Lightweight IoT Device Identification Algorithm Based on Multimodal and Adversarial Networks

Authors: Jie Xu, Jie Xu

Date: 2026-03-18T21:27:53+00:00

Abstract

Accurate and rapid identification of IoT devices is a critical research area in IoT security. Addressing the issues of high computational complexity and excessive resource consumption in existing IoT device identification methods, this paper proposes CAGLite, a lightweight IoT device identification algorithm based on flow records. The algorithm extracts statistical attributes, directional sequences, packet header byte sequences, and packet payload sequences from network traffic as traffic features. To improve the accuracy of the algorithm, CNN, BiLSTM, and Transformer adversarial networks are employed respectively according to the characteristics of different attributes. By fusing the extracted feature vectors and analyzing them using a fully connected network, precise identification of IoT devices is achieved. Experimental results demonstrate that while maintaining low computational complexity, the CAGLite algorithm achieves an F1-score of 98.99%, representing a 2.85% improvement over existing state-of-the-art algorithms, thereby providing effective support for the security management of IoT devices.

Full Text

Preamble

Lightweight IoT Device Identification Algorithm Based on Multimodal and Adversarial Networks

Xu Jie[1], Song Fuyao[1], Lan Haoliang[1], Zhang Bo[2] (1. Jiangsu Police Institute, Nanjing, Jiangsu 210031, China)

Abstract

With the rapid proliferation of Internet of Things (IoT) devices, ensuring the security of IoT networks has become a critical challenge. Accurately identify-

ing the types of devices connected to a network is a fundamental prerequisite for implementing effective security policies. However, existing device identification methods often struggle with high computational overhead or insufficient accuracy when dealing with encrypted traffic and diverse device categories. This paper proposes a lightweight IoT device identification algorithm based on multimodal data fusion and Generative Adversarial Networks (GAN). By extracting features from both packet headers and payload statistics, the model captures comprehensive device signatures. To address the problem of imbalanced datasets and improve robustness, an adversarial training mechanism is employed. Furthermore, the model architecture is optimized for deployment on resource-constrained gateway devices. Experimental results on public datasets demonstrate that the proposed algorithm achieves high identification accuracy while significantly reducing inference latency compared to state-of-the-art methods.

1. Introduction

The Internet of Things (IoT) has integrated deeply into various sectors, including smart homes, industrial automation, and healthcare. As the number of connected devices grows exponentially, the heterogeneity and inherent vulnerabilities of these devices pose significant security risks. Malicious actors can exploit compromised IoT devices to launch large-scale Distributed Denial of Service (DDoS) attacks or gain unauthorized access to sensitive networks. Therefore, the ability to automatically and accurately identify the type and manufacturer of IoT devices is essential for network inventory management, anomaly detection, and the enforcement of fine-grained access control.

Traditional device identification techniques primarily rely on MAC address OUI (Organizationally Unique Identifier) lookups or Deep Packet Inspection (DPI). However, MAC addresses can be easily spoofed, and the increasing adoption of encryption protocols (such as TLS/SSL) renders traditional DPI methods ineffective as they cannot inspect the encrypted payload. Recent research has shifted toward machine learning and deep learning approaches that analyze traffic patterns and statistical characteristics. While these methods show promise, they often face two major hurdles: first, the high dimensionality of features leads to significant computational costs; second, the scarcity of labeled data for certain device types results in poor generalization.

To address these challenges, this paper introduces a lightweight identification framework. The main contributions are as follows:

2. 泰兴市公安局情报指挥中心，江苏泰兴 225300）

Abstract: The accurate and rapid identification of IoT devices is a critical research area within IoT security. To address the challenges of high computational complexity and excessive resource consumption inherent in existing IoT device

identification methods, this paper proposes CAGLite, a lightweight IoT device identification algorithm based on flow records.

The CAGLite algorithm extracts statistical properties, direction sequences, packet header byte sequences, and packet payload sequences from network traffic as primary traffic attributes. To enhance algorithmic accuracy, specialized architectures including Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM), and Transformer-based adversarial networks are employed to process these distinct attribute types. By fusing the extracted feature vectors and analyzing them through a fully connected network, the algorithm achieves precise identification of IoT devices.

Experimental results demonstrate that the CAGLite algorithm achieves an F1-score of 98.99% while maintaining low computational complexity. This represents a 2.85% improvement over current state-of-the-art algorithms, providing effective support for the security management of IoT devices.

关键词

Internet of Things (IoT); Device Identification; Transformer; Attribute Fusion
CLC Number: TP393

Document Code: A

A Lightweight IoT Device Identification Algorithm Based on Flow Records Xu Jie1, Song Fuyao1, Lan Haoliang1, Zhang Bo2 1. Jiangsu Police Institute, Nanjing Jiangsu 210031, China;

Abstract

Accurate and rapid identification of IoT devices is a critical research topic in the field of IoT security. Addressing the challenges of high computational complexity and significant resource consumption prevalent in existing identification methods, this paper proposes a lightweight IoT device identification algorithm based on flow records, named CAGLite.

The algorithm extracts a comprehensive set of traffic features, including statistical attributes, direction sequences, header byte sequences, and payload sequences. To optimize identification accuracy, CAGLite employs a multi-model architecture tailored to the specific characteristics of these features, utilizing Convolutional Neural Networks (CNN), Bidirectional Long Short-Term Memory (BiLSTM) networks, Transformers, and Generative Adversarial Networks (GANs). By fusing the extracted feature vectors and processing them through a fully connected network, the system achieves precise identification of IoT devices.

Experimental results demonstrate that while maintaining low computational complexity, the CAGLite algorithm achieves an F1-score of 98.99%. This rep-

resents a 2.85% improvement over state-of-the-art algorithms, providing robust and effective support for the security management of IoT devices.

Foundation Items: - Natural Science Research Project of the Department of Education of Jiangsu Province (24KJB600003, *Research on Anonymous Network Traffic Fingerprint Analysis and Identification*) - Teaching Reform Project of Jiangsu Police Institute (2025B17, *Research on AI-Based Practical Teaching of Web Forensics*)

Key words:

Internet of Things (IoT); Device Identification; Transformer; Attribute Fusion

1 引言

With the rapid development of Internet of Things (IoT) technology, application scenarios such as smart homes, industrial IoT, and smart cities have become deeply integrated into daily life and social infrastructure. According to Statista' s projections, there will be 75 billion IoT devices connected to the Internet globally by 2025, with smart homes accounting for 41%, or 128.6 billion units. However, IoT devices generally suffer from inherent defects such as constrained resources, weak security mechanisms, and lagging firmware updates, making them ideal targets for cyber attackers. These vulnerabilities not only lead directly to functional abnormalities in the devices themselves, affecting normal use, but can also be exploited and manipulated by attackers.

The rapid identification of IoT devices is the foundation of IoT device management and security prevention. This paper proposes CAGLite, a lightweight IoT device identification algorithm based on flow records. The algorithm innovatively integrates four categories of key traffic features (statistical attributes, directional sequences, packet header byte sequences, and packet payload sequences) and matches exclusive extraction networks for different feature characteristics to achieve feature complementarity and precise representation. The main contributions of this paper are as follows: (1) An innovative multi-dimensional feature fusion framework is designed, which for the first time combines four types of features—packet direction sequences, flow statistical features, the first 100 bytes of packet headers, and the first 100 bytes of payloads—to achieve full-dimensional feature coverage ranging from communication patterns and statistical laws to protocol content, thereby enhancing the uniqueness and accuracy of device identification. (2) A lightweight feature extraction strategy is proposed. By using ANOVA F-values to filter highly discriminative statistical features and combining multi-layer CNN networks to process directional sequences, the strategy reduces computational overhead while ensuring identification performance, supporting real-time detection deployment. (3) A Transformer-based adversarial network module is introduced to capture long-range dependencies of features through a self-attention

mechanism, optimizing parameter robustness and effectively reducing noise interference and the confusion of similar devices. (4) Experimental validation based on three public datasets (UNSW2016, Yourthings, and CICIoT22) demonstrates that the CAGLite algorithm achieves an F1-score of 98.99%, a 2.85% improvement over existing state-of-the-art algorithms, while maintaining low computational complexity, providing an efficient and feasible solution for IoT device security management.

2 相关工作

Current identification methods for Internet of Things (IoT) devices are categorized into hardware/driver-based identification and traffic-based identification. Since IoT devices produced by the same manufacturer generally utilize similar hardware or internal drivers, the former approach has gradually become obsolete. At present, traffic-based identification methods have become more prevalent.

The AuDI algorithm developed by Marchal et al. is based on periodic communication traffic. Aksoy et al. researched the SysID algorithm, which utilizes genetic algorithms for feature selection and classifies devices through single-packet analysis, making it suitable for encrypted traffic. Chowdhury et al. investigated device fingerprinting methods based on network and transport layer features using various machine learning algorithms. The IoT ETEI algorithm developed by Yin et al. employs an end-to-end CNN+BiLSTM model to directly learn spatio-temporal features from raw traffic.

Cao et al. utilized TextCNN and BiLSTM_{Attention} in parallel to extract local and global features of application-layer service information, which are then weighted, fused, and fed into an MLP for classification. Song et al. collected traffic from the device startup phase and employed a two-stage approach. The first stage uses a one-vs-rest SVM; if one classifier returns positive, the device type is determined immediately. If multiple classifiers return positive, the process enters a second stage using an improved cosine similarity for secondary classification. Finally, the cosine similarity between the device to be identified and the parameters of each category is calculated, with the maximum value determining the final result. This algorithm offers good scalability as it only requires adding a corresponding binary classifier when a new device type is introduced, without needing to rebuild the model. Zhang et al. proposed an online IoT device identification method based on the Self-Organizing Incremental Neural Network (SOINN). They utilized the ARE method to collect and label network data as training data for brand and model identification. Subsequently, they trained an SVM classifier using features based on the page DOM tree structure for brand identification, employed regular expression matching for model identification, and calculated TF-IDF weights to generate a model feature library.

Liu et al. achieved model identification by monitoring 802.11 data frame traffic from smart home devices in a WiFi environment, extracting features, and utilizing an improved CART decision tree algorithm. Hao et al. proposed a two-

stage algorithmic framework: the first stage uses a Bi-LSTM neural network to extract fingerprints, and the second stage employs the Jaro-Winkler text similarity matching algorithm to identify models and device types, establishing a model knowledge base for identification through matching. Li et al. proposed an LCNN-LSTM structure, where the LCNN extracts spatial features of traffic and the LSTM captures temporal dependencies; this dual-feature identification enhances robustness. Zheng et al. proposed a deep machine learning algorithm based on LSTM, training a model with collected device information to perform identification. Zou et al. proposed an IoT device identification method based on a federated learning architecture, decomposing the task into local feature extraction on edge devices and global knowledge aggregation on a central server. The HSGAN-IoT model proposed by Jin et al. is a deep learning-based IoT device classification method that innovatively combines hierarchical concepts with semi-supervised Generative Adversarial Networks (GANs).

The HSGAN-IoT model designs a two-stage classification framework: the first-layer discriminator filters IoT devices from raw traffic, while the second-layer discriminator performs fine-grained classification of device types, functions, and manufacturers. HSGAN-IoT introduces a multi-classification head structure, decoupling the discriminator' s real/fake discrimination task from the classification task, and optimizes them via a joint loss function. Rimmer et al. proposed an automated website fingerprinting attack method based on deep learning, automating the feature learning process and designing three deep learning models for website identification.

This study constructed the largest website fingerprinting dataset to date and proposed an end-to-end classification framework. First, raw traffic is converted into Tor cell sequences as input, after which a deep network automatically extracts traffic features for website classification. The research introduced semi-automated hyperparameter tuning and parallel training strategies to improve model adaptability and efficiency. Guan et al. proposed a model named BAPM (Block Attention Profiling Model) targeting multi-label browsing behavior on the Tor network. The model combines Block Division with a Multi-head Attention mechanism to design an end-to-end attack framework: first, a convolutional neural network transforms the raw directional sequence into a label-aware representation; next, block division segments the mixed data to mitigate information confusion; finally, the attention mechanism dynamically aggregates blocks belonging to the same webpage label to identify multiple websites simultaneously from a global perspective.

Sirinam et al. proposed a model called Deep Fingerprinting (DF), a deep learning-based website fingerprinting attack method designed for high-precision identification of encrypted traffic on the Tor network. This study combines Convolutional Neural Networks (CNN) with modern deep learning architectural designs to create an end-to-end attack framework. It first transforms raw traffic direction sequences into fixed-length vector inputs, then automatically extracts robust features through multi-block convolutional layers, batch normalization,

and dropout mechanisms, and finally performs website classification via fully connected layers. Sirinam et al. also proposed an algorithm called Triplet Fingerprinting, which utilizes triplet networks to achieve N-shot learning, significantly enhancing the practicality and portability of the attack. Bhat et al. proposed a new website fingerprinting attack method called Var-CNN; its core idea is to combine manually extracted cumulative features with automatically extracted deep features to form a semi-automatic feature extraction model. In addition to deep learning networks, there are also algorithms that utilize Transformers for identification. Deng et al. proposed a new IoT device identification algorithm based on Transformers and clustering. The process mainly involves converting network traffic into multi-byte token sequences as input to a Transformer model to extract temporal features; using angular margin loss to optimize the feature space and improve inter-class separation; generating multiple prototype vectors for each device class through HDBSCAN clustering; and finally achieving known device classification and unknown device detection through cosine similarity matching. This method can complete identification with only 1 minute of traffic, achieves over 99.6% accuracy on public datasets, and supports unknown device detection. However, its identification performance is limited for devices with few training samples, devices of the same brand are easily confused, the model complexity is high, and it does not support incremental learning. Luo et al. utilized Transformers to extract temporal features from normal traffic for device classification, improving identification accuracy through a majority voting ensemble algorithm.

The advantage of this method lies in its ability to eliminate abnormal traffic during device identification, while its disadvantages include the need to accumulate traffic over a long period to achieve high precision and the potential for confusion between similar devices. Jin et al. proposed an algorithm for identifying website fingerprints under multi-label browsing behavior in the Tor network. This model transforms the multi-label website fingerprinting problem into an ordered set prediction task. By introducing a learnable "label query" mechanism, the model can adaptively separate and identify traces of different webpages from mixed traffic without prior knowledge of the number of labels. The HSGAN-IoT algorithm was the first to apply multi-head attention adversarial networks to IoT device identification, achieving excellent results. However, the HSGAN-IoT algorithm requires 1400-byte packet header and payload sequences, which consumes significant resources and is unfavorable for large-scale deployment.

3 CAGLite 算法

This paper proposes CAGLite (Cnn and ANova attributes Fusion), an algorithm for device identification based on network traffic characteristics. The CAGLite algorithm identifies devices using network flow records generated by the Internet of Things (IoT). Unlike other device identification algorithms, CAGLite utilizes four distinct attributes to identify devices: packet direction sequences, statistical features of flow records, the first 100 bytes of all packet headers, and the first 100

bytes of all packet payloads. The packet direction sequences and flow record statistical features can be obtained rapidly without scanning packet content, ensuring they are not restricted by encrypted traffic. [Figure 1: see original paper] illustrates the complete workflow of the CAGLite algorithm.

As shown in [Figure 1: see original paper], the CAGLite algorithm first extracts traffic from the raw network data based on the five-tuple. For each flow, it extracts traffic features, direction sequences, packet header byte arrays, and packet payload arrays. Subsequently, different networks are employed to perform feature extraction for these various attributes, followed by feature fusion through a fully connected network. To improve the accuracy of device identification, the CAGLite algorithm also incorporates a Transformer-based adversarial network (the same adversarial network architecture used in the HSGAN-IoT algorithm).

3.1 网络流量处理

During the network traffic processing stage, raw data packets are first acquired and filtered to eliminate invalid packets. The remaining valid packets are then partitioned into flows based on their five-tuple (source/destination IP addresses, source/destination ports, and protocol). Subsequently, these flows undergo further filtering to remove short flows and anomalous traffic.

Raw network data can be provided either as stored network traffic data files in PCAP format or as real-time packet streams captured from network interface cards or routers. For each packet, CAGLite first extracts the five-tuple and then performs normalization on the five-tuple at line 5.

The five-tuple is normalized because the CAGLite algorithm requires bidirectional flows, where a single network flow contains packets from both directions. During normalization, the IP addresses and ports within the packet can be sorted based on internal/external IP ranges or the lexicographical order of the IP strings. The normalized five-tuple serves as the flow key, which is used to retrieve specific flows from the partitioned flow set.

While scanning each packet, CAGLite records the packet' s direction into the direction sequence of the corresponding flow; however, it does not immediately calculate the statistical features of that flow. This is because the statistical features of a flow depend on all the packets it contains, making it impossible to calculate them at this stage.

After scanning each packet, the flow preprocessing algorithm detects whether the flow has terminated. When a flow termination condition is triggered (for example, upon receiving a "FIN" packet or a packet timeout), CAGLite begins outputting the flow' s attribute features. First, short flows are filtered based on the parameter β. Short flows typically consist of a small number of packets and may not fully reflect the communication patterns and application characteristics of a device. For instance, the communication patterns of IoT devices during normal operation may exhibit specific characteristics such as packet sequences

and size distributions. These features are not prominent in short flows, which can interfere with the accurate analysis of device communication behavior and thus reduce the accuracy of device identification.

Furthermore, short flows are often generated by brief, atypical operations of a device and may be unrelated to the device' s primary functions or applications. For example, some IoT devices may briefly send or receive a small amount of data during startup; these short flows differ significantly from the traffic characteristics of the device during normal operation. If not filtered, these short flows might be misidentified as characteristics of other devices, increasing the risk of false identification. Filtering out short flows avoids the impact of these anomalies on device identification results and enhances the robustness of the identification process.

For certain IoT devices that produce only short flows, the traffic patterns are limited and the packet count is low; in such cases, Deep Packet Inspection (DPI) methods can be used to analyze the payload of each packet to determine the device type. For long flows, however, using DPI to analyze every packet would consume excessive computational resources, making real-time device detection impossible. This study focuses on lightweight detection for IoT devices with long flows, and therefore filters out short flows first. CAGLite calculates the attribute list for a flow based on the attribute names specified in the configuration and then outputs the flow' s direction sequence and statistical features. While many attributes can be extracted from network traffic, including every attribute in the list would consume significant computational resources. To improve traffic processing speed, CAGLite filters traffic attributes, selecting only those with high importance for device identification to be added to the attribute list. Once the statistical attributes of the flow are calculated, the flow data can be output.

Data preprocessing does not require waiting for all packets to be scanned before outputting flow attributes; instead, it outputs attributes such as the direction sequence and statistical features immediately upon the termination of each individual flow. This enables CAGLite to process an infinite stream of packets in real-time and allows it to be deployed at gateways for real-time flow extraction.

3.2 特征提取

After obtaining network traffic, the process of extracting device traffic features can begin. Feature extraction is divided into two parts: extracting packet direction sequences and extracting traffic statistics. Packet direction sequences capture the sequential communication relationships between IoT devices. Different types of devices often exhibit distinct patterns in the number and order of packets sent and received. For instance, sensor-based devices typically transmit collected data to a server at specific intervals; their packet direction sequences primarily manifest as a unidirectional transmission pattern from the device to the server with a regular frequency. In contrast, actuator devices primarily receive control commands from a server, resulting in sequences dominated by

unidirectional reception from the server to the device.

Directly inputting packet direction sequences extracted from network traffic into a classification algorithm for IoT device identification yields suboptimal results. This is due to the limited and inconsistent lengths of direction sequences across different traffic flows, which prevents classification algorithms from being fully utilized. To address this, this paper first inputs the direction sequences into a multi-layer Convolutional Neural Network (CNN) for analysis to derive high-dimensional feature vectors. Furthermore, processing the sequences through the CNN ensures they are of uniform length, which effectively enhances identification performance.

However, relying solely on direction sequences does not capture the full spectrum of traffic characteristics. Consequently, the CAGLite algorithm also incorporates traffic statistical features. Network traffic contains dozens of statistical attributes; for example, tools like CICFlowmeter can extract over 80 types of transport-layer traffic statistics. These features include flow duration, packet count, packet size, and bit rate, categorized by forward and backward directions. The importance of these attributes varies depending on the specific network traffic application. Utilizing low-importance features can even degrade the accuracy of device identification. It is essential to identify features that exhibit significant differences across different categories, as these are more valuable for IoT device identification. The ANOVA F-value serves as a critical metric for measuring the significance of such differences. Analysis of Variance (ANOVA) is a statistical method used to test whether there are significant differences between the means of multiple populations.

CAGLite utilizes the ANOVA F-value to select the optimal features. In network traffic-based IoT device identification algorithms, the ANOVA F-statistic serves as an efficient feature selection method. Its core principle lies in quantifying the inter-group differences of traffic attributes across various IoT device categories. When using ANOVA to select IoT traffic features, the Sum of Squares Within groups (SSW) and the Sum of Squares Between groups (SSB) are first calculated. Assuming there are k different devices, where the i-th device has n_i different observations, $\bar{x}_i$ represents the mean of the i-th device category, and $\bar{x}$ represents the overall mean. SSW and SSB can be calculated according to the following formulas:

SSW = åå (xij - xi) 2

i =1 j =1

SSB = å ni (xi - x) 2

The F-value can then be calculated using Equation (3). Subsequently, significant traffic statistical features can be identified by performing a statistical hypothesis test based on the calculated F-value.

SSB /(k - 1) SSW /(N - k)

In Equation (3), N represents the total number of traffic flows. Due to functional differences among IoT devices—such as the contrast between high-bandwidth camera streaming and short-packet sensor transmissions—their traffic attributes (e.g., packet length distribution and transmission cycles) exhibit inherent variations. ANOVA directly quantifies the dispersion of attribute means across different device categories, which aligns precisely with the objectives of the identification task.

The ANOVA F-value reflects the intrinsic characteristics of the dataset. For different datasets, the F-values of various attribute features vary according to the specific device traffic patterns. After sorting the features based on their F-values, we calculate the relative contribution and cumulative contribution rate of each feature.

The relative contribution of the i-th feature can be calculated according to the following formula:

ri = m i

In Equation (4), r_i represents the relative contribution of the i-th feature to the classification, F_i denotes the ANOVA F-value of the i-th feature, and m is the total number of features. The formulas for calculating the cumulative contribution of the top n features and the feature selection criteria are as follows:

Cn = å r(i)

Cn ^{3} t

Here, C_n represents the cumulative contribution of the top n ranked features, where θ is a preset threshold (set to 0.9 in this study).

By setting a cumulative contribution rate threshold, the system automatically selects the top-ranked features until their cumulative contribution reaches or exceeds this value. This method ensures that the selected feature subset retains the vast majority of the discriminative classification information from the original feature set while achieving significant dimensionality reduction. For example, while the original feature set may contain dozens of features, screening via cumulative distribution allows for the retention of only about half of the key features while still covering over 90% of the classification capability.

This screening mechanism based on cumulative distribution offers multiple advantages. First, it reduces model complexity and the risk of overfitting by eliminating redundant features with weak discriminative power. Second, the retained features possess clear physical significance—such as total packet count, flow direction ratio, and packet interval statistics—which enhances the interpretability of the model. Finally, this method adaptively determines the size of the feature subset, avoiding the need for manual specification of the number of features. After calculating the ANOVA F-values for the IoT experimental datasets UNSW2016, Yourthings, and CICIoT22, the features were ranked from

highest to lowest according to their F-values; the results are shown in [Figure 2: see original paper] through [Figure 4: see original paper].

In [Figure 2: see original paper] to [Figure 4: see original paper], the vertical axis represents the selected attributes, arranged in descending order of importance. Each traffic attribute is identified by a number and a name, separated by a colon. As can be seen from [Figure 2: see original paper] to [Figure 4: see original paper], the importance of different attributes varies across different IoT traffic environments. For instance, the "Bwd Init Win Bytes" attribute in [Figure 2: see original paper]—which statistics the window size of the last packet in the backward flow—plays a crucial role in distinguishing IoT devices within the UNSW2016 traffic. However, in other IoT traffic datasets, the importance of "Bwd Init Win Bytes" decreases, and it does not even appear among the top 10 most important attributes for the Yourthings traffic.

Furthermore, regarding the distribution of attribute importance, the Yourthings traffic exhibits a relatively even distribution. In contrast, for the UNSW2016 and CICIoT22 traffic, the importance of the primary attribute is significantly higher than that of the others. This indicates that the three experimental datasets possess distinct characteristics, which can be utilized to analyze the performance of algorithms under different traffic patterns.

Next, we calculate the contribution of each attribute value and perform accumulation. In the UNSW2016 dataset, when accumulating up to "Flow IAT Min," the cumulative contribution of the first 36 feature attributes (including "Flow IAT Min") exceeds 90%; therefore, the top 36 feature attributes are selected for subsequent processing. In the Yourthings dataset, the cumulative contribution of the first 44 feature attributes reaches 90%. In the CICIoT22 dataset, the cumulative contribution of the first 37 feature attributes reaches 90%. These data are presented in .

Number of attributes when cumulative contribution $> 90\%$

Cumulative contribution value

CICIoT22

90.53%

UNSW2016 Yourthings

90.04% 90.22%

3.3 多层全连接网络分类器训练与识别

The attributes filtered by the ANOVA F-test algorithm are considered low-dimensional raw features. To prevent overfitting and extract high-dimensional features from statistical properties for device identification, the CAGLite algorithm first performs a feature mapping of the selected traffic features through

a fully connected network. This process maps statistical features into a high-dimensional space to obtain a statistical attribute feature vector. The neural network architecture used by CAGLite for feature extraction and fusion is illustrated in [Figure 6: see original paper].

In the CAGLite algorithm, the attributes extracted from each network flow consist of four components: flow statistical attributes, packet direction sequences, packet header bytes, and packet payload bytes. Different neural networks are employed to extract features from these distinct network traffic attributes. [Figure 6: see original paper] details the structure of each respective network.

As shown in [Figure 6: see original paper], the number of flow statistical features is relatively small (with a maximum of 44 selected attributes); therefore, a fully connected network is used to map these features into a high-dimensional space. Within the same network environment, the number of statistical attributes per flow is fixed. However, the number of packets contained in each flow varies, leading to inconsistent lengths in the direction sequences. To improve processing efficiency, the CAGLite algorithm fixes the direction sequence length to n.

When the number of packets exceeds n, only the first n direction sequences are retained; if the number is less than n, the sequence is padded via replication. It is important to note that for flows with direction sequences shorter than n, the CAGLite algorithm does not simply pad the remaining slots with zeros. Instead, it replicates the preceding direction sequences. This approach effectively preserves the original feature distribution of the direction sequence and avoids the bias introduced by zero-padding, which could otherwise interfere with the classifier' s discriminative ability. For example, if a flow has a direction sequence length of 3 and the preset length is $n = 5$, the algorithm replicates the first two elements of the sequence to complete it, rather than filling it with invalid zeros. This dynamic padding strategy ensures uniform input dimensions while maximizing the retention of the device' s original communication patterns. Experimental results indicate that CANFusion achieves high accuracy when n is set to 200.

To improve the accuracy of the algorithm, the first 100 bytes of all packet headers and the first 100 bytes of all packet payloads are incorporated alongside the packet direction sequences and flow statistical feature vectors for device identification. The first 100 bytes of the packet header serve to extract protocol stack metadata, while the first 100 bytes of the payload characterize application-layer behavior. The header portion captures structured protocol information from the network and transport layers, such as IP addresses, port numbers, protocol identifiers, and TCP flags. However, since IP and MAC addresses do not represent the intrinsic behavior of device traffic, this information is removed during data preprocessing; to ensure that packet length remains unaffected, these fields are padded with zeros. The payload portion contains the actual content of application-layer protocols, such as User-Agent and Host fields in HTTP headers, TLS handshake information, DNS query details, or command structures of specific IoT protocols. These high-level features reveal device ap-

plication behavior patterns, manufacturer information, service interaction logic, and firmware characteristics. For the processing of packet header bytes and payload bytes, CAGLite employs a segmented Convolutional Neural Network (CNN) architecture and a BiLSTM, respectively.

The combination of these elements forms a comprehensive feature set covering everything from low-level communication protocols to high-level application behaviors. This allows the algorithm to identify device names more accurately and significantly enhances the uniqueness and recognition accuracy of device fingerprints.

To fully utilize these four categories of features, CAGLite concatenates the feature vectors extracted from the different units and feeds them into a fully connected neural network for device identification. Through the fusion of these four feature types, CAGLite achieves higher recognition accuracy than existing algorithms.

3.4 基于 Transformer 的对抗网络

To address the issues of noise interference and feature confusion present in IoT device traffic data, the CAGLite algorithm introduces a Transformer-based

对抗网络模块，该模块通过自注意力机制增强了关键特征的表示能力。该模块采用编码器-解码器架构。编码器由两个堆叠的 Transformer 层组成，每层包含八个注意力头，通过多头注意力机制捕捉流量特征之间的长程依赖关系。解码器组件被设计为生成对抗网络（GAN）的判别器，与编码器形成对抗训练机制。判别器利用全连接网络进行二分类。

simulated traffic attributes of the same length (comprising 200 packet directions, 100-byte headers, and 100-byte payloads). The network first maps low-dimensional inputs to a high-dimensional feature space through a linear layer and a ReLU activation function. It then performs deep feature extraction via two encoding blocks containing multi-head attention mechanisms and feed-forward networks. Each encoding block incorporates residual connections, layer normalization, and dropout regularization to ensure training stability and generalization capability. Finally, a linear layer transforms the encoded features into a multi-dimensional vector of simulated traffic attributes. By utilizing the Transformer network, high-dimensional and semantically rich deep features can be extracted from simulated IoT network traffic data and condensed into structured traffic attribute vectors. This provides reliable feature support for subsequent tasks such as traffic classification, anomaly detection, and device identification.

For the simulated traffic attributes, the feature extraction network described in Section 3.3 is also used to extract feature vectors of the same length as the real traffic, followed by a binary classification against the real traffic data. Based on the classification results, the parameters within the adversarial network and the feature extraction network are updated. During the adversarial training process, the objective of the generator (encoder) is to produce device feature

representations that are difficult for the discriminator to distinguish, while the discriminator must accurately identify the source of the features. By alternately optimizing the parameters of both components, the feature vectors output by the encoder eventually retain device-specific information while achieving robustness against noise.

4 实验与分析

To validate the performance of the CAGLite algorithm in IoT device identification, this study utilizes three public datasets—UNSW2016, Yourthings, and CICIoT22—to conduct a comparative analysis against several existing IoT device identification algorithms. These include AWF, BAPM, TF, TMWF, DF, TikTok, TBD, VarCNN, DITC, HSGAN, and CAGLite.

4.1 实验数据

The dataset used in this experiment encompasses a wide variety of IoT scenarios. The input data consists of raw PCAP files. In this study, network data are partitioned into flow records based on the five-tuple (source/destination IP addresses, source/destination ports, and protocol), and IoT device identification is subsequently performed using these flow records as the basic unit. To evaluate and compare the accuracy of different algorithms, the three datasets were divided into training, validation, and testing sets according to a ratio of 70%, 10%, and 20%, respectively. The model was trained for 30 epochs with a batch size of 256. We employed the Adam optimizer with an initial learning rate of 2×10^{-4}. The experimental environment was Ubuntu 22.04, utilizing a single RTX-3060 GPU with 12GB of video memory.

To perform effective IoT device identification, a sufficient number of packets must be available to extract the network traffic characteristics of the device. Consequently, we filtered the traffic length in the experimental data and selected only those flows containing more than 20 packets.

Furthermore, some devices in the experimental data generated too few flows to be reliably identified. For instance, in the CICIoT22 dataset, the "D-Link DCHS-161 Water Sensor" produced only a single flow, whereas all other devices generated at least 853 flows. To enable the algorithm to better learn the characteristics of IoT traffic, we removed devices with fewer than 100 flows from the experimental data. Additionally, the dataset was labeled according to device names. Taking the CICIoT22 dataset as an example, the distribution of device names is illustrated in [Figure 8: see original paper].

The three devices with the highest number of flows are the Amazon Alexa Echo Dot, Amazon Alexa Echo Studio, and Philips Hue Bridge. (In the dataset, there are two Amazon Alexa Echo Dot devices: Amazon Alexa Echo Dot 1 and Amazon Alexa Echo Dot 2. Since the device names are identical, these two devices were merged during the experiment.) The first two are smart speakers

under the Amazon brand, capable of voice interaction and smart home control. The third is a bridge for smart devices used to control Hue-compatible hardware.

As indicated by the data distribution, the datasets used in this paper cover a diverse range of device types and applications, providing a robust basis for verifying the effectiveness of IoT device classification algorithms.

4.2 算法准确率分析

To evaluate the performance of the algorithms more comprehensively, four evaluation metrics were employed in the analysis: Accuracy, Precision, Recall, and the F1-score. The calculation formulas for these metrics are as follows:

Accuracy =

(TPi + FN i)

Pr ecision =

1 N

å Pr ecisioni N i =1

Re call =

Re calli

Each experimental dataset contains multiple devices, constituting a multi-class classification problem. In equations (7) through (10), N represents the number of devices within each respective dataset.

Accuracy refers to the proportion of correct predictions among all predictions made. In equation (7), TP_i denotes the number of instances where the i-th device was correctly identified, while FN_i represents the number of instances where the i-th device was incorrectly identified as another device. The term $TP_i + FN_i$ represents the total number of samples for device i in the dataset. In this experiment, the overall Precision, Recall, and F1-score are calculated as the average of the Precision, Recall, and F1-score values for each individual device. The calculation formulas for $Precision_i$, $Recall_i$, and $F1_i$ in equations (8) through (10) are as follows.

TPi + FPi

TPi + FN i

Pr ecisioni ´ Re calli Pr ecisioni + Re calli

Pr ecisioni = Re calli = F1i = 2 *

In Equation (11), FP_i represents the number of other devices incorrectly predicted as device i. The $F1_i$ score incorporates both $Precision_i$ and $Recall_i$, providing a more comprehensive measure of the identification performance for

device i. A higher $F1$ value indicates superior classification performance of the algorithm.

In this study, we compare the CAGLite algorithm with several existing methods, including AWF, BAPM, TF, TMWF, DF, TikTok, TBD, VarCNN, DITC, and HSGAN.

As illustrated in [Figure 9: see original paper], the CAGLite algorithm achieved the highest performance across all metrics on the three datasets, demonstrating its exceptional cross-dataset generalization capability and consistent technical advantages. Notably, the HSGAN algorithm, which is based on Generative Adversarial Network (GAN) features, also exhibited high accuracy across various metrics. However, while HSGAN utilizes the first 650 bytes of the packet header and the first 750 bytes of the payload for device identification, CAGLite operates on a significantly smaller data scale. Specifically, CAGLite uses the first 100 bytes of the packet header, the first 100 bytes of the payload, a 200/8-byte packet direction sequence, and $n \times 4$ bytes of traffic attributes (where n varies by dataset), totaling approximately 550 bytes. Consequently, CAGLite requires substantially fewer hardware resources and offers faster computation speeds while outperforming HSGAN across all evaluated metrics.

BAPM[22] TF[24] TMWF[28] DF[23] TikTok[19] TBD[27] VarCNN[25] DITC[26] HSGAN[20] CAGLite

Accuracy

Precision

Recall

F1_{score}

The average data for accuracy, precision, recall, and F1-score are presented. As shown in the tabular data, the performance of the eleven device identification algorithms exhibits a distinct gradient distribution. The CAGLite algorithm achieves the best performance across all evaluation metrics, reaching an accuracy of 99.18%, with precision, recall, and F1-score all exceeding 0.989. This performance level approaches near-perfect identification, placing CAGLite in the first tier of performance.

The HSGAN and DITC algorithms rank second and third with accuracies of 97.57% and 94.39%, respectively. Although a performance gap exists between these and CAGLite, they still significantly outperform the remaining algorithms. The table further illustrates a positive correlation between algorithm performance and complexity; specifically, algorithms employing Generative Adversarial Networks (GANs), fusion strategies, or deep ensemble methods generally outperform traditional convolutional neural networks and basic fingerprinting models. These comparison results clearly demonstrate the varying effectiveness of different algorithms in device identification tasks. Based on the average values of the four metrics, the CAGLite algorithm consistently outperforms all

other models, further validating its superiority in identifying device names.

4.3 混淆矩阵分析

To analyze which devices are prone to misclassification during IoT device identification, this study conducts a confusion matrix analysis using CAGLite across three datasets. The confusion matrix for the CAGLite algorithm on the CICIoT22 dataset is shown in Figure 10 [Figure 10: see original paper].

As illustrated in Figure 10, confusion within the CICIoT22 dataset primarily occurs among device groups with highly similar functions or brands. For instance, mutual misclassifications exist between Amazon' s Echo Dot, Echo Spot, and Echo Studio smart speakers. The Google Nest Mini is occasionally misidentified as a Smart Board, as both serve as voice interaction hubs. Furthermore, the Atomi smart coffee maker is confused with smart plugs from brands like Teckin and Yutron, likely because they all exhibit intermittent network behavior characterized by scheduled power cycling.

As shown in Figure 11 [Figure 11: see original paper], the CAGLite algorithm achieves zero misclassifications for devices such as the Amazon Echo and Samsung SmartCam on the UNSW2016 dataset. However, significant unidirectional confusion exists between the Belkin Wemo smart switch and motion sensor, with as many as 14 switch samples misclassified as sensors. This highlights the inherent identification challenges posed by highly similar network behavior patterns among devices of the same brand and protocol family. Additionally, one Netatmo weather station sample was misclassified as a Welcome camera from the same brand, and isolated instances of confusion occurred between Insteon and TP-Link cameras. These findings further confirm that similarities in device functionality or branding are the core reasons for model errors.

As demonstrated in Figure 12 [Figure 12: see original paper], the classification model performs well across most device categories in the Yourthings dataset, maintaining high overall predictive accuracy. Several devices, including the AppleTV, BelkinWeMoCrockpot, LIFXVirtualBulb, and BoseSoundTouch10, were classified without any errors. Nevertheless, some distinct instances of confusion remain; for example, mutual misclassifications occur between the GoogleHome and GoogleHomeMini. This suggests that these two device types possess similar features, making it difficult for the model to distinguish between them clearly. Overall, while the model demonstrates robust device identification capabilities, there remains room for improvement in differentiating sub-categories or devices with overlapping functions.

Based on its performance across the three datasets, the CAGLite algorithm demonstrates high classification and identification capabilities, particularly exhibiting superior accuracy and robustness for mainstream IoT devices with distinct features. However, in certain scenarios—such as when different devices utilize the same underlying chipsets, communication protocols, or nearly identical network behavior patterns—CAGLite still struggles to achieve perfect differ-

entiation. This remains a challenge that current mainstream algorithms have yet to fully resolve. Future optimizations should focus on capturing and amplifying subtle but critical differences between these similar devices, potentially enhancing model discriminative power through fine-grained traffic analysis or the integration of hardware-specific characteristics.

5 结束语

The CAGLite algorithm proposed in this paper achieves efficient identification of diverse IoT device types by innovatively combining packet direction sequences, statistical traffic features, the first 100 bytes of packet headers, and the first 100 bytes of packet payloads. Packet direction sequences and statistical traffic features represent two distinct categories of network traffic characteristics, each playing a unique role in the identification of Internet of Things (IoT) devices. By leveraging deep neural networks and feature fusion techniques, the CAGLite algorithm achieves superior performance in IoT device identification. Experimental results demonstrate that CAGLite performs exceptionally well in device name identification tasks, significantly outperforming existing algorithms. While the effectiveness of the CAGLite algorithm depends on thorough feature analysis and the rational selection of characteristics from the dataset, the traffic patterns of new devices often exhibit variations. These differences can render original features inapplicable, thereby affecting the algorithm' s performance in identifying novel devices. Consequently, our future research will focus on emerging device types and the development of incremental learning algorithms tailored for IoT device identification.

References: [1] LU X L, LI Z H. Internet of Things Device Identification Method Based on Multi-Modal IoT Device Fingerprint and Ensemble Learning [J]. Computer Science, 2024, 51(09): 371-382. [2] FAN L N, LI C L, WU Y C, et al. A Survey on Internet of Things Device Identification and Anomaly Detection [J]. Journal of Software, 2024, 35(01): 288-308.

[3] YANG W C, GUO Y B, LI T, et al. Internet of Things Device Identification Method Based on Traffic Fingerprint and Internet of Things Security Model [J]. Computer Science, 2020, 47(07): 299-306. [4] MARCHAL S, MIETTINEN M, NGUYEN T, et al. AuDI: toward autonomous IoT device-type identification using periodic communication [J]. IEEE Journal on Selected Areas in Communications, 2019, 37(6). [5] AKSOY A, GUNES M. Automated IoT device identification using network traffic [C]//Proceedings of the 2019 IEEE International Conference on Communications, 2019: 1-7. [6] CHOWDHURY R R, CHE I A, ABAS P E. Internet of things device classification using transport and

network layers communication traffic traces [J]. International Journal of Computing and Digital Systems, 2022, 12(1): 545-555. [7] YIN F, YANG L, WANG Y, et al. IoT ETEI: end-to-end IoT device identification method [C]//Proceedings of the 2021 IEEE Conference on Dependable and Secure

Computing, 2021: 1-8. [8] CAO W K, LIN H G. Internet of Things Device Identification Method Based on Weighted Feature Fusion [J]. Computer Science, 2024, 51(S2): 885-893.

[9] SONG Y B, QI X Y, HUANG Q, et al. Internet of Things Device Identification Algorithm Based on Two-Stage Multi-Classification [J]. Tsinghua Science and Technology, 2020, 60(05): 365-370. [10] ZHANG S S, HUANG J, QI C Y, et al. Online Internet of Things Device Identification Method Based on SOINN [J]. Journal of Southeast University (Natural Science Edition), 2021, 51(04): 715-723.

[11] LIU D D, HAN Y, LIU X Y, et al. Smart Home Identification Method Based on WiFi Data Frame Features [J]. Computer Engineering and Applications, 2023, 59(15): 274-280. [12] HAO Q X, RONG Z, XIE L J, et al. Online Internet of Things Device Identification Method Based on Bi-LSTM [J]. Journal of Xi' an University of Science and Technology, 2023, 43(02): 422-430. [13] LI Z H, WANG Z H. Internet of Things Device Identification Method Based on LCNN and LSTM Hybrid Structure [J]. Information Network Security, 2023, 23(06): 43-54.

[14] ZHENG Y, TIAN H, MA Q. Wireless Internet of Things Device Identification Method Based on LSTM Model [J]. Radio Communication Technology, 2024, 50(03): 597-602. [15] ZOU X X, ZHOU Z R, WANG H L, et al. Distributed Internet of Things Device Identification Method Based on Federated Learning [J]. Computer Engineering and Applications, 2024, 60(23): 155-167. [16] SIVANATHAN A, HABIBI GHARAKHEILI H, LOI F, et al. Classifying IoT Devices in Smart Environments Using Network Traffic Characteristics [J]. IEEE Transactions on Mobile Computing, Aug.

[17] ALRAWI O, LEVER C, ANTONAKAKIS M,et al. SoK: Security Evaluation of Home-Based IoT Deployments[J], in IEEE Symposium of Security & Privacy, May 2019. [18] DADKHAH S, MAHDIKHANI H,DANSO P, et al, Towards the development of a realistic multidimensional IoT profiling dataset[C]//2022 19th annual international conference on privacy, security & trust (PST). New York: IEEE Computer Society, 2022:1-11. [19] RAHMAN M S, SIRINAM P, MATTHEWS N,et al. Tik-tok: the utility of packet timing in website fingerprinting attacks[J]. Proceedings on Privacy Enhancing Technologies, 2020, 2020(3):5-24. [20] JIN Y L, ZHOU J H, GAO Y. HSGAN-IoT: A hierarchical semi-supervised generative adversarial network for IoT device classification[J]. Computer Networks, 2024, 243: 110299. [21] RIMMER V, PREUVENEERS D, JUAREZ M, et al. Automated Website Fingerprinting through Deep Learning [C]// Proc of the Network and Distributed Systems Security (NDSS) Symposium 2018. San Diego, CA: IEEE Computer Society, 2018: 1-15. [22] GUAN Z, XIONG G, GOU G, et al. BAPM: Block Attention Profiling Model for Multi-tab Website Fingerprinting Attacks on Tor[C]// Proc of the 37th Annual Computer Security Applications Conference (ACSAC ' 21). New York, NY: ACM, 2021: 248-259. [23] SIRINAM P, IMANI M, JUAREZ M, et al. Deep Fingerprinting: Undermining Website Fingerprinting

Defenses with Deep Learning[C]// Proc of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS ' 18). New York, NY: ACM, 2018: 1928-1943. [24] SIRINAM P, MATHEWS N, RAHMAN M S, et al. Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-shot Learning[C]// Proc of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS ' 19). New York, NY: ACM, 2019: 1131-1148. [25] BHAT, S., LU, D., KWON, A., & DEVADAS, S. (2019). Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning.[J]Proceedings on Privacy Enhancing Technologies, 2019(4), 292-310.

[26] DENG L T, GU D L, LIN Z, IoT device identification method based on transformer and clustering[J], Computer Networks, Volume 273, 2025, 111791, ISSN 1389-1286, [27] LUO Y, CHEN X, GE N, et al.Transformer-based device-type identification in heterogeneous iot traffic[J].

IEEE Internet of Things Journal, 2023, 10(6):5050-5062. [28] JIN Z X,LU T B,LUO S, et al. Transformer-based model for multi-tab website fingerprinting attack[C]//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security.

2023:.

Note: Figure translations are in progress. See original paper for figures.