. E. Zhang, Q. Z. Sheng et al., "Iot device identification via network-flow based fingerprinting and learning," in *Proceedings of the 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 103–111, IEEE, Rotorua, New Zealand, August 2019.

[7] S. Marchal, M. Miettinen, T. D. Nguyen, A.-R. Sadeghi, and N. Asokan, "AuDI: toward autonomous IoT device-type identification using periodic communication," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1402–1412, 2019.

[8] A. Hameed and A. Leivadeas, "Iot traffic multi-classification using network and statistical features in a smart environment," in *Proceedings of the 25th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–7, IEEE, Pisa, Italy, September 2020.

[9] V. Thangavelu, D. Divakaran, R. Sairam et al., "Deft: a distributed iot fingerprinting technique," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 940–952, 2018.

[10] F. Yin, Li Yang, Y. Wang, and J. Dai, "Iot etei: end-to-end iot device identification method," in *Proceedings of the IEEE Conference on Dependable and Secure Computing (DSC)*,

pp. 1–8, IEEE, Aizuwakamatsu, Fukushima, Japan, February 2021.

[11] Y. Qin, W. Wang, S. Zhang, and K. Chen, "An exploit kits detection approach based on http message graph," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3387–3400, 2021.

[12] E. Anthi, L. Williams, M. Slowinska, G. Theodorakopoulos, and P. Burnap, "A supervised intrusion detection system for smart home iot devices," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042–9053, 2019.

[13] T. Li, Z. Hong, and Li Yu, "Machine learning-based intrusion detection for iot devices in smart home," in *Proceedings of the 16th IEEE International Conference on Control & Automation (ICCA)*, pp. 277–282, IEEE, Singapore, October 2020.

[14] A. Hamza, H. Habibi Gharakheili, T. A. Benson et al., "Detecting volumetric attacks on lot devices via sdn-based monitoring of mud activity," in *Proceedings of the ACM Symposium on SDN Research (SOSR)*, pp. 36–48, IEEE, Budapest, Hungary, April 2019.

[15] M. Yair, M. Bohadana, Y. Mathov et al., "N-baiot - network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12–22, 2018.

[16] Y. Wan, K. Xu, G. Xue et al., "Iotargos: a multi-layer security monitoring system for internet-of-things in smart homes," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, pp. 874–883, IEEE, Toronto, ON, Canada, July 2020.

[17] K. Yang, Q. Li, and L. Sun, "Towards automatic fingerprinting of iot devices in the cyberspace," *Computer Networks*, vol. 148, pp. 318–327, 2019.

[18] X. Feng, Q. Li, H. Wang et al., "Acquisitional rule-based engine for discovering internet-of-things devices," in *Proceedings of the 27th USENIX Security Symposium*, pp. 327–334, USENIX, Baltimore, MD, USA, August 2018.

[19] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, "Gtid: a technique for physical device and device type fingerprinting," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 519–532, 2015.

[20] S. Aneja, N. Aneja, and Md S. Islam, "Iot device fingerprint using deep learning," in *Proceedings of the IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*, pp. 174–179, IEEE, Bali, Indonesia, November 2018.

[21] N. Msadek, R. Soua, and T. Engel, "Iot device fingerprinting: machine learning based encrypted traffic analysis," in *Proceedings of the IEEE wireless communications and networking conference (WCNC)*, pp. 1–8, IEEE, Marrakesh, Morocco, April 2019.

[22] A. J. Pinheiro, J. Bezerra, C. A. P. Burgardt, and D. R. Campelo, "Identifying iot devices and events based on packet length from encrypted traffic," *Computer Communications*, vol. 144, pp. 8–17, 2019.

[23] J. Frahim, C. Pignataro, J. Apcar et al., "Securing the internet of things: a proposed framework," in *White Paper*, CISCO, San Jose, CA, USA, 2015.

[24] T. Yu, V. Sekar, S. Srinivasan et al., "Handling a trillion (unfixable) flaws on a billion devices: rethinking network security for the internet-of-things," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, pp. 1–7, ACM, Baltimore, MD, USA, November 2015.

[25] M. Antonakakis, Tim April, M. Bailey et al., "Understanding the mirai botnet," in *Proceedings of the 26th USENIX security symposium*, pp. 1093–1110, USENIX, Vancouver, BC, Canada, August 2017.

[26] G. Kambourakis, C. Kolias, and Angelos Stavrou, "The mirai botnet and the iot zombie armies," in *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pp. 267–272, IEEE, Baltimore, MD, USA, October 2017.

[27] Thien Duc Nguyen, S. Marchal, M. Miettinen et al., "Dïot: a federated self-learning anomaly detection system for iot," in *Proceedings of the 39th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 756–767, IEEE, Dallas, TX, USA, July 2019.

[28] Y. Wan, K. Xu, F. Wang et al., "Characterizing and mining traffic patterns of iot devices in edge networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 89–101, 2020.

[29] T.-Yu Lin, J.-P. Wu, P.-H. Hung et al., "Mitigating syn flooding attack and arp spoofing in sdn data plane," in *Proceedings of the 21st Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 114–119, IEEE, Daegu, South Korea, September 2020.

[30] J.-S. Park and M.-S. Kim, "Design and implementation of an snmp-based traffic flooding attack detection system," in *Proceedings of the 11th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 380–389, Springer-Verlag, Berlin Heidelberg.

[31] N. Ugtakhbayar, D. Battulga, and Sh Sodbileg, "Classification of artificial intelligence ids for smurf attack," *International Journal of Artificial Intelligence & Applications*, vol. 3, no. 1, pp. 47–51, 2012.

[32] M. Yair, M. Bohadana, A. Shabtai et al., "Detection of unauthorized iot devices using machine learning techniques," *Machine Learning for IoT Security Analytics*, vol. 1709, p. 4647, 2017.

[33] J. Sun, K. Sun, and C. Shenefiel, "Automated iot device fingerprinting through encrypted stream classification," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer-Verlag, in *Proceedings of the International Conference on Security and Privacy in Communication Systems*, pp. 147–167, U.S. ARO grant, December 2019.

[34] R. S. Mustafizur, G. Blanc, Z. Zhang et al., "Iot devices recognition through network traffic analysis," in *Proceedings of the IEEE International Conference on Big Data*, pp. 5187–5192, IEEE, Seattle, WA, USA, December 2018.

[35] F. Zhou, W. Huang, Y. Zhao, Y. Shi, X. Liang, and X. Fan, "Entvis: a visual analytic tool for entropy-based network traffic anomaly detection," *IEEE computer graphics and applications*, vol. 35, no. 6, pp. 42–50, 2015.