symmetry

Article

Identification of IoT Devices Based on Hardware and Software Fingerprint Features

Yu Jiang [1,2,3,4,*], Yufei Dou [1] and Aiqun Hu [1,4,5,6]

[1] School of Cyber Science and Engineering, Southeast University, Nanjing 210000, China; jullemon@163.com (Y.D.); aqhu@seu.edu.cn (A.H.)
[2] Purple Mountain Laboratories, Nanjing 210000, China
[3] Key Laboratory of Computer Network Technology of Jiangsu Province, Nanjing 210000, China
[4] Frontiers Science Center for Mobile Information Communication and Security, Southeast University, Nanjing 210000, China
[5] School of Information Science and Engineering, Southeast University, Nanjing 210000, China
[6] State Key Laboratory of Mobile Communication, Southeast University, Nanjing 210000, China
[*] Correspondence: jiangyu@seu.edu.cn

Abstract: Unauthenticated device access to a network presents substantial security risks. To address the challenges of access and identification for a vast number of devices with diverse functions in the era of the Internet of things (IoT), we propose an IoT device identification method based on hardware and software fingerprint features. This approach aims to achieve comprehensive "hardware–software–user" authentication. First, by extracting multimodal hardware fingerprint elements, we achieve identity authentication at the device hardware level. The time-domain and frequency-domain features of the device's transient signals are extracted and further learned by a feature learning network to generate device-related time-domain and frequency-domain feature representations. These feature representations are fused using a splicing operation, and the fused features are input into the classifier to identify the device's hardware attribute information. Next, based on the interaction traffic, behavioral information modeling and sequence information modeling are performed to extract the behavioral fingerprint elements of the device, achieving authentication at the software level. Experimental results demonstrate that the method proposed in this paper exhibits a high detection efficacy, achieving 99% accuracy in both software and hardware level identification.

Keywords: Internet of things; hardware and software fingerprint features; device identification; multimodal

check for updates

Citation: Jiang, Y.; Dou, Y.; Hu, A. Identification of IoT Devices Based on Hardware and Software Fingerprint Features. *Symmetry* **2024**, *16*, 846. https://doi.org/10.3390/sym16070846

Academic Editors: Shangce Gao and Jingang Shi

Received: 7 May 2024
Revised: 13 June 2024
Accepted: 28 June 2024
Published: 4 July 2024

1. Introduction

The Internet of things (IoT) is a key component of today's digital world, profoundly changing the way we live and conduct business. IoT applications are everywhere, ranging from smart home devices to industrial automation, and from health monitoring to urban infrastructure. However, with the continuous expansion of the Internet of Things (IoT), concerns regarding IoT security are increasingly garnering attention.

In the era of IoT, the network environment is filled with a huge number of IoT devices with different functions and types. How to accurately identify and effectively regulate these devices in a complex and dangerous network environment is an urgent problem for all industries in the IoT ecosystem. IoT security faces many challenges. First, the sheer number of IoT devices, with their wide variety and different functions, makes device security management very complex. Second, numerous IoT devices lack high security configurations due to environmental constraints, posing challenges in achieving strong security authentication with limited configurations. Lastly, IoT devices are usually exposed to diverse environmental conditions and are susceptible to physical attacks.

Traditional authentication schemes for IoT devices, whether reliant on lightweight public key algorithms or pre-shared key authentication techniques, encounter challenges such as key leakage and the elevated costs associated with key generation and distribution.

In contrast, emerging hardware and software fingerprint technology can be used as potent tools for IoT device authentication, owing to their heightened security and simplicity of deployment. However, relying only on a single hardware or software fingerprinting technology cannot completely solve the problem. While software fingerprinting can capture the usage behavior of a device, extracted traffic features tend to be more intricate and less directly linked to the device. Hardware fingerprinting, on the other hand, provides a more stable device signature, but is susceptible to variations in signal transmission environments. Therefore, to achieve highly precise device identification within intricate IoT environments, this paper proposes a multi-level authentication method based on the combination of hardware and software features. In summary, our main contribution in this paper include the following:

- In this paper, through the combination of software and hardware fingerprints, device features can be acquired from various dimensions. This approach reduces misjudgments caused by the instability of individual features and enhances the security of authentication. In addition, the combination of hardware and software features can better adapt to different usage scenarios and environmental changes, thereby enhancing the security and robustness of device identification.
- A multimodal hardware fingerprint element extraction method is proposed in hardware level authentication. In this paper, time-domain feature information and frequency-domain feature information are integrated into a unified model, wherein a feature learning network is utilized to delve deeper into the time-domain feature representation and frequency-domain feature representation. Subsequently, multimodal fusion features are generated through splicing operations to achieve more reliable identification of device hardware attributes.
- In software-level authentication, which does not strictly rely on identifying specific fields of a packet, its applicability is broader and can be utilized for feature extraction without prior knowledge of the protocol. In addition, this paper emphasizes the extraction of behavioral features derived from the network packet behavior. It models the communication behavior of network packets from devices, constructs a sequence to encapsulate device behavior, and preserves the temporal relationship among interacting traffic. Compared to the individual packet level, this paper not only considers the device behavior itself but also incorporates timing information regarding interaction behavior. This approach contributes to a more precise classification of different devices.

The rest of the paper is organized as follows. In Section 2, we review related work. In Section 3, the methodological architecture of this paper is presented. Sections 4 and 5 describe the software level and hardware level authentication methods for IoT devices, respectively. Section 6 is the experimental part where the experimental results are compared and discussed. Finally, we conclude the paper in Section 7.

2. Related Works

Due to the large number and diversity of IoT devices [1], device identification has become a significant issue in system security. Most IoT devices have limited computing and storage resources, which makes it difficult to implement strict security measures on these devices. However, the emerging hardware and software fingerprinting technology has gained widespread recognition in the realm of device identification. Its high security and ease of implementation make it a hot trend in the current network security field.

2.1. Software Fingerprint Feature Authentication Technology

Device authentication technology based on software fingerprints refers to the identification of IoT devices by capturing software traffic characteristics related to devices such as browsers and wireless drivers. Through the collection of network traffic and the layered parsing of protocol information, the fundamental communication information and behavioral attribute characteristics of network users can be obtained.

Zhang et al. [2] conducted a detailed analysis of the traffic of different hosts by capturing a large number of packets. The hosts were identified through feature extraction at the host level. By comparing the network traffic variations of different hosts over a day and analyzing the trends to obtain features, statistical and differential features are incorporated to construct the time-varying characteristics of the network flow for host identification. The experimental results demonstrate that utilizing the time-varying characteristics of network flow for host identification can achieve good results. Yang et al. [3] summarized a series of statistical features by analyzing a large amount of network packets. Radhakrishnan et al. [4] utilized packet inter-arrival time (IAT) and transmission time (TT) as device features, employing Bayesian regularization as well as a quantified conjugate gradient approach to generate device fingerprints. Yang et al. [5] proposed a method for device identification to accomplish access control of suspicious devices. This method involves setting up a whitelist, constructing characteristic fingerprints of communication traffic, and employing the random forest method to train the device identification model. Next, in order to manage the internal devices, an intelligent security management model is proposed, which constructs an ontological threat model based on assets, vulnerabilities, security mechanisms, and other factors. Finally, the effectiveness of the device recognition model is verified through experiments, achieving a recognition accuracy exceeding 96%. Pinheiro et al. [6] models the behavior of the network packets communicated by the devices. It classifies devices at the individual packet level using generalizable features. In addition to the 111 features extracted from the network packet headers, payload entropy [7], protocol (from TCP-IP layers), source and destination port class [8] were included. Kostas et al. [9] proposes a solution that uses packet length statistics extracted from encrypted traffic to characterize the behavior of IoT devices and events in a smart home scenario. The solution uses only the statistical mean, the standard deviation, and the number of bytes transmitted over a one-second window as features to achieve device identification.

The above literature focuses on software fingerprinting using specific network traffic characteristics or statistical features. However, these methods have certain limitations. Firstly, they are often restricted to specific communication protocols, which limits their applicability. Secondly, these methods do not account for the temporal relationships between packets, resulting in an inability to fully capture the dynamic behavior of devices. Finally, the extraction and analysis of statistical features usually require significant computational resources, making it difficult to apply these methods effectively on resource-constrained devices. In summary, these limitations indicate that existing software fingerprint authentication methods have certain challenges in practical applications and require further optimization and improvement.

2.2. Hardware Fingerprint Feature Authentication Technology

With the popularity of wireless devices, IoT needs to manage an increasing number of devices. Device intrusion detection is a top priority, and traditional authentication methods struggle to meet the application requirements of distributed heterogeneous IoT. Therefore, hardware fingerprint identification techniques based on the physical layer, which cannot be copied and replaced, have received a lot of attention.

Knox et al. [10] extracted phase information from baseband signals transmitted by wireless devices, utilizing it as a hardware fingerprint feature to identify the transmitter. Carbino et al. [11] proposed cross-model discrimination (CMD) and like-model discrimination (LMD), focusing on solving the identification problem of similar devices. Carbino employed nearest neighbor (NN) and maximum likelihood (ML) algorithms to construct the adjudicator, achieving a recognition accuracy of 76.73% with the NN algorithm and 91.38% with the ML algorithm on the more effective CMD model. Guillem et al. [12] and Amani et al. [13] both employ time-domain and frequency-domain features for device identification. Shen et al. [14] utilizes radio frequency (RF) fingerprinting based on spectrograms for device identification. Through conducting carrier frequency offset (CFO) compensation to ensure system stability during experiments, a classification accuracy of 97.61% was attained in distinguishing 20 LoRa devices in real wireless environments. Given that the signal differences between different devices of the same type are caused by hardware damage and are mainly concentrated in the high frequencies, Liao et al. [15] proposes reconstructing the high-frequency component of the signal through Fourier transform, attention mechanism, and inverse Fourier transform. Subsequently, features are extracted from the time dimension based on the reconstructed high-frequency component of the signal for device identification. Almashaqbeh et al. [16] investigate the effectiveness of wavelet decomposition, specifically the dual-tree complex wavelet transform (DT-CWT), in extracting robust features for radio frequency fingerprinting (RFF) of Bluetooth devices. Abbas et al. [17] provides a comprehensive review of various radio frequency (RF) fingerprinting methods used for device identification. It also discusses the principles, techniques, and applications of RF fingerprinting.

It can be found that methods for device authentication based on hardware fingerprints generally suffer from the problem of feature homogeneity. Many studies solely utilize either frequency or time domain features and neglect to sufficiently integrate multiple features in a comprehensive analysis. However, methods that rely on a single feature may not perform well when equipment or environment changes. For example, methods based on frequency-domain features may be unstable when channel conditions change, methods based on time-domain features may be unreliable when device locations change, etc.

Meanwhile, a substantial amount of current research results utilize only software fingerprints or hardware fingerprints for device identification. However, to date, there has been no method that combines both for device identification. Based on this, this paper proposes a method that combines software and hardware fingerprints for device identification. During the software-level authentication phase, device behavioral features are extracted based on network flows. During the hardware-level authentication phase, time domain and frequency domain features are effectively fused for identification. At both stages, more comprehensive feature information is captured, and through multi-level authentication, the reliability of authentication is enhanced.

3. Method Architecture

The subtle differences and inherent properties of the device can be captured through hardware feature extraction, while the network interaction traffic of the device includes amount of behavioral information. Moreover, there are additional temporal relationships within the interaction traffic. Therefore, in order to precisely identify IoT devices and improve the security and robustness of identification, this paper proposes an IoT device identification method based on hardware and software fingerprint features. The method consists of two steps: hardware-level authentication and software-level authentication, achieving comprehensive “hardware–software–user” authentication. Its organizational structure is visually presented in Figure 1.

Hardware level authentication: In this paper, we propose a multimodal hardware fingerprint element extraction method designed to capture hardware information within device transient signals. The main steps are as follows: multimodal feature extraction, based on the transient signal of the device to extract the device’s feature information from different modalities; multimodal feature fusion, in which the extracted features from different modalities are fused to generate more comprehensive and accurate feature representations; and hardware attribute information identification, in which, based on the fused feature representation, the hardware attribute information of the device is identified, thus achieving identity authentication at the hardware level.

Software-level authentication: The main steps are as follows: behavioral information modeling, for the behavioral information contained in the device traffic, based on the packet level to extract device-related behavioral feature information; sequence information modeling, in which, considering the behavioral information contained in the device interaction traffic, the timing features within the device interaction traffic are further extracted based on the network flow level to construct device behavioral features; and software attribute information identification, in which, utilizing the extracted behavioral features, the device's behavioral pattern is analyzed to identify its software attribute information.

Finally, the combination of hardware-level and software-level identification results achieves multi-level device authentication and enhances the reliability of authentication.

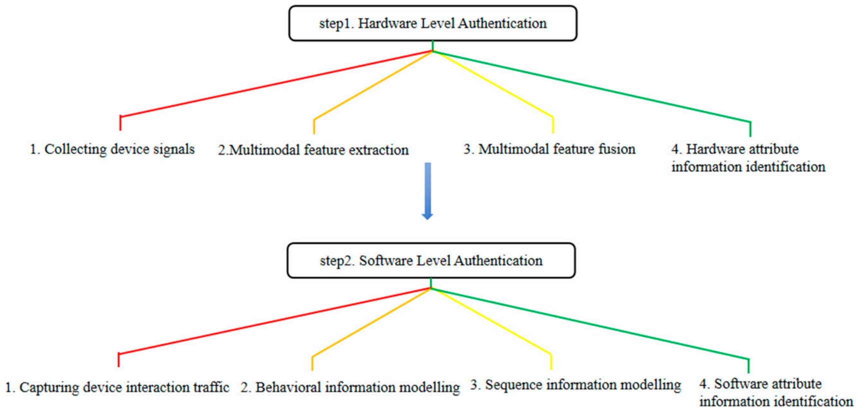

step1. Hardware Level Authentication
1. Collecting device signals
2.Multimodal feature extraction
3. Multimodal feature fusion
4. Hardware attribute information identification
step2. Software Level Authentication
1. Capturing device interaction traffic
2. Behavioral information modelling
3. Sequence information modelling
4. Software attribute information identification

Figure 1. The steps pursed in the study.

4. Hardware Level Authentication

To achieve effective authentication at the device hardware level, this paper proposes a multimodal hardware fingerprint element extraction method. By capturing the feature information from different modalities, a more comprehensive feature representation is generated to enhance the reliability of device hardware level authentication. The method consists of multimodal feature extraction, multimodal feature fusion, and hardware attribute information authentication. Its architecture is shown in Figure 2. The main functions of each part are designed as follows:

1. Multimodal feature extraction: extract time-domain and frequency-domain feature information related to the device from the device's transient signals. Subsequently, employ the feature learning network to conduct in-depth feature learning, thereby generating time-domain and frequency-domain features of the device.
2. Multimodal feature fusion: the extracted time-domain features and frequency-domain features are fused using a splicing operation to obtain multimodal fusion features.
3. Hardware attribute information identification: device hardware attribute information is identified based on fusion features. The captured multimodal fusion features are used as inputs to generate classification probabilities using fully connected neural networks and softmax activation functions, aimed to detect device identity.

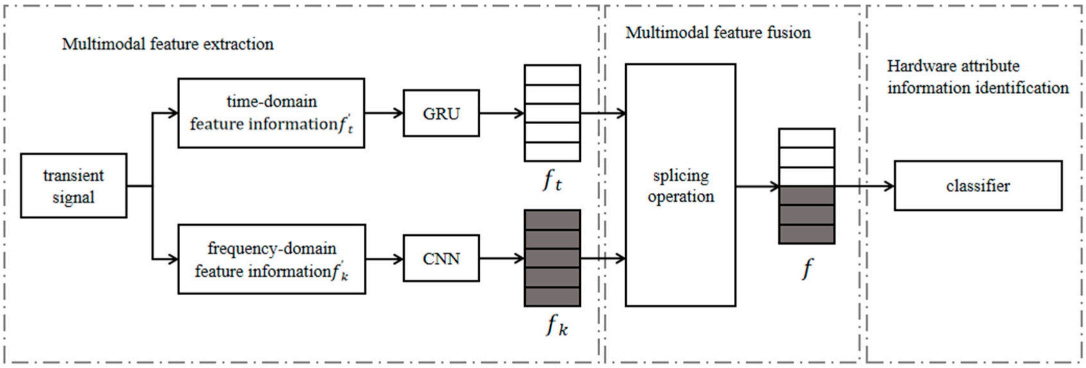

Multimodal feature extraction
transient signal
time-domain feature information f_t'
frequency-domain feature information f_k'
GRU
CNN
f_t
f_k
Multimodal feature fusion
splicing operation
f
Hardware attribute information identification
classifier

Figure 2. Architecture of the multimodal hardware fingerprint element extraction method.

4.1. Multimodal Feature Extraction

4.1.1. Signal Collection

This paper uses the USRP N210 device for Bluetooth signal collection. Specifically, we enable the Bluetooth device to establish a connection with the collection device and ensure that the two can communicate properly. Next, we configure the collection device to monitor the operating frequency band of the Bluetooth device and thus can complete the collection of the signals. The collected raw Bluetooth signal is shown in Figure 3.

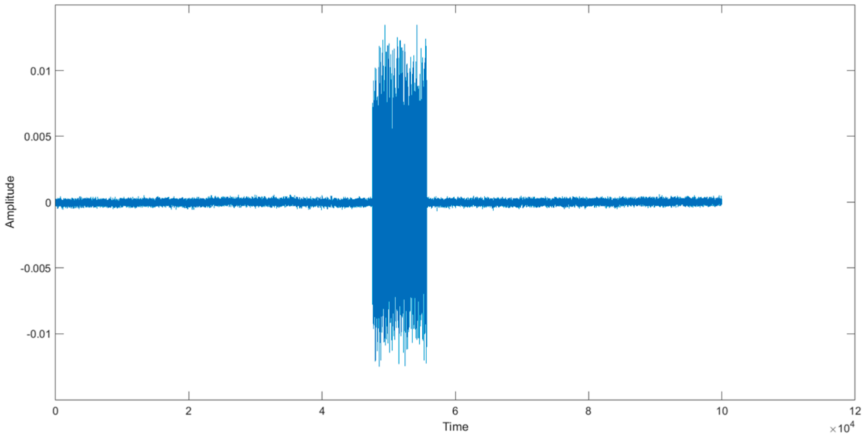


Figure 3. Raw Bluetooth signal.

In order to analyze and extract the hardware fingerprint information of the device, the collected data need to be processed. For each data sample, it is first necessary to extract the valid signals from the sampled data. Then, normalization operations are performed on these signals. The processed signals are shown in Figure 4.

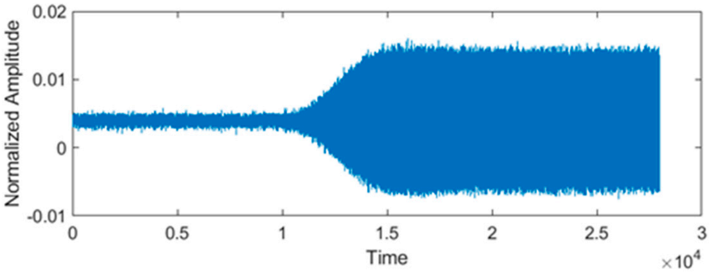


Figure 4. Processed signal.

4.1.2. Data Preprocessing

The transient segment of each signal can be identified through energy analysis, as shown in Figure 5. We extract the transient segments of all signals to form the initial sample dataset.

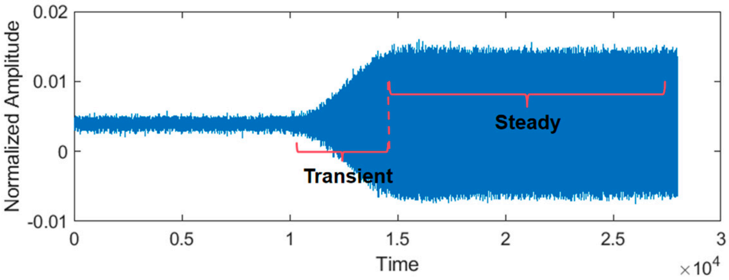


Figure 5. Identify signal transient segments.

In order to enrich the sample dataset and improve the robustness and generalization ability of the model, we adopt the method of signal sample point interval extraction. Specifically, we extract sample points at certain intervals from different starting points within the transient signal segments to form new, non-repetitive samples. In this way, the number of samples in the dataset is effectively increased, while retaining the basic characteristics of the transient segment of the signal. Figure 6 shows the new transient signal segments acquired after preprocessing.

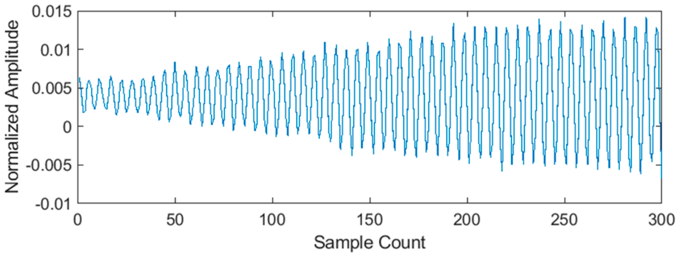


Figure 6. The new transient signal segment acquired after preprocessing.

4.1.3. Time-Domain Feature Information

In this paper, we propose to augment the number of sample points in the preprocessed transient signal segments by up-sampling by 10 times the original number, resulting in 3000 sample points, thereby emphasizing local features in the signal. Then, the up-sampled 3000 sample points are divided into 10 segments, each segment containing 300 sample points. The signal before and after up-sampling is shown in Figure 7.

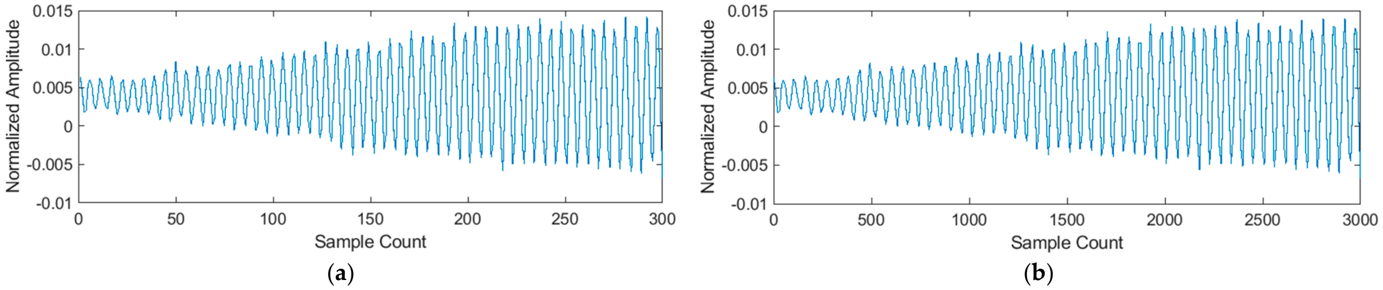


Figure 7. Signal before and after up-sampling. (**a**) Preprocessed transient signal segment. (**b**) Up-sampling of transient segment.

Time-domain feature extraction is performed separately for each segment of samples. Instantaneous phase, frequency, and amplitude values are computed for each segment consisting of 300 sample points using the Hilbert transform (HT). Equation (1) expresses the HT for an original signal $x(t)$. Instantaneous phase, instantaneous frequency, and instantaneous amplitude are expressed by Equations (2)–(4).

$$h(t) = H[x(t)] = \frac{1}{\pi}\int_{-\infty}^{+\infty}\frac{x(\tau)}{t-\tau}dt \tag{1}$$

$$p_x(t) = \arctan(\frac{h(t)}{x(t)}) \tag{2}$$

$$f_x(t) = \frac{1}{2\pi}[\frac{dp_x(t)}{dt} mod(2\pi)] \tag{3}$$

$$a_x(t) = \sqrt{x(t)^2 + h(t)^2} \tag{4}$$

The statistical metrics used in this paper and their associated calculations are shown in Table 1. In the equations, $F(N)$ represents the instantaneous phase, frequency or amplitude of each segment, and N is the number of sample points contained in each segment.

Standard deviation and variance are statistics used to measure the dispersion of data, while the arithmetic mean and median are common trend metrics used to describe the central tendency of the data distribution. The arithmetic mean assigns equal weight to each data point. For datasets that obey a normal distribution, the arithmetic mean is a good representation of the center of the data. Unlike the arithmetic mean, the harmonic mean assigns greater weight to smaller values, offering a more precise estimate particularly in situations where extreme values or biases within the dataset. The geometric mean can indicate the relative relationship between values in a dataset. Skewness is a statistic used to assess the asymmetry of a distribution. When the arithmetic mean and median are not equal, skewness shows deviations from a normal distribution and measures the degree to which individual sample values deviate from the mean. Kurtosis is a statistical feature that provides information about the density of sample values within a distribution and can indicate whether these values are tightly clustered around the mean or spread over a wider range. When calculating the median, the following factors are taken into consideration: the lower class boundary l of the median class, the size h of the median class interval, the frequency f corresponding to the median class, the total number of observations N (the sum of the frequencies), and the cumulative frequency c of the median class.

Table 1. Statistical indicators and their formulas.

Number	Statistic	Equations
1	harmonic mean ()	$N\frac{1}{\sum_{n=1}^{N}\frac{1}{F(n)}}$
2	geometric mean ()	$\left(\prod_{n=1}^{N} F(n)\right)^{\frac{1}{N}}$
3	arithmetic mean (μ)	$\frac{1}{N}\sum_{n=1}^{N}\lvert F(n)\rvert$
4	standard deviation (σ)	$\sqrt{(\lvert F(n)\rvert-\mu)^2}$
5	skewness (γ)	$\frac{1}{N}\sum_{n=1}^{N}\left(\frac{\lvert F(n)-\mu\rvert}{\sigma}\right)^3$
6	kurtosis (k)	$\frac{1}{N}\sum_{n=1}^{N}\left(\frac{\lvert F(n)-\mu\rvert}{\sigma}\right)^4$
7	variance (σ^2)	$\frac{1}{N}\sum_{n=1}^{N}\left(\frac{\lvert F(n)-\mu\rvert}{\sigma}\right)^2$
8	median($\widetilde{x}$)	$l+\frac{h}{f(\frac{N}{2}-c)}$

The instantaneous phase, instantaneous frequency, and instantaneous amplitude of each segment are computed using the above statistics, resulting in the features shown in Table 2. This process generates an array of features for each segment $t_{seg} = \{a_1, a_2, \cdots, a_{20}\}$. An array of features for all segments is extracted to construct the sequence time-domain features $f_t' = \{t1_{seg}, t2_{seg}, \cdots, t10_{seg}\}$, representing the device-related time-domain feature information.

Table 2. Feature description.

Number	Feature
1	Harmonic mean value of $a_x(t)$
2	Geometric mean value of $a_x(t)$
3	Arithmetic mean value of $a_x(t)$
4	Standard deviation value of $a_x(t)$
5	Skewness value of $a_x(t)$
6	Kurtosis value of $a_x(t)$
7	Variance value of $a_x(t)$
8	Median value of $a_x(t)$
9	Arithmetic mean value of $p_x(t)$
10	Standard deviation value of $p_x(t)$
11	Skewness value of $p_x(t)$

Table 2. *Cont.*

Number	Feature
12	Kurtosis value of $p_x(t)$
13	Variance value of $p_x(t)$
14	Median value of $p_x(t)$
15	Arithmetic mean value of $f_x(t)$
16	Standard deviation value of $f_x(t)$
17	Skewness value of $f_x(t)$
18	Kurtosis value of $f_x(t)$
19	Variance value of $f_x(t)$
20	Median value of $f_x(t)$

4.1.4. Frequency-Domain Feature Information

Performing fast Fourier transform (FFT) on a signal is a commonly used method of feature extraction in the frequency domain. By applying FFT, the signal can be transformed from the time domain to the frequency domain, providing a representation of the signal in the frequency domain. In this paper, the FFT is performed on the preprocessed transient signal segment of the device to obtain the frequency-domain feature information f_k' related to the device. Equation (5) expresses the FFT for an input signal $x(n)$. Here, N is the length of the input signal, and $X(k)$ represents the complex value of the k-th frequency component in the frequency domain. The frequency-domain feature information f_k' is expressed by Equation (6).

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi kn/N}, k = 0, 1, \cdots, N-1 \tag{5}$$

$$f_k' = |X[1 : N/2 + 1]| \tag{6}$$

As shown in Figure 8, (a) is the preprocessed transient signal segment, and (b) is the corresponding one-sided spectrogram of the signal segment, i.e., the device related frequency-domain feature information. In addition, Figure 9 shows the frequency-domain feature information of different devices. It can be observed that the frequency-domain feature information of different devices exhibits significant differences.

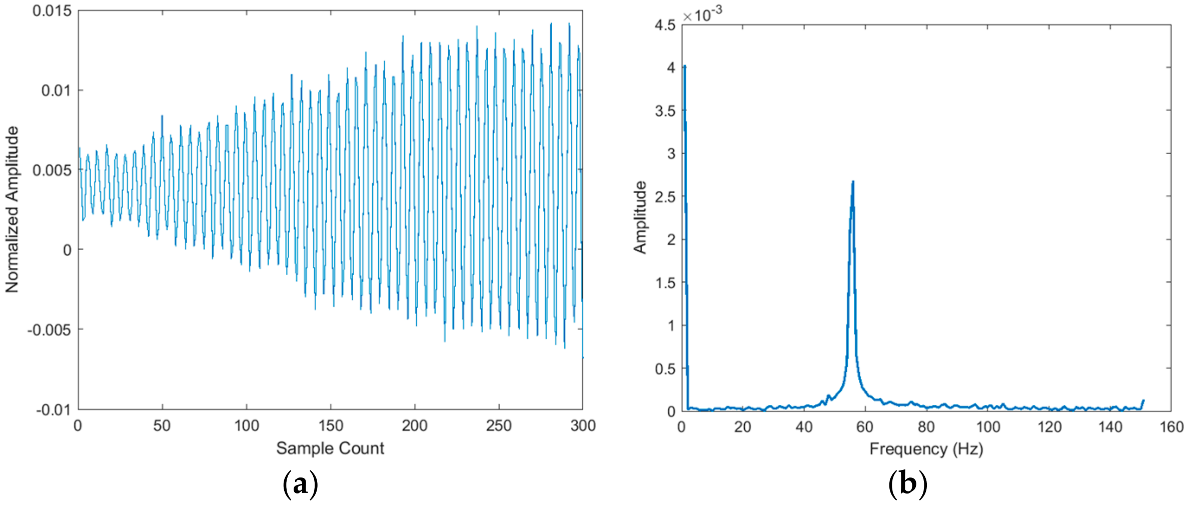


Figure 8. Frequency-domain feature extraction. (**a**) Preprocessed transient signal segment. (**b**) One-sided spectrogram of the signal segment.

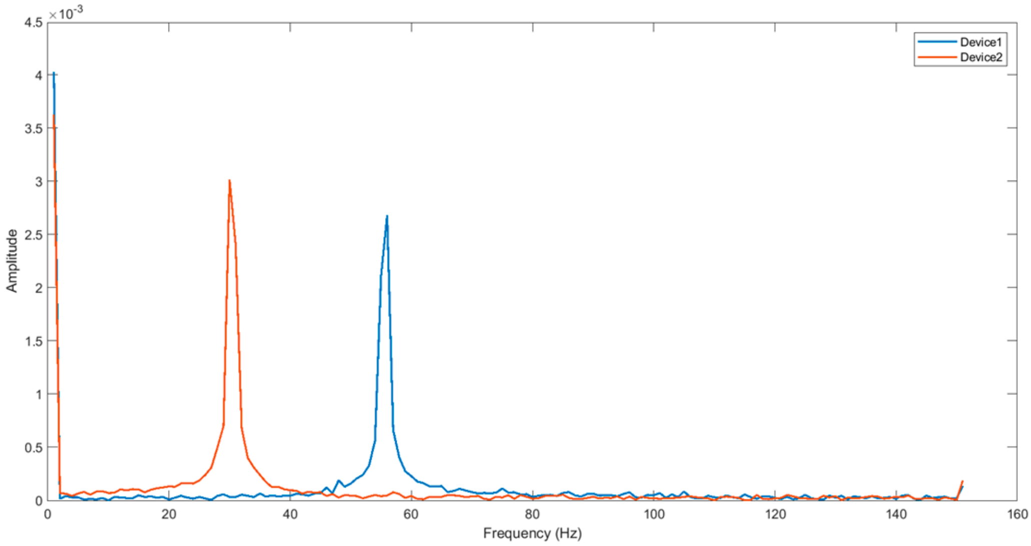


Figure 9. Frequency-domain features of different devices.

4.1.5. Feature Generation

For the captured time-domain and frequency-domain feature information, the feature learning network is used to further extract locally important information and capture deeper feature representations. The feature learning network consists of GRU model and CNN model. The input time-domain feature information and frequency-domain feature information are learnt separately to generate the final time-domain feature representation and frequency-domain feature representation.

1. Time-domain feature extraction based on GRU model

Considering that gated recurrent unit (GRU) is a neural network suitable for processing and analyzing time series data, it can effectively capture the temporal dependencies in time-domain features. Compared with the traditional RNN, GRU simplifies the network structure and reduces the computational complexity by introducing a gating mechanism. Therefore, this paper proposes further learning time-domain feature information through the GRU model.

GRU is a variant of recurrent neural network. Compared with the traditional RNN structure, GRU introduces gating mechanisms, which allow the model to adaptively choose how much historical information should be retained at the current moment and how much new information should be updated at the current moment. These gating mechanisms can effectively alleviate the gradient vanishing and explosion problems of RNN model. Additionally, they can also capture longer-term dependencies and improve the model's expressiveness and generalization ability.

The algorithmic procedure for further learning of time-domain feature information using the GRU model is shown in Algorithm 1.

The specific steps are as follows:

- Initialize the GRU model and define model-related parameters.
- Input the feature matrix into the GRU model: for each time step t, the feature matrix x_t and the hidden state h_{t-1} of the previous time step are input into the GRU model to obtain the output h_t of the current time step.
- Obtaining the output of the GRU model: combining the output h_t from all time steps to get the final time-domain feature f_t.

Algorithm 1: Time-domain feature extraction based on GRU model

```
Input: The feature matrix corresponding to the time-domain feature information $f_t'$
Output: The obtained time-domain feature representation $f_t$ after GRU model processing
1: $W_z$, $W_r$, $W_h$, $U_z$, $U_r$, $U_h$, $b_r$, $b_h$ = init_GRU_parameters()
2: /*Defining the initial state*/
   $h_0$ = torch.zeros(batch.size, hidden.size)
3: /*Inputting the feature matrix into the GRU model*/
   outputs = []
      $h_t$ = h0
      for $t \in range(sequence\_length)$ do
      $x_t$ = feature_matrix[:,t,:]
      $z_t$ = sigmoid (torch.matmul($x_t$, $W_z$) + torch.matmul($h_t$, $U_z$) + $b_z$)
      $r_t$ = sigmoid (torch.matmul($x_t$, $W_r$) + torch.matmul($h_t$, $U_r$) + $b_r$)
      tilde_$h_t$ = tanh (torch.matmul($x_t$, $W_h$) + torch.matmul($r_t \times h_t$, $U_h$) + $b_h$)
      $h_t$ = $z_t \times h_t + (1 - z_t) \times$ tilde_$h_t$
      outputs.append($h_t$)
      end for
4: /*Output of the GRU model*/
   $f_t$ = outputs
End
```

2. Frequency-domain feature extraction based on CNN model

Since convolutional neural network (CNN) is good at processing two-dimensional data, such as images and spectrograms, it can effectively extract local features and spatial relationships. Therefore, in this paper, we design to further learn the frequency-domain feature information through the CNN model. The mechanism of local connectivity and weight sharing of CNN makes it more effective in capturing local features. In the convolutional layer, multiple convolutional kernels of different sizes can be used to capture the feature information of multiple abstract levels, so as to obtain deeper frequency-domain features.

In the frequency-domain feature extraction task, the purpose of the convolutional layer is to perform further learning on the input frequency-domain feature information. Each convolution kernel can be regarded as a specific filter, and the sliding window convolution operation on the input frequency-domain feature information can effectively extract the local features of the input data.

The convolution kernel is convolved with the input frequency-domain feature information f_k' to obtain the feature matrix corresponding to each convolution kernel. The convolution operation is expressed by Equation (7). Here, S represents the feature matrix extracted after the convolutional layer operation, while the weight matrix W and bias vector b are the parameters learned by this network.

$$S = \beta(Wf_k' + b) \tag{7}$$

After the convolution operation, it is necessary to perform nonlinear mapping of the convolution results to enhance the expression ability of the feature matrix, thereby facilitating better capture of deeper frequency-domain feature information.

Nonlinear mapping is typically achieved using activation functions, which map the convolution result into a nonlinear space. Commonly used activation functions include ReLU, sigmoid, and tanh. Among them, ReLU is one of the most commonly used activation functions because it is able to maintain the nonlinearity and also the computational speed is relatively fast. In this paper, the ReLU function is used as the activation function. It is expressed by Equation (8).

$$\beta = \max(0, x) \tag{8}$$

The feature representations produced by all the convolution kernels are merged to generate the final frequency-domain feature f_k.

4.2. Multimodal Feature Fusion and Device Hardware Attribute Information Identification

Multimodal feature fusion is performed on the final extracted device-related time-domain features and frequency-domain features. Then, the fused features are input into a classifier to detect device identity. The classifier consists of a fully connected neural network layer and softmax activation function. The specific recognition method is designed as follows:

1. Multimodal feature fusion of time-domain feature f_t and frequency-domain feature f_k is performed through a splicing operation. The result f is used as an input to the classifier. And its expression is given in Equation (9).

$$f = f_t + f_k \tag{9}$$

2. Mapping captured distributed feature representations to the class labeling space of device samples using fully connected neural network layers.
3. The classification probability of the device category is calculated using the softmax activation function, as shown in Equation (10).

$$\hat{y}_i = \sigma(\omega f_i + b) \tag{10}$$

Here, $\sigma(\cdot)$ denotes the softmax activation function. f_i denotes the multimodal fusion feature representation of device i. $\hat{y}_i$ denotes the recognition probability of the classification result of device i. ω and b denote the weights and biases, respectively.

5. Software Level Authentication

In order to achieve multi-level authentication of IoT devices and enhance the reliability of device identification, we intend to extract the behavioral features of the device by behavioral information modeling and sequence information modeling, thereby identifying the device's software attribute information. The method architecture diagram is shown in Figure 10.

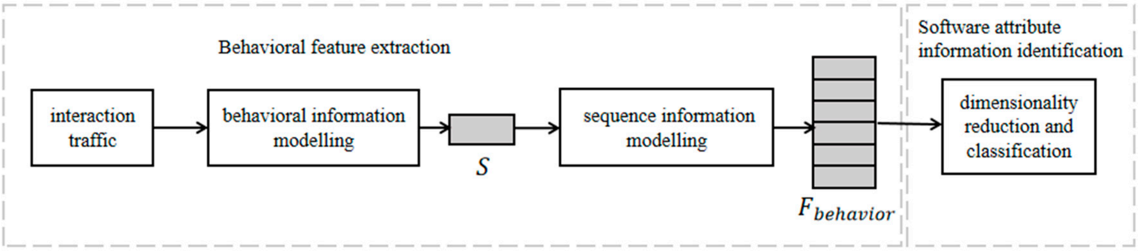

Behavioral feature extraction
Software attribute information identification
interaction traffic
behavioral information modelling
S
sequence information modelling
F_behavior
dimensionality reduction and classification

Figure 10. Software level authentication method architecture diagram.

Thus, on the basis of hardware-level authentication, through the software-level authentication, we can further realize the device "hardware–software–user" authentication.

5.1. Behavioral Feature Extraction

On the one hand, individual packets are unambiguous and may match the behavior of multiple devices. Therefore, the success rate of device identification based on individual packets is limited. On the other hand, interaction traffic is a sequence of packets with the same source/destination IP addresses in both directions of the communication flow. Compared to the unidirectional traffic, interaction traffic contains timing information about the interaction behavior between devices. Therefore, this paper performs device behavioral feature extraction based on the interaction traffic of the target device.

Currently, most of the device fingerprinting based on network traffic focuses on the specific field information in the packet. However, considering the diversity of protocols used by IoT devices, this paper directly performs packet feature information extraction based on message hexadecimal byte streams. This method does not require parsing of packet contents and can perform feature extraction without knowing the priori protocol information, which has wider applicability, even if packet encryption is not affected.

The device behavioral feature extraction method proposed in this paper consists of two main steps: behavioral information modeling and sequence information modeling. Behavioral information modeling is based on the individual packet level, while sequence information modeling is based on the network flow level.

5.1.1. Behavioral Information Modeling

Different IoT devices may use different protocols, but both public and proprietary protocols consist of a series of fields whose values are either relatively fixed or relatively random. Therefore, this paper proposes to analyze packet byte streams in device interaction traffic based on information entropy. The goal is to distinguish the importance of different fields, observe and analyze key-blocks that may serve as behavioral feature information and ultimately generate feature information for each packet.

As shown in Figure 11, the specific process of identifying key-blocks is designed as follows:

1. Packet stacking: stack multiple packets from device interaction traffic together. Ensure that these packets cover multiple patterns of device behavior so that specific patterns or regularities in packet content can be captured across the entire dataset.
2. Left alignment: left-aligned different packets starting from the first byte. The purpose of this step is to make it possible to identify byte changes at the same position by entropy calculations and thus find possible key-blocks.
3. Entropy calculation: local information entropy is calculated for the data in the stacked region according to byte positions. This process helps to determine relatively stable byte positions. Specifically, all bytes appearing at a byte position are considered as a sequence $\{t_1, t_2, \cdots, t_n\}$ containing n distinct bytes, and the information entropy at this particular position is defined by Equation (11). Here, m_i is the number of occurrences of byte t_i and m is the length of the sequence at that particular location.

$$Entropy = -\sum_{i=1}^{n} \frac{m_i}{m} \log_n \frac{m_i}{m} \tag{11}$$

4. Key-block identification: based on the results of entropy calculation, the locations that can be identified as key-blocks are analyzed and determined. Specifically, Figure 12 shows the entropy distribution of packet byte streams in interaction traffic from different IoT devices at different locations. Some byte locations exhibit very low entropy, indicating that the content of these location is relatively stable and can be selected as key-blocks. In particular, for events with equal probability, each $\frac{m_i}{m}$ is equal to $\frac{1}{n}$ and the value of information entropy approaches 1. Given that IoT devices often exhibit more periodic information during normal operation, there are instances of equal probability for specific locations. In other words, these locations may represent variable fields with a limited number of byte-value types. Therefore, we also identify locations with information entropy approaching 1 as key-blocks.

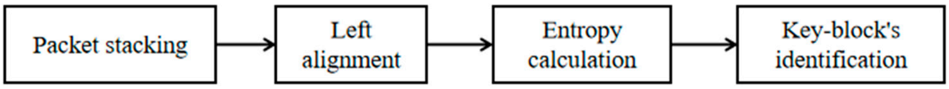


Figure 11. The process of identifying key-blocks.

In order to describe the specific process of behavioral information modeling, we make the following problem definition. We define the packet content $data = f_1 f_2 \cdots f_m$, which consists of the splicing of the different fields of the message, corresponding to the hexadecimal byte stream data as $data \rightarrow b_1 b_2 \cdots b_n$. Based on the entropy values of different byte positions, the key-blocks are analyzed and determined. All the key-blocks form a key-block set S_{key}, which can be used as the behavioral feature information of the current packet as shown in Figure 13.

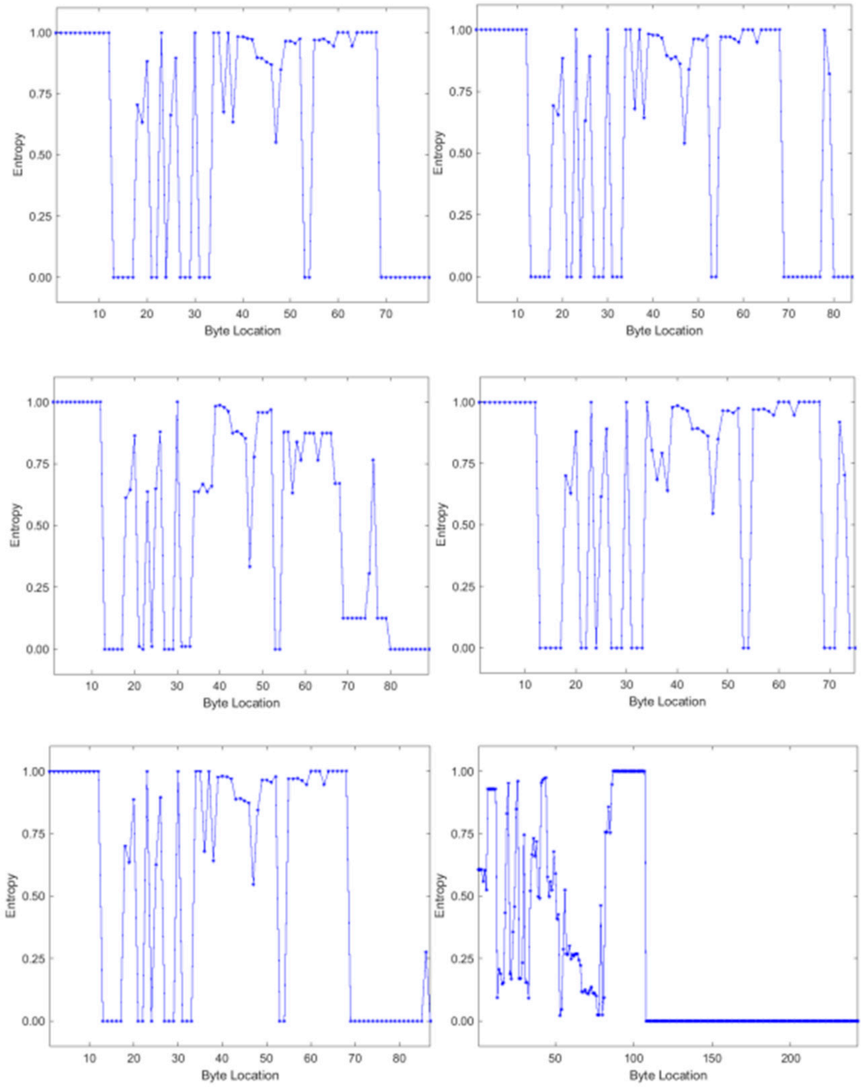
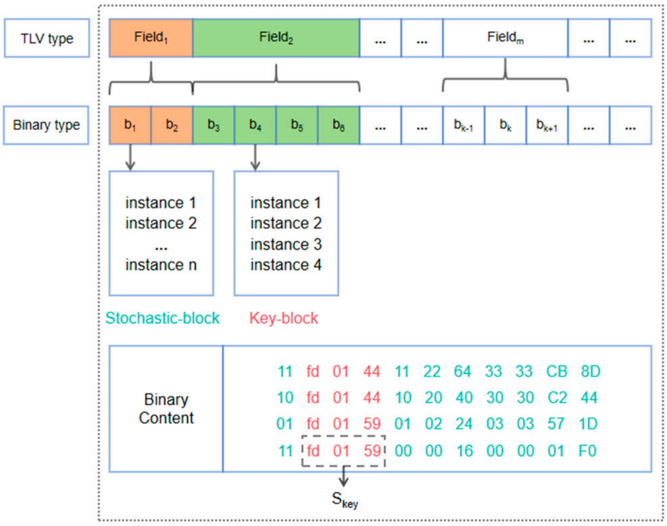


Figure 12. Distribution of information entropy for different devices.



```
11 fd 01 44 11 22 64 33 33 CB 8D
10 fd 01 44 10 20 40 30 30 C2 44
01 fd 01 59 01 02 24 03 03 57 1D
11 fd 01 59 00 00 16 00 00 01 F0
```

Figure 13. The schematic diagram of behavioral information modeling.

As shown in Figure 12, the distribution location of key-blocks is mainly concentrated in the front and back of the packet. Therefore, in the behavioral information modeling stage, it is designed to intercept m bytes from the forward and reverse direction of each packet byte stream, respectively, to constitute the behavioral feature information S of the packet. The length of the feature field is $L = 2m$.

5.1.2. Sequence Information Modeling

The process of sequence information modeling is shown in Figure 14. Based on the interaction traffic of the target IoT device, the behavioral feature information is extracted from g consecutive packets to construct the behavioral feature sequence $F_{behavior} = \{S_1, S_2, \cdots, S_g\}$. $F_{behavior}$ represents the behavioral feature of the device.

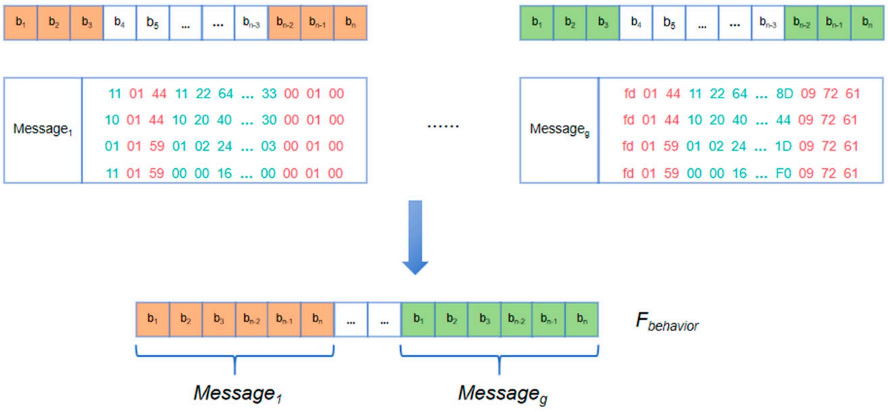

b1 b2 b3 b4 b5 ... ... bn-3 bn-2 bn-1 bn
b1 b2 b3 b4 b5 ... ... bn-3 bn-2 bn-1 bn
Message1
11 01 44 11 22 64 ... 33 00 01 00
10 01 44 10 20 40 ... 30 00 01 00
01 01 59 01 02 24 ... 03 00 01 00
11 01 59 00 00 16 ... 00 00 01 00
......
Messageg
fd 01 44 11 22 64 ... 8D 09 72 61
fd 01 44 10 20 40 ... 44 09 72 61
fd 01 59 01 02 24 ... 1D 09 72 61
fd 01 59 00 00 16 ... F0 09 72 61
b1 b2 b3 bn-2 bn-1 bn ... ... b1 b2 b3 bn-2 bn-1 bn
Fbehavior
Message1
Messageg

Figure 14. The schematic diagram of sequence information modeling.

5.2. *Feature Dimensionality Reduction and Device Software Attribute Information Identification*

Considering the high dimensionality of the extracted device behavioral feature sequences in this paper, this paper proposes to use the linear discriminant analysis (LDA) algorithm to reduce the dimensionality of the extracted behavioral features and use it for the identification of the device's software attribute information.

The principle of LDA is based on maximizing the ratio of the interclass scatter matrix to the intraclass scatter matrix to select the optimal projection direction for data dimensionality reduction and classification. During the process of dimensionality reduction, LDA tries to project the samples into a low-dimensional subspace. This projection aims to maximize the separation between samples from different classes while retaining as much class information as possible. This process is achieved by optimizing an objective function that improves the interclass scatter matrix while minimizing the intraclass scatter matrix to achieve effective feature extraction and maximize classification performance.

6. Experiment

6.1. *Experimental Purpose*

In this paper, the multi-level identity attribute information of IoT devices is captured through the multimodal hardware fingerprint elements extracted based on transient signals and the behavioral fingerprint elements extracted based on the interaction traffic for device identification.

In order to verify the effectiveness of the proposed method and to evaluate the classification performance of the software and hardware level authentication, the following experiments are designed in this paper:

The effect of noise on the multimodal hardware fingerprint element extraction method proposed in this paper is evaluated by performing device hardware-level authentication at different SNR levels.

The effectiveness of the multimodal hardware fingerprint element extraction method proposed in this paper is verified by designing experiments to remove time-domain feature information, frequency-domain feature information, and feature learning network ablation.

By conducting experiments under different settings of feature field lengths in behavior information modeling and sequence lengths in sequence information modeling, the optimal parameter values are verified and determined. Through this process, the features are optimized to achieve the best classification effect.

The classification performance of the method in this paper is evaluated by comparing the soft and hardware recognition methods proposed in this paper with the existing soft and hardware recognition methods, respectively.

6.2. Experimental Dataset

In this paper, we plan for the 6 IoT devices shown in Table 3 to collect device Bluetooth signals and network interaction traffic, so that we can perform device hardware level authentication by collecting the transient portion of the signals and device software level authentication based on the collected network interaction traffic.

Table 3. Introduction of device types.

Category	Brand	Model Number	Number
smart speaker	Xiaodu	Xiaodu Smart Speaker Flagship Version	1
	Xiaodu	Xiaodu Smart Speaker Flagship Version	2
smart socket	Xiaomi	Mijia smart socket	3
	Xiaomi	Mijia mesh mobile socket	4
smart door lock	Xiaomi	Xiaomi Smart Door Lock E20 Cat's Eye Version	5
smart camera	Chuancheng	ST-8296	6

Specifically, data collection was conducted separately for each device, including 100 Bluetooth signals and 20,000 interactive traffic packets. Subsequently, we preprocessed 100 Bluetooth signals corresponding to each device and acquired 1000 preprocessed transient signal segments. Therefore, the experimental dataset in this paper consists of 1000 transient signal segments and 20,000 interactive traffic packets for each device.

6.3. Evaluation Metrics

For the classification task of device identification, the accuracy rate is used as the evaluation index. Meanwhile, to comprehensively evaluate the classification performance, in addition to the accuracy rate (Accuracy), the true positive rate (TPR), and false positive rate (FPR) are included as supplementary evaluation metrics for the device identification task. The accuracy, true positive rate, and false negative rate are expressed by Equations (12)–(14).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$TPR = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

$$FPR = \frac{FN}{TP + FN} \tag{14}$$

Here, true positive (TP) denotes the number of samples correctly categorized as Category A and actually belonging to Category A, true negative (TN) denotes the number of samples correctly categorized as Non-Category A and actually not belonging to Category A, false positive (FP) denotes the number of samples incorrectly categorized as Category A but actually not belonging to Category A, and false negative (FN) denotes the number of samples incorrectly categorized as Non-Category A but actually belonging to Category A.

6.4. Noise Impact Analysis

Additive white Gaussian noise (AWGN) is a prevalent noise model in communication systems, characterized by a uniform power spectral density and Gaussian-distributed amplitude. Introducing AWGN to a signal is often used to simulate a noisy environment in a communication system to evaluate the performance of the system. In this paper, by adding AWGN to the preprocessed transient signal segments (as shown in Figure 6), we conduct device hardware-level authentication experiments at different SNR levels to assess the impact of noise on the proposed multimodal hardware fingerprint element extraction method. The following is a step-by-step description of adding AWGN to a signal:

1. Determining the signal and noise power: First, the power of the original signal and the desired SNR must be determined. The signal power can typically be obtained by calculating the average power of the signal. The signal power is expressed by Equation (15). Here, N is the number of sampling points of the signal, and $x[n]$ is the amplitude value of the signal. The signal-to-noise ratio is typically measured in decibels (dB) and can be converted to a linear ratio using Equation (16).

$$P_{signal} = \frac{1}{N}\sum_{n=0}^{N-1}|x[n]|^2 \tag{15}$$

$$SNR_{linear} = 10^{\frac{SNR_{dB}}{10}} \tag{16}$$

2. Calculating the noise power: The required noise power is calculated based on the signal-to-noise ratio and the signal power, as shown in Equation (17).

$$P_{noise} = \frac{P_{signal}}{SNR_{linear}} \tag{17}$$

3. The standard Gaussian noise is adjusted according to the desired noise power, resulting in Gaussian noise with zero mean and variance P_{noise}. The generated Gaussian noise is expressed by Equation (18).

$$noise = \sqrt{P_{noise}} \cdot standard\_gaussian\_noise \tag{18}$$

4. The generated noise is added to the original signal to obtain the signal with noise, as shown in Equation (19).

$$noise\_signal = signal + noises \tag{19}$$

The accuracy rate is used as an evaluation metric to measure the recognition performance under different values of SNR. The experimental results are shown in Figure 15.

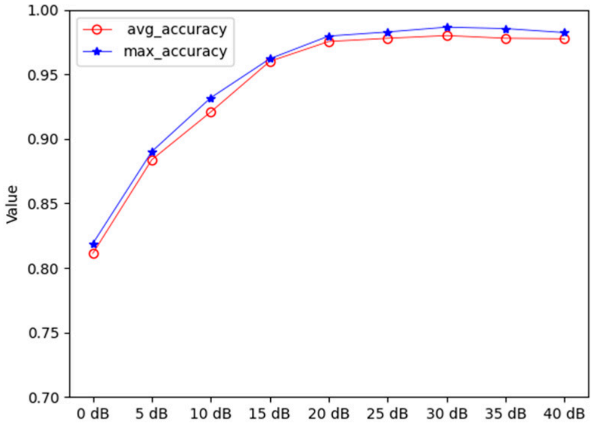


Figure 15. Comparison of performance at different SNR.

By comparing the recognition accuracies under different SNR conditions, it is evident that accuracy exceeds 80% even under extreme SNR conditions. In addition, the accuracy peaks at SNR = 30 dB.

Therefore, in this study, the SNR value is set to 30 dB, based on which hardware level identification can achieve better classification performance.

6.5. Ablation Experiments and Analysis of Results

Since the multimodal hardware fingerprint element extraction method contains several key components, several variants are designed in the experiments to validate the effectiveness of each part of the method.

Method 1: Time-domain feature information is removed on the basis of multimodal hardware fingerprint element extraction method. The frequency-domain feature information is learned using the CNN model to generate the frequency-domain feature representation. Subsequently, device identification is conducted using a fully connected neural network and softmax function.

Method 2: Frequency-domain feature information is removed on the basis of multimodal hardware fingerprint element extraction method. The time-domain feature information is learned by GRU model to generate the time-domain feature representation. Subsequently, device identification is conducted using a fully connected neural network and softmax function.

Method 3: The feature learning network is deleted based on the multimodal hardware fingerprint element extraction method. The multimodal fusion features of this variant consist of directly splicing of the original frequency-domain feature information and time-domain feature information. This comparison method is used to verify the effectiveness of the feature learning network.

The accuracy results for the ablation experiment are shown in Figure 16, from which the following can be seen:

1. By comparing the identification results of Method 1, Method 2 and our study, it can be seen that the identification performance of the method proposed in this paper is better than that of Method 1 and Method 2. It is demonstrated that the combination of the time-domain information and the frequency-domain information can provide more comprehensive information about the attributes of the device hardware, thereby improving the accuracy of the authentication at the device hardware level.
2. Comparing the experimental results of Method 3 and our study, it is observed that the identification accuracy of the method proposed in this paper is better than that of Method 3. This suggests that the feature learning network can capture deeper feature representations, which can improve the identification performance.

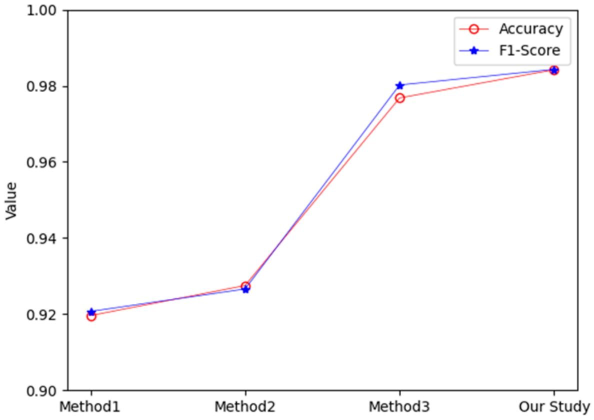
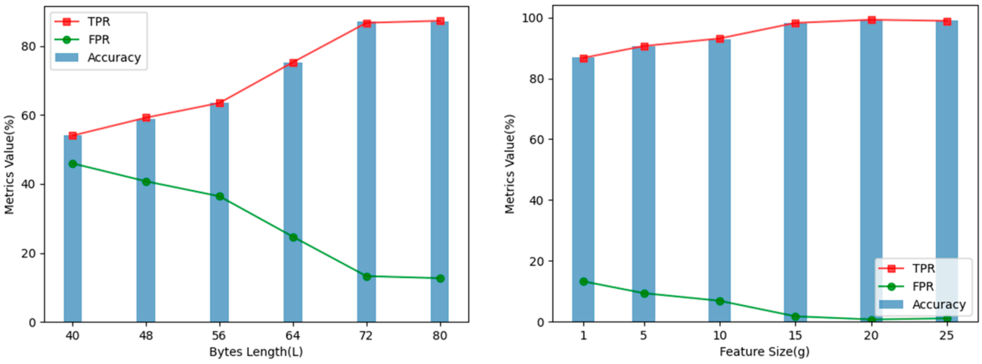


Figure 16. Comparison of ablation experiment results.

6.6. Parameter Setting and Analysis

In the device's software level authentication phase, device-related software attribute information is extracted through behavioral information modeling and sequence information modeling. In order to analyze the influence of feature field length L in behavioral information modeling and sequence length g in sequence information modeling on the classification performance, different values of feature field length L and sequence length g are set to conduct experiments, respectively. The accuracy rate is used as an evaluation index to measure the performance under the influence of different values. The experimental results are shown in Figure 17.



(**a**) (**b**)

Figure 17. The influence of different parameter values on identification performance. (**a**) Different values of feature field length L. (**b**) Different values of sequence length g.

The identification performance for different parameter values is shown in Figure 17, from which the following can be seen:

1. The effect of feature field length in the behavioral information modeling stage: as the feature field length increases, the classification accuracy keeps changing, reaching its peak and subsequently leveling off when the feature field length value L is 72.
2. The effect of sequence length in the sequence information modeling stage: with the change of the value of the sequence length, the classification accuracy tends to increase and then plateau. When the sequence length g is 20, the classification accuracy is the highest.

In summary, in this study, the feature field length value is set to 72, and the sequence length in the sequence information modeling stage is set to 20. Based on these settings, the software level identification can achieve better classification performance.

6.7. Comparative Experiments and Analysis of Results

A multitude of research methods have been proposed in the field of device identification. Among them, most of the hardware fingerprint feature authentication techniques are based on either the time-domain feature or the frequency-domain feature, while the software fingerprint feature authentication relies more on statistical features. However, this paper proposes combining software and hardware fingerprint feature authentication technology to achieve multi-level device authentication. In order to validate the effectiveness of the method in this paper, several representative software and hardware fingerprint feature authentication methods are selected for comparison with the software and hardware level authentication methods proposed in this paper. The performance of these methods is shown in Tables 4 and 5.

Table 4. Performance comparison of hardware fingerprint feature authentication methods.

Ref.	Devices	Feature	Accuracy Rate
[12]	Base stations on the POWER platform	Time-domain RF signal	92.97%
[13]	NI N210 and NI X310	Time-domain RF signal	92.5%
[14]	LoRa DUT	RF signal spectrum	97.61%
Our study	IoT devices	Time-domain feature and frequency-domin feature	99%

Table 5. Performance comparison of software fingerprint feature authentication methods.

Ref.	Feature	Accuracy Rate
[6]	Statistical features of individual packet level	94.3%
[9]	Statistical features of network flow level	96%
Our Study	Behavioral feature of network flow level	99%

The following can be seen from Tables 4 and 5:

1. The hardware level authentication method proposed in this paper exhibits higher identification accuracy compared to the hardware fingerprint feature authentication method solely utilizing either time-domain features or frequency-domain features. It shows that combining time-domain and frequency-domain features can capture more comprehensive feature information, thereby improving the identification accuracy at the hardware level of the device.
2. The performance of the software fingerprint feature authentication method based on statistical features is inferior to the software level authentication method proposed in this paper. This indicates that behavioral features are better at capturing the behavioral patterns and dynamic changes of the device compared to statistical features, thereby resulting in higher accuracy in device identification. Additionally, it can be clearly observed that extracting features based on network flow for device identification yields a higher accuracy rate. This suggests that extracting features at the network flow level can provide a global perspective and contain richer feature information.

7. Conclusions

To address the access security problem of IoT devices, this paper proposes an IoT device identification method based on hardware and software fingerprint features. In the hardware level authentication stage, more comprehensive device hardware attribute information is captured by combining the time-domain and frequency-domain feature information with the multimodal hardware fingerprint element extraction method proposed in this paper. In addition, the feature learning network is used to further acquire deeper feature representations. At the software level authentication stage, the behavioral features related to the device are extracted through the behavioral information modeling and sequence information modeling proposed in this paper. This approach can be used for feature extraction without knowing a priori protocol information, which effectively improves the scope of application of the method.

The experimental results demonstrate that the method proposed in this paper achieves high recognition accuracy in both the software and hardware level authentication stages. It effectively improves the security of device identification through multi-level authentication.

Author Contributions: Methodology, Y.J. and Y.D.; validation, Y.J. and Y.D.; formal analysis, Y.J. and Y.D.; investigation, Y.D.; resources, Y.J.; writing—original draft preparation, Y.D.; writing—review and editing, Y.J. and A.H.; visualization, Y.D.; supervision, Y.J. and A.H.; project administration, Y.J. and A.H.; funding acquisition, Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported in part by the National Key R&D Program of China (No. 2022YFB2902202), the National Natural Science Foundation of China under Grant 62001106, Natural Science Foundation of Jiangsu Province under Grant BK20200350, BK20200352, Jiangsu Provincial Key Laboratory of Network and Information Security No. BM2003201, Project of State Key Laboratory of Mobile Communication, Southeast University No. 2020B05, the Fundamental Research Funds for the Central Universities No. 2242022k30007 and the Purple Mountain Laboratories for Network and Communication Security.

Data Availability Statement: The raw data supporting the conclusions of the article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Jing, Q.; Vasilakos, A.V.; Wan, J.; Lu, J.; Qiu, D. Security of the Internet of Things: Perspectives and challenges. *Wirel. Netw.* **2014**, *20*, 2481–2501. [CrossRef]
2. Zhang, X. Analysis of Time-Varying Characteristics of Network Flows. Master's Thesis, University of Electronic Science and Technology of China, Chengdu, China, 2016.
3. Yang, K.; Li, Q.; Sun, L. Towards automatic fingerprinting of IoT devices in the cyberspace. *Comput. Netw.* **2019**, *148*, 318–327. [CrossRef]
4. Radhakrishnan, S.V.; Uluagac, A.S.; Beyah, R. GTID: A technique for physical device and device type fingerprinting. *IEEE Trans. Dependable Secur. Comput.* **2014**, *12*, 519–532. [CrossRef]
5. Yang, W.; Guo, Y.; Li, T.; Zhu, B. Method for IoT Device Identification and IoT Security Model Based on Traffic Fingerprinting. *Comput. Sci.* **2020**, *47*, 299–306.
6. Pinheiro, A.J.; De M. Bezerra, J.; Burgardt, C.A.P.; Campelo, D.R. Identifying Iot Devices and Events Based on Packet Length from Encrypted Traffic. *Comput. Commun.* **2019**, *144*, 8–17. [CrossRef]
7. Bezawada, B.; Bachani, M.; Peterson, J.; Shirazi, H.; Ray, I.; Ray, I. Behavioral fingerprinting of iot devices. In Proceedings of the 2018 Workshop on Attacks and Solutions in Hardware Security, Toronto, ON, Canada, 15–19 October 2018; pp. 41–50.
8. Miettinen, M.; Marchal, S.; Hafeez, I.; Frassetto, T. 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), IEEE, Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184.
9. Kostas, K.; Just, M.; Lones, M.A. Iotdevid: A behaviour-based fingerprinting method for device identification in the iot. *arXiv* **2021**, arXiv:2102.08866.
10. Knox, D.A.; Kunz, T. Wireless fingerprints inside a wireless sensor network. *ACM Trans. Sens. Netw.* **2015**, *11*, 1–30. [CrossRef]
11. Carbino, T.J.; Temple, M.A.; Lopez, J., Jr. A comparison of phy-based fingerprinting methods used to enhance network access control. In Proceedings of the IFIP International Information Security and Privacy Conference, Hamburg, Germany, 26–28 May 2015; Springer International Publishing: Cham, Switzerland, 2015; pp. 204–217.
12. Reus-Muns, G.; Jaisinghani, D.; Sankhe, K.; Chowdhury, K.R. Trust in 5G open RANs through machine learning: RF fingerprinting on the POWDER PAWR platform. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
13. Al-Shawabka, A.; Restuccia, F.; D'Oro, S.; Jian, T.; Rendon, B.C.; Soltani, N.; Dy, J.; Ioannidis, S.; Chowdhury, K.; Melodia, T. Exposing the fingerprint: Dissecting the impact of the wireless channel on radio fingerprinting. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 646–655.
14. Shen, G.; Zhang, J.; Marshall, A.; Peng, L.; Wang, X. Radio frequency fingerprint identification for LoRa using spectrogram and CNN. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications, Virtual, 10–13 May 2021; pp. 1–10.
15. Liao, Y.; Li, H.; Cao, Y.; Liu, Z.; Wang, W.; Liu, X. Fast Fourier Transform with Multi-head Attention for Specific Emitter Identification. *IEEE Trans. Instrum. Meas.* **2023**, *73*, 2503812.
16. Almashaqbeh, H.; Dalveren, Y.; Kara, A. A study on the performance evaluation of wavelet decomposition in transient-based radio frequency fingerprinting of Bluetooth devices. *Microw. Opt. Technol. Lett.* **2022**, *64*, 643–649. [CrossRef]
17. Abbas, S.; Abu Talib, M.; Nasir, Q.; Idhis, S.; Alaboudi, M.; Mohamed, A. Radio frequency fingerprinting techniques for device identification: A survey. *Int. J. Inf. Secur.* **2024**, *23*, 1389–1427. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.