Hindawi
Security and Communication Networks
Volume 2022, Article ID 8486080, 10 pages
https://doi.org/10.1155/2022/8486080

Research Article

A Lightweight Flow Feature-Based IoT Device Identification Scheme

Ruizhong Du, Jingze Wang, and Shuang Li

School of Cyber Security and Computer Science, Hebei University, Bao Ding, Hebei, China

Correspondence should be addressed to Jingze Wang; wjz9921@gmail.com

Received 5 October 2021; Revised 17 December 2021; Accepted 28 December 2021; Published 15 January 2022

Academic Editor: Weiwei Liu

Internet of Things (IoT) device identification is a key step in the management of IoT devices. The devices connected to the network must be controlled by the manager. For this purpose, many schemes are proposed to identify IoT devices, especially the schemes working on the gateway. However, almost all researchers do not pay close attention to the cost. Thus, considering the gateway's limited storage and computational resources, a new lightweight IoT device identification scheme is proposed. First, the DFI (deep/dynamic flow inspection) technology is utilized to efficiently extract flow-related statistical features based on in-depth studies. Then, combined with symmetric uncertainty and correlation coefficient, we proposed a novel filter feature selection method based on NSGA-III to select effective features for IoT device identification. We evaluate our proposed method by using a real smart home IoT data set and three different ML algorithms. The experimental results showed that our proposed method is lightweight and the feature selection algorithm is also effective, only using 6 features can achieve 99.5% accuracy with a 3-minute time interval.

1. Introduction

With the popularization and development of high-speed networks, artificial intelligence, big data, and other technologies, the number of IoT (Internet of Things) devices connected to the Internet has also rapidly increased. According to Cisco's forecast, there will be 500 billion IoT devices by 2030 to access the Internet [1]. The mounting number of IoT devices poses threats to the network [2] and brings more challenges to network managers [3]. In Cisco's recent comprehensive report on network security [4], it was stated that an increasing number of hackers utilize the vulnerabilities of IoT devices to carry out cyberattacks. In the current Internet environment, exploiting IoT devices to implement DDoS (distributed denial of service) attacks has become a primary form of attack [5]. Therefore, learning how to manage IoT devices and ensuring the security of the IoT network system have become the issues of most concern for network managers.

Presently, there are methods to ensure the security of IoT systems by authenticating IoT devices through cryptographic approaches or deep learning [6]. However, these methods are generally costly and unsuitable for the characteristics of low energy consumption and low computing power of networked devices, which will affect the performance of IoT system's effectiveness. At the same time, the traditional anomaly detection system judges whether the device exhibits abnormal behavior by detecting the abnormality of the traffic pattern. However, the Internet of Things devices have massive and heterogeneous characteristics, and it is unmanageable to identify abnormal data behavior patterns. Therefore, identifying the types of IoT devices connected to the network is of great significance to the management of IoT devices, especially in a low cost way. In the case of limited gateway computing resources, efficiently and accurately identifying devices is a problem that needs to be urgently solved.

To better identify devices on the gateway, this study proposed a lightweight IoT device identification method based on flow features. This solution studies the flow-related statistical characteristics intensively; then to pursue less cost, a novel NSGA-III-based [7, 8] filter type feature selection algorithm is proposed; and finally, the extra random tree algorithm is used to build a device recognition model to classify devices. The features used in this paper are elaborated: first, the features are at the transport layer, so this method is suitable for all IoT devices that communicate on TCP/IP protocol stacks; second, they also do not include plaintext features, effectively avoiding the problem of feature invalidation caused by encrypted transmission and at the same time efficiently perform feature extraction, model construction, and IoT device identification; last, the proposed novel feature selection method also plays an important role in reducing the cost through the device identification process.

Some of the important contributions of our present work are listed below:

(1) To solve the problem of IoT device identification in a low-overhead manner, we develop a lightweight IoT device identification scheme based on feature selection and machine learning algorithms. We also demonstrate its ability to identify IoT devices with over 99.5% accuracy with less cost than other schemes.

(2) In-depth research has been carried out on flow-related statistical characteristics and the time interval of feature extraction. DFI technology is used to build features to avoid the unavailability of plaintext features due to data encryption and improve the performance of feature extraction, model construction, and device identification.

(3) Based on NSGA-III, we introduce symmetric uncertainty and correlation coefficient and propose a novel low-overhead feature selection method to perform feature selection on the extracted flow-related statistical features in IoT device identification, and the valid features are filtered while reducing the dimensionality of the features.

(4) Experiments are conducted on a real data set. The experimental results show that the proposed feature selection method performs well and the proposed scheme can achieve higher accuracy in a short time window. Its cost is much lower than the existing method. It can also achieve the same accuracy as the actual scheme.

The remainder of this paper is arranged as follows: Section 2 demonstrates the related works. In Section 3, we explain our proposed feature selection method and the IoT device identification model. In Section 4, we exhibit the experimental results and data set. Finally, Section 5 contains the conclusion of this work.

2. Related Works

Recently, researchers have proposed a variety of solutions for identifying IoT devices. The current IoT device identification schemes can be classified into two categories from the perspective of fingerprint acquisition methods: one is the active detection method, and the other is the passive traffic analysis method. The active detection method obtains the response by sending requests to the target device and extracts the banner for device identification by analyzing the content of the response. The passive method extracts features by analyzing the daily traffic generated by the device. Feng et al. [9] proposed an active detection method for device discovery and identification, which uses the application layer response generated by the device to extract the banner and builds a fingerprint database and then establishes the map between device response and device type, vendor, and model. They achieved a very fine-grained device identification scheme, but this approach needs to send massive packets to the network, which will bring huge cost to the devices. These methods focus more on device discovery rather than management. To better manage IoT devices and offer low cost, our proposed method extract feature is passive.

Miettinen et al. [10] proposed a framework to identify the types of networked devices and restrict the communication of vulnerable ones. They used 23 features generated from the traffic packets of the IoT devices to construct fingerprints for each device. A classifier was trained for each device type to identify vulnerable devices. This method can differentiate vulnerable devices from normal devices easily, but they only detect whether the devices are normal when they are first introduced into the network. This approach is not intended for long-term device management. We resolve this problem in our method by continuously collecting the traffic devices produced. Furthermore, Marchal et al. [11] proposed AuDI, which divides the network traffic into "flows," which are several time series. They defined the flow as the traffic that uses a specific protocol to communicate with a MAC address. When a packet is sent in 1 second, it is marked as 1 in the time sequence. Then, the DFT (discrete Fourier transform) periodic features of traffic are calculated and obtained, and 33 features related to traffic cycle are used to classify the devices combined with the kNN algorithm. This novel method uses DFT to construct features, but the features have high dimension, and the DFT process also introduces much cost to the identification system. However, our method avoids these expensive calculations.

Santos et al. [12] utilized the four statistical features of traffic characteristics combined with the text information of the user agent extracted from the packet payload and the random forest algorithm to classify the devices. Le et al. [13] proposed a method for device classification based on DNS traffic. They extract the content of DNS traffic packets, using the TF-IDF algorithm for feature construction to classify and identify the type, vendor, and model of the device. Msadek et al. [14] proposed a dynamic sliding window traffic segment method, and they used DPI (deep packet inspection) technology for feature construction and a variety of machine learning algorithms for model construction and evaluation. These three methods use plaintext features for device identification. Nevertheless, for encrypt traffic, these features will be invalid. Our method avoids using plaintext features for this reason. Aksoy and Gunes [15] proposed a method using GA (genetic algorithm) to reduce the dimension of the feature vector and utilized a variety of machine learning algorithms to build a secondary classification model to classify devices in a genre-model granularity. Nonetheless, the GA algorithm and secondary classification introduce more cost into the system. Shahid et al. [16] used the size of the first n packets and $N-1$ time intervals of TCP session interaction between devices as features and various machine learning algorithms for device identification. This method is also not suitable for long-term device management. There are also some research developing device identification schemes based on signal process, like [17, 18]; their research focuses on physical layer performance of the devices, which is not our point, but as effective methods in IoT device identification, we also consider their works.

The main contributions of these studies were to construct special features associated with device type accomplishing device type identification by machine learning algorithms. The features are essential in this type of work. Sivanathan et al. [19] deeply investigated the characteristics of traffic in a flow level, and they constructed a 2-stage classifier for device classification. In the first stage, they extract DNS queries, port number, and cipher suits from these text features to obtain a class and confidence value. In the second stage, they combined the output of the first stage and flow-level statistics with random forest to classify devices. We used their method as a baseline method for comparison. Based on their work, we optimized the feature selection to reduce the IoT device identification system's cost, attaining a lightweight method with comparable identification accuracy.

3. The Proposed Device Identification Model

The system model is pictured in Figure 1. First, we take the captured traffic as input and select a fixed time interval to split the traffic; second, we generate flows from the split traffic, extract flow-level features by a statistical method, and then filter out invalid and redundancy features by the proposed feature selection method, which is based on NSGA-III; finally, a variety of machine learning algorithms and the features selected in the previous step are integrated to classify devices and multiple time intervals are selected for experimentation. The most suitable time interval and machine learning algorithm is then selected to build the efficient device classification model.

3.1. Feature Description. The purpose of this article is to build an efficient and accurate IoT device identification scheme based on flow-related statistical features for device identification. The first step for device identification is using flow statistical values to represent the behavior of IoT devices. In addition, the method in this paper selects the flow generated in a fixed time window, which prevents the problem of low efficiency of feature extraction caused by a flow of too long duration. At the same time, it was found that when the bilateral flow is used for feature extraction, the features generated by the large amount of flow data produced from the frequent mutual access of devices in the LAN will decrease classification accuracy. This is mainly because the frequent mutual access of the devices generates a large amount of the same traffic, which results in similar features. For example, the traffic between the Belkin Wemo switch and motion sensor in the data set has this problem.

Table 1 shows the result of address statistics on the pcap data of the Belkin Wemo motion sensor using Wireshark. DstIpAddress represents the destination IP address of the packets, and Count is the count of packets. 192.168.1.223 is the IP address of the Belkin Wemo switch. 64.14% of the traffic is accessing each other, which will produce a large number of similar features, leading to the deterioration of the device identification model. In view of the fact that a large number of network attacks require access to the Internet, the flow features used in this solution are all bidirectional flows when local devices interact with external network services or devices.

Flow [19] is identified by a five-tuple group: source IP address, destination IP address, source port, destination port, and protocol. The related statistical characteristics of flow are flowVolume's (the sum of bytes of two-way flow upload and download) median, mode, maximum, minimum, information entropy, mean and variance, flowRate's (flowVolume/duration of flowVolume) the same statistics as flowVolume. At the same time, the port number accessed by the device can also be used as a part of the basis for classification. To fit the machine learning algorithm, the port number-related features are processed as follows in this scheme: first, the port numbers are classified into three categories: the port numbers 0–1023 are assigned to certain services as one category, represented as port1; 1024–49151 are loosely bound to the port numbers of some services as a category, represented as port2; 49 151–65535 dynamic or private ports are in a category, and binary encoding is performed on this three categories, represented as port3. The number of occurrences of the port number is recorded, denoted as port1Cnt, port2Cnt, and port3Cnt. Moreover, the number of occurrences of flows that belong to different protocols (TCP/UDP) is recorded, denoted as (udpCnt, tcpCnt).

For ease of deployment, this solution extracts flow-related information within a fixed time window as classification features. The choice of time window will affect the effect of the solution. When the time window is short, the overhead of storing and extracting features is small. However, in a short period of time, the flow statistical characteristics of some devices show high similarity, which will lead to a decrease in the accuracy of the model; when a long-time window is selected, the storage and extraction of the features will be costly, but the flow statistical features of different devices relatively deviate from each other. Therefore, it is necessary to make a trade-off between the storage and extraction feature overhead and the classification accuracy. The gateway device is sensitive to the storage and calculation overhead, so the time window should be shortened appropriately.

3.2. Feature Selection. The purpose of feature selection is to select a valid subset of attributes and to remove irrelevant or redundant attributes. Traditional feature selection methods can be divided into three categories, namely the filter, wrapper, or embedded methods. Compared with the other

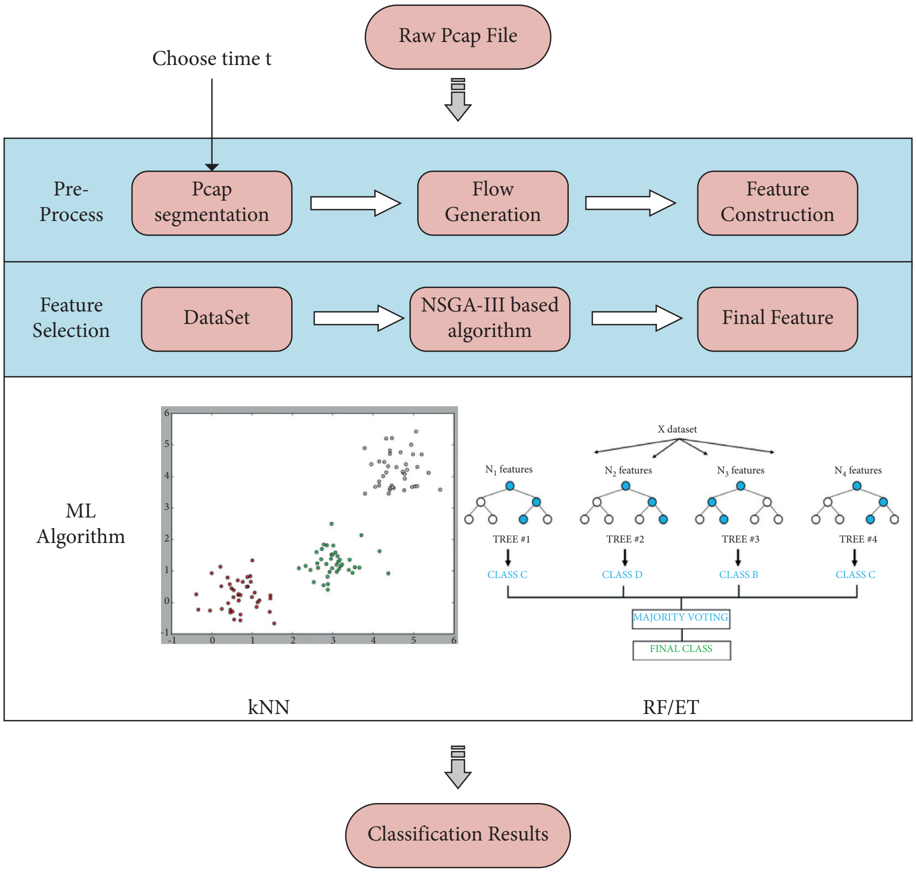

Figure 1: System model.

Table 1: Belkin Wemo motion sensor traffic statistics.

DstIp Address	Count	Percent (%)
239.255.255.250	1339	1.58
192.168.1.249	15 006	17.72
192.168.1.223	54 303	64.14
192.168.1.208	3403	4.02
192.168.1.1	8124	9.60
184.73.174.14	1494	1.76
174.129.217.97	992	1.17

two types of methods, the filter method does not require machine learning algorithm training in the feature selection process and is the least expensive method of the three. The filter method assumes that the selected optimal feature combination is a set of valid features. How to evaluate the utility of the feature is a key issue in the filter method. To better ensure the effect of selecting features, a feature selection method based on multiple objective functions using NSGA-III is proposed.

To ensure the effectiveness of features, this method models feature selection as a multiobjective optimization problem and uses NSGA-III to search for the optimal solution. There are three objective functions/evaluation functions. In the following description, F represents the set of all the features, SF represents the selected feature subset, and NSF represents the unselected feature subset, which have the following relationship:

(1) $F = SF \cup NSF$

(2) $SF \cap NSF = \varnothing$

3.2.1. Symmetric Uncertainty [20] Based Objective Function. Mutual information (MI) of two variables is a measure of the degree of interdependence between variables. The value of mutual information represents the degree to which the uncertainty of the other variable is reduced when one variable is known. Mutual information $\mathrm{MI}(X;Y)$ between two random variables X and Y is shown in equation (1).

$$\mathrm{MI}(X,Y) = \sum_{x\in X y\in Y} \sum_{p} p(x,y)\log_b \frac{p(x,y)}{p(x)p(y)}. \qquad (1)$$

The value of b is 2, $p(X)$ and $p(Y)$ are the probability density functions of X and Y, respectively, and $p(X,Y)$ is the joint probability density function of X and Y. Symmetric uncertainty is standardized mutual information, which makes the information shared between random variables comparable, and it is always used in the feature selection process. The calculation of symmetric uncertainty is exhibited by using equation (2).

$$\mathrm{SU}(X,Y) = 2.0 \times \frac{\mathrm{MI}(X;Y)}{-\left(\sum_x p(x)\log\, p(x) + \sum_y p(y)\log\, p(y)\right)}. \tag{2}$$

The value range of $\mathrm{SU}(X,Y)$ is between 0 and 1. The closer the symmetric uncertainty value is to 1, the more relevant the variables X and Y are. At this point, we obtain the first objective function, which is represented by using equation (3).

$$F_1 = \frac{\sum_{f_i,f_j\in\mathrm{SF},f_i\neq f_j}\mathrm{SU}\left(f_i,f_j\right)}{\sum_{f\in\mathrm{SF},c=\mathrm{class}}\mathrm{SU}(f,c)}, \tag{3}$$

$\mathrm{SU}(f_i,f_j)$ is the symmetric uncertainty between feature i and feature j in SF, and $\mathrm{SU}(f,c)$ is the symmetric uncertainty between feature f and class in SF. The smaller the function value, the better the classification effect of feature set SF.

3.2.2. Correlation Coefficient-Based Objective Function. Correlation coefficient is also a method used to measure the degree of correlation between variables. The difference between symmetric uncertainty and the correlation coefficient is that the latter measures the degree of correlation between variables from the perspective of statistics, while the former measures the degree of correlation from the perspective of information entropy. The calculation of the correlation coefficient is shown in equation (4).

$$\mathrm{COR}(X,Y) = \frac{\mathrm{cov}(X,Y)}{\sigma_X\sigma_Y}, \tag{4}$$

$\mathrm{cov}(X,Y)$ is the covariance of random variables X and Y, and σ_X and σ_Y are the standard deviations of X and Y, respectively. We can design the second objective function, defined as equation (5).

$$F_2 = \frac{\sum_{f_i,f_j\in\mathrm{SF},f_i\neq f_j}\mathrm{COR}\left(f_i,f_j\right)}{\sum_{f\in\mathrm{SF},c=\mathrm{class}}\mathrm{COR}(f,c)}, \tag{5}$$

f_i, f_j, c have the same meaning as equation (3). The smaller the function value, the better the classification result of feature set SF. To enable the feature selection method to achieve the purpose of dimensionality reduction, the third objective function is introduced by using equation (6).

$$F_3 = |\mathrm{SF}|. \tag{6}$$

3.2.3. NSGA-III Algorithm. The framework of the NSGA-III [7, 8] algorithm is roughly the same as the NSGA-II algorithm. The main difference lies in the individual selection mechanism of the offspring: NSGA-II selects the offspring based on the crowding distance, and NSGA-III uses the method based on reference points. NSGA-III solves insufficient algorithm convergence and diversity when multi-objective optimization problems with three or more objective functions are involved. The algorithm also makes it easier to find the optimal solution.

To optimize the proposed three objective functions (F_1, F_2, F_3), the steps of the NSGA-III algorithm are as follows:

(1) Generate an initial population that has N individuals. Individuals are a sequence of random values between 0 and 1. A value larger than 0.5 represents a selected feature, otherwise, the feature is not selected.

(2) Generate reference points set Z^* based on the three objective functions.

(3) Use evolutionary operators to generate a child population and evaluate objective values for every individual.

(4) Use nondominated sort for the combination of father and child populations.

(5) Nondominated ranking based on Pareto dominance on the combined population.

(6) Select N individuals as the next generation based on the former reference point set.

(7) Repeat (3)–(6) until it reaches the maximum iteration times to obtain the Pareto optimal solution set.

3.3. Machine Learning Algorithm. To achieve the best results, we selected three machine learning algorithms based on their descriptions in literature [21], evaluating them from the perspectives of accuracy and training speed and selecting the best performing algorithm to ensure that the method proposed in this article has a higher classification accuracy with less overhead.

The following briefly introduces the three machine learning algorithms used in the experiment:

(1) *k-Nearest Neighbor (kNN) Algorithm.* kNN is a classification algorithm with no training process. The most important parameter is k, if the input sample x is given, x will be classified into the k samples closest to x in the training set for most samples in the same category. kNN is used in the preliminary experimental verification process.

(2) *Random Forest (RF).* RF is an ensemble learning method that contains multiple CART decision trees. There have been many articles using RF to construct the IoT device identification scheme that achieved excellent results, indicating it is suitable for the device identification system.

(3) *Extremely Randomized Trees (ET).* ET is very similar to RF. The difference between this method and RF is that the selection of the node bifurcation attributes of the decision tree in ET is random, while the node division in RF of the bifurcation attribute is selected after Gini index calculation. Given its high similarity with RF, we select this algorithm as a part of the device identification system for comparison with the RF's results.

4. Data Set, Experiment Results, and Analysis

In this section, we will conduct a detailed analysis of the used data set [19] and the selected features of this scheme and use different machine learning algorithms at different time intervals to evaluate the classification results and cost. Finally, the best performing ML algorithm is given, and the model is constructed based on this algorithm.

The experimental environment is a personal computer, the detailed configuration is Intel core i5 9400 2.90 GHz, memory 8 GB, win10 64-bit operating system. The experimental steps are as follows: first, the collected data are subpackaged at fixed time intervals, and then the joy tool [19] is used to extract the flow information; second, Python script is used to calculate the relevant statistical values from the output of joy and constructs the features for storage and finally uses the machine learning algorithm provided by scikit-learn [22] to establish machine learning models and classify the devices and evaluate the classification results.

4.1. Data Set. The data set used in this article comes from the public data set of the paper [19], which is obtained by collecting the traffic of smart home devices in the laboratory under the campus network environment. The IoT devices in the data set include cameras, smart lighting tools, activity sensors, and health monitors. The TP-Link router acts as a gateway through which all devices connect to the Internet. In the data collecting progress, they connect to the router through an additional device, use tools such as tcpdump to passively collect the traffic of all devices, and save the traffic collected every day as a pcap file, which is stored in the hard disk connected to the device. This article uses opened 20-day data for experiments. Because the solution in this paper is based on the characteristics of the transport layer construction and classification, the provided data set only gives the mac address corresponding to the device, and we also analyze the IP address corresponding to the devices.

4.2. Feature Selection Results. This solution uses the filter feature selection method based on NSGA-III to remove redundant features while reducing the dimensionality of the features, that is, to reduce the computational cost of the model while ensuring the accuracy of the classification. NSGA-III is a variant of the GA algorithm. For individual construction: the number of elements contained in the individual is the same as the cardinal of the full set of features; initially, the value of each element is a random number between 0 and 1, and an element greater than 0.5 represents the feature is selected. When conducting the experiment, the number of individuals used is 40, and the number of iterations is set to 100. We performed feature selection on the 1-min time interval for small overhead introduced to the system. Figure 2 shows the results of NSGA-III operation.

As can be seen in Figure 2, the results appear to have the minimum value of three at the same time. The features selected in the Pareto front are port2 (destination port between 1024 and 49151), port2Cnt, tcpCnt, udpCnt, the mode of flowVolume, and the variance of flowVolume. The time complexity of NSGA-III is $O(N^2M)$, where N is the number of individuals in the population (40) and M is the number of objective functions (3). The feature selection process only brings less additional overhead to the system. Through our feature selection method, we select six features from the 22 features we described in Section 3.1. For our objective to be lightweight, this approach markedly reduces the classification and training overhead.

We also compared the features used in this research and the baseline method, and the features and the selection status are shown in Table 2. Our purpose is to deeply investigate the applicability of flow-related statistics and establish a lightweight IoT device identification scheme; therefore, we construct the feature set almost from the flow-related statistics because it is easy to get the flow-related statistics, which means the feature extraction progress only bring little cost to the system. The baseline method just uses the mode of flow volume and flow rate and then also forms word bag models for port, domain name, and cypher suit, and these text features are imported to a Bayes classification to generate the class and probability for final classification. From a lightweight point of view, we only use one-level classifier and remove the text features on account of text features need to be processed additionally and cause extra cost. In the selection process, the features are also selected properly to further cut the cost. At the same time, the classification performance can be maintained above a high level, and the classification details are shown in the following.

About the selected features after feature selection progress, we attempt to explain why our feature selection algorithm chooses them. First, port2 and port2Cnt represent the devices access the port between 1024 and 49151, users' customized services always run on these ports, as different devices access different services, the access times and whether access these ports should show great discrimination between devices. The variance and mode of flow volume represent the quantity of device traffic and the fluctuation of traffic, and they describe device communication behavior from the traffic view. And for the TCP and UDP flow counts, they represent the protocol discrimination between the devices, as different devices access different services, the flows always use different protocols, and these features describe the devices' behavior from the view of protocol. Combine all selected features, we can describe the device communication behavior comparatively comprehensively, and therefore, the classification results can reach a high level on accuracy.

4.3. Classification Results. In this section, we will evaluate our scheme mainly from two points of view. The first is the classification performance, which is used to measure the applicability of an IoT identification method, and to prove our scheme's lightweight characteristics, the second view is the cost of our method.

4.3.1. Classification Performance. The following will show the results of classifying the data set using the three machine learning algorithms mentioned before and the features

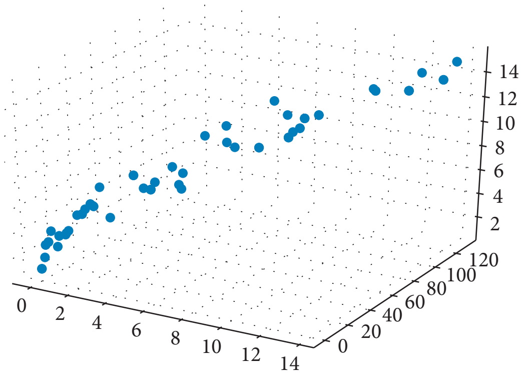

Figure 2: Pareto optimal fronts obtained by NSGA-III.

Table 2: Features used in this research.

Name	Explanation	Selection status		
		Before selection	After selection	Baseline selection
VOL_MED	Flow volume's median	✓		
VOL_MOD	Flow volume's mode	✓	✓	
VOL_MAX	Flow volume's maximum	✓		
VOL_MIN	Flow volume's minimum	✓		
VOL_IE	Flow volume's information entropy	✓		
VOL_AVG	Flow volume's average	✓		
VOL_VAR	Flow volume's variance	✓	✓	✓
RATE_MED	Flow rate's median	✓		
RATE_MOD	Flow rate's mode	✓		
RATE_MAX	Flow rate's maximum	✓		
RATE_MIN	Flow rate's minimum	✓		
RATE_IE	Flow rate's information entropy	✓		
RATE_AVG	Flow rate's average	✓		
RATE_VAR	Flow rate's variance	✓		✓
PORT1	Whether the flow access port between 0 and 1023 appeared	✓		
PORT2	Whether the flow access port between 1024 and 49591 appeared	✓	✓	
PORT3	Whether the flow access port between 49592 and 65535 appeared	✓		
PORT1_CNT	The count of remote IP port between 0 and 1023	✓	✓	
PORT2_CNT	The count of remote IP port between 1024 and 49591	✓		
PORT3_CNT	The count of remote IP port between 49592 and 65535	✓		
UDP_CNT	The count of flows use UDP	✓	✓	
TCP_CNT	The count of flows use TCP	✓	✓	
DUR_MOD	Flow duration's mode			✓
SLP_TIME	Time intervals' mode between flows			✓
DNS_INT	DNS intervals' mode			✓
BAG_PORT_NUM	Word bag model of port which flow accessed			✓
BAG_DOMAIN	Word bag model of DNS domain names			✓
BAG_CS	Word bag model of cipher suit			✓

obtained by feature selection progress. 80% of the data are used as the training set, and the remaining 20% as the test set. We conducted experiments and evaluations at intervals of 1 min, 2 min, 3 min, 4 min, 10 min, 30 min, and 1 hour. For every algorithm, we use 10-fold cross-validation to ensure the result is stable and repeatable. Evaluation indicators include model training time and classification results related to the evaluations. The following indicators were used when evaluating the classification results:

(1) Precision: Pr = TP/TP + FP

(2) Recall: Re = TP/TP + FN

(3) Accuracy: Acc = TP + TN/TP + FP + TN + FN

(4) F_1score: $F_1 = 2 \times \text{Pr} \times \text{Re}/\text{Pr} + \text{Re}$

TP represents the number of positive examples correctly classified in the data, FP is the number of positive examples incorrectly classified, TN is the number of negative examples correctly classified, and FN is the number of negative examples incorrectly classified.

Due to the selected algorithms having hyperparameters, different parameters will have an impact on the accuracy and training speed of the model. RandomizedSearchCV [22] is used in the parameter selection to ensure that the performance of the model in each time interval is the best. The accuracy shown in Figure 3 is the result obtained on the test set. It can be seen that, for the performance of accuracy, the longer the time interval, the greater the deviation of characteristics in the streams of different devices, which brings better classification results. When the time interval is longer than 3 min, the accuracy of the RF and baseline method is stable at about 99.5%. However, a decrease occurred for the kNN algorithm. As we inspected the feature set used in the training, we found that as the time segment became longer, the feature extract frequency became lower, so the feature set became smaller. For the kNN algorithm, the result is strongly dependent on the scale of the feature set unlike the other algorithms. However, in a comparable time segment, the performance of kNN is much worse than that of the other algorithms. To prove that our scheme is statistically better than the baseline method, we conduct 100 times of training and prediction on a 1-minute time segment. As shown in Figure 4, the accuracy of our scheme is statistically 1.5% higher than that of the baseline method.

As shown in Figures 5 and 3, in a short time window, our method's classification performance is better than the baseline method's. As we inspect the features, the DNS interval, NTP interval, and sleep time that the baseline method used are meaningless in a short time interval, but the features chosen in our method always are meaningful. In other words, with a short time segment, some features in the baseline method especially time interval features become homogenized and are inadequate to discriminate different devices. But the features used in our method, constructed from flow-related statistics and selected after the NSGA-III-based feature selection method, are adequate to distinguish devices whether the time segment is long or not.

We also present the detailed classification performance on 3-min time segment because as shown in Figure 5 the accuracy will increase and reach a peak value till the time segment is 3 min. As Table 3 shows, our proposed method based on RF and ET can reach a comparative level with the baseline method. The results show our method's strength clearly: comparative or superior classification performance and much less overheads, which will be clarified in the following.

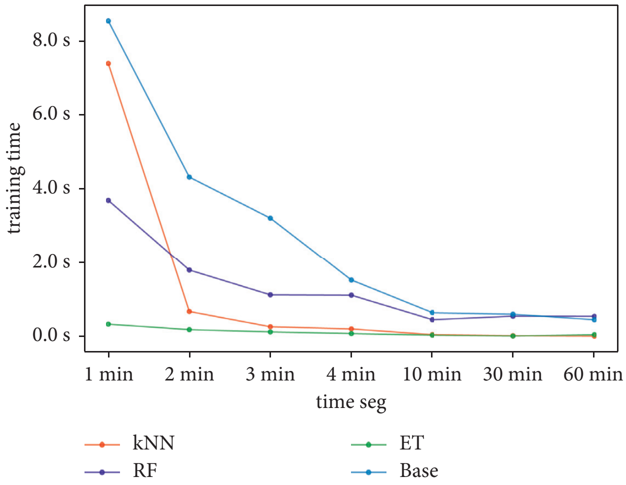

Figure 3: Time cost of different algorithms.

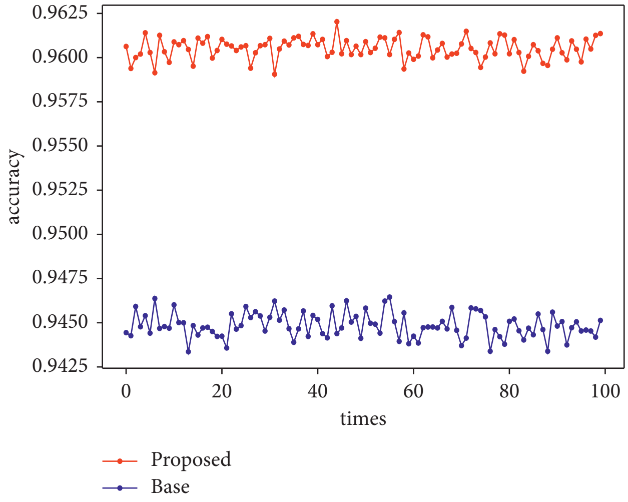

Figure 4: 100 times test.

4.3.2. Overhead of Proposed Method. In terms of training time, the training time of ET is always the shortest, as the time intervals become longer, the shorter the time cost to train the model, and this is mainly for longer time intervals, making the feature set smaller. We should notice that when evaluating the training time of the baseline method, only the time for the second-level classification is considered. The first-level classification will generate a label and a degree of confidence for each sample, and this process will cause heavy cost especially for an enormous data set.

Our method also uses less storage space after feature extraction; as shown in Table 4, as the time intervals become

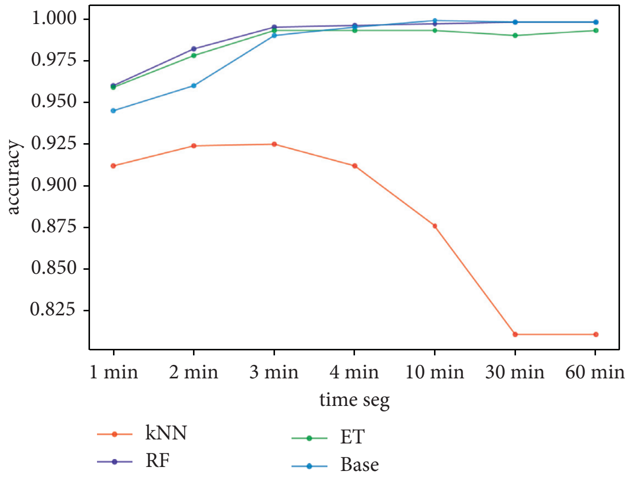

Figure 5: Accuracy of different algorithms.

Table 3: Detailed evaluation on 3 min.

Algorithm	Accuracy	Precision	Recall	F_1 score
kNN	0.912	0.912	0.913	0.912
Base	0.994	0.993	0.994	0.994
RF	0.995	0.995	0.995	0.995
ET	0.993	0.993	0.993	0.993

Table 4: Storage usage.

Time Seg	Proposed (MB)	Baseline (MB)
1 min	35	75
2 min	22.2	62.3
3 min	17	49
4 min	12.1	35.3
10 min	7.4	20
30 min	4.1	16
1 hour	3.3	13.4

longer, the storage used by the proposed method is much less than the baseline method, and this is mainly caused by the text features used in the baseline method. Therefore, our method is superior to the baseline method on storage cost.

Whether in terms of training time or feature dimension, our scheme achieves better performance with less cost. We also obtained a detailed evaluation when the time interval was 3 min. As shown in Table 2, the performance of ET using the selected features in this article was very close to that of the baseline method, while the overhead was significantly reduced. The accuracy of ET is close to the best, which RF achieved, but ET's training is much faster than RF, and on the basis of trade-off on time cost and classification accuracy, we proved that ET is also a valid algorithm to construct an IoT device identification scheme.

5. Conclusion

As the popularization of IoT devices are connected to the Internet, managing and annotating these devices is an essential problem for keeping network security. In this paper, we propose a lightweight IoT device identification scheme based on traffic analysis. This scheme used flow-related statistical features to represent the behavior of IoT devices and a filter feature selection method based on NSGA-III to select effective features. Machine learning algorithms are used to classify devices. Experimental results showed that our proposed scheme can achieve comparable accuracy with much less overhead. Based on the ET algorithm combined with the six attributes port2, port2Cnt, tcpCnt, udpCnt, flowVolume's mode, and flowVolume's variance, the best classification result can be achieved, and the training speed is the fastest. When the time interval is 1 min, an accuracy of 95.8% can be achieved, while the accuracy of the base method is only 94.5%. As for a long time interval like 3 min, our method can achieve an accuracy of 99.3%. At the same time, the overhead is greatly reduced compared with the base method. This method is suitable for deployment on the gateway to identify IoT devices. Future work will focus on cloud services. How to integrate the models, ensure the trustworthiness of the gateway, and improve the performance and security of the distributed device identification system will be the focus of future work.

Data Availability

The data set is the same as the paper "Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics" used and the access link is https://iotanalytics.unsw.edu.au/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (61972073), the Key Program of Natural Science Foundation of Hebei Province of China (F2019201290), and the Natural Science Foundation of Hebei Province of China (F2018201153), Post-Graduate's Innovation Fund Project of Hebei University.

References

[1] Cisco, "Internet of things: connected means informed," Technical Report, 2020, https://www.cisco.com/c/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.html.

[2] A. Riahi Sfar, E. Natalizio, Y. Challal, and Z. Chtourou, "A roadmap for security challenges in the internet of things," *Digital Communications and Networks*, vol. 4, no. 2, pp. 118–137, 2018.

[3] M. A. Khan and K. Salah, "IOT security: review, blockchain solutions, and open challenges," *Future Generation Computer Systems*, vol. 82, pp. 395–411, 2018.

[4] Cisco, "Annual cybersecurity report," Technical Report, 2018, https://www.cisco.com/c/dam/m/digital/elq-cmcglobal/witb/acr2018/acr2018final.pdf.

[5] H. Griffioen and C. Doerr, "Examining mirai's battle over the internet of things," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 743–756, New York, NY, USA, October 2020.

[6] Y. Huang, W. Wang, H. Wang, T. Jiang, and Q. Zhang, "Authenticating on-body iot devices: an adversarial learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5234–5245, 2020.

[7] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[8] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2013.

[9] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional rule-based engine for discovering internet-of-things devices," in *Proceedings of the 27th USENIX Security Symposium*, pp. 327–341, Baltimore, MD, USA, August 2018.

[10] M. Miettinen, S. Marchal, I. Hafeez, N. Asokan, A.-R. Sadeghi, and S. Tarkoma, "IOT sentinel: automated device-type identification for security enforcement in iot," in *Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pp. 2177–2184, IEEE, Atlanta, GA, USA, June 2017.

[11] 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.

[12] M. R. Santos, R. M. Andrade, D. G. Gomes, and A. C. Callado, "An efficient approach for device identification and traffic classification in iot ecosystems," in *Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC)*, pp. 00304–00309, IEEE, Natal, Brazil, June 2018.

[13] F. Le, J. Ortiz, D. Verma, and D. Kandlur, "Policy-based identification of IoT devices' vendor and type by DNS traffic analysis," in *Proceedings of the Policy-Based Autonomic Data Governance*, pp. 180–201, Springer, Berlin, Germany, 2019.

[14] N. Msadek, R. Soua, and T. Engel, "IOT device fingerprinting: machine learning based encrypted traffic analysis," in *Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–8, IEEE, Marrakesh, Morocco, April 2019.

[15] A. Aksoy and M. H. Gunes, "Automated IOT device identification using network traffic," in *Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, Shanghai, China, May 2019.

[16] M. R. Shahid, G. Blanc, Z. Zhang, and H. Debar, "IOT devices recognition through network traffic analysis," in *Proceedings of the 2018 IEEE International Conference on Big Data (Big Data)*, pp. 5187–5192, IEEE, Seattle, WA, USA, December 2018.

[17] Y. Liu, J. Wang, J. Li et al., "Zero-bias deep learning for accurate identification of internet-of-things (iot) devices," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2627–2634, 2021.

[18] B. Charyyev and M. H. Gunes, "Locality-sensitive iot network traffic fingerprinting for device identification," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1272–1281, 2021.

[19] A. Sivanathan, H. H. Gharakheili, F. Loi et al., "Classifying iot devices in smart environments using network traffic characteristics," *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[20] S.-Y. Jiang and L.-X. Wang, "Efficient feature selection based on correlation measure between continuous and discrete features," *Information Processing Letters*, vol. 116, no. 2, pp. 203–215, 2016, https://www.sciencedirect.com/science/article/pii/S0020019015001271.

[21] M. Woźniak, M. Grana, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3–17, 2014.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in python," *The Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.