electronics

MDPI

Article

IoT Device Identification Method Based on Causal Inference

Xingkui Wang [1], Yunhao Wu [2], Dan Yu [2], Yuli Yang [2], Yao Ma [2] and Yongle Chen [2,*]

[1] College of Software, Taiyuan University of Technology, Taiyuan 030024, China
[2] College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China
* Correspondence: chenyongle@tyut.edu.cn

Abstract: With the development of 5G, the number of IoT (Internet of Things) devices connected to the Internet will grow explosively. However, due to the vulnerability of the devices, attackers can launch attacks on the vulnerable IoT devices, causing great impact on the security of the network environment. Fine-grained identification of IoT devices can help network administrators set up appropriate security policies based on the functionality and heterogeneity of the devices, while enabling timely updates and upgrades for devices with security vulnerabilities or the isolation of these dangerous devices. However, most of the existing IoT device identification methods rely on a priori knowledge or expert experience in selecting features, which cannot weigh the identification performance and labor cost. In this paper, we design a fine-grained identification method for IoT devices based on causal inference, which automatically extracts key features in the protocol fields of device communication from the perspective of causality and then classifies key features using a Stacking integrated learning method to achieve high-precision and fine-grained device identification. Through experimental verification, the proposed method achieves 96.3% and 97.7% device model identification accuracy under HTTP/TCP and SSH/TCP protocol clusters.

Keywords: Internet of Things; device identification; causal inference; multi-protocol probe

Citation: Wang, X.; Wu, Y.; Yu, D.; Yang, Y.; Ma, Y.; Chen, Y. IoT Device Identification Method Based on Causal Inference. *Electronics* **2023**, *12*, 2727. https://doi.org/10.3390/electronics12122727

Academic Editors: Shibo He, Huan Zhou, Victor C. M. Leung, Fangyuan Xing and Lei Yang

Received: 30 May 2023
Revised: 13 June 2023
Accepted: 14 June 2023
Published: 19 June 2023

1. Introduction

Many IoT devices are inherently vulnerable, and attackers use the vulnerable devices to access the target network and lurk for the opportunity to launch attacks, thus leading to serious security threats to the target network [1]. IoT device identification technology can help administrators to set up corresponding security policies according to the functionality and heterogeneity of devices, and at the same time, they can update and upgrade the devices with security vulnerabilities or isolate these dangerous devices in time.

The existing IoT device identification technology mainly uses the protocols supported by devices to obtain the communication messages between devices in the network by proactive or passive methods and then analyzes the content features of the messages using statistical or machine learning methods to realize the identification of devices. The passive device identification method needs to obtain communication traffic between network nodes for analysis; however, in the IoT environment, it is difficult to deploy traffic detection software to each network node, and it is not easy to obtain communication traffic between devices.

The proactive device identification method involves sending request data to devices, obtaining a response, and judging the device type according to the content of the returned response message. The existing proactive identification methods are mainly classified into two categories: the first category is the identification method based on the banner in the response message. The message content of the application layer protocol for device communication usually contains device identity information, including the device type, manufacturer, and specific model [2]. The well-known Shodan search engine [3] belongs to this type of method, which identifies devices by monitoring the banner information

generated by various device ports in the application layer protocols, with high identification accuracy but low recall. The second category is fingerprint-based device identification methods. The Nmap [4] is widely used by network security researchers for device identification and system detection. At the beginning of the identification process, 15 sets of probe messages are sent to devices, and the device type is identified by constructing a device fingerprint based on the content features of the returned messages from the TCP/IP protocol. The identification accuracy of the operating system of the device and the device type is fair, but the time consumption of the detection and identification is too large, and the identification accuracy decreases significantly with the increase in the brands and models of IoT devices.

Most of the existing IoT device identification methods rely on a priori knowledge or expert experience in selecting features, which cannot trade-off the identification performance and labor cost. In this paper, we propose an IoT device identification method based on causal inference, which actively establishes a connection with devices and sends a request message for the cross-layer protocol and then obtains a cross-layer response message. The constraint-based causal inference method is used to discover the causal relationship between protocol fields and identification capability, and the key features with a direct causal relationship with device identification are automatically extracted from the causality point of view, and the devices are classified based on the key features, so as to achieve high-precision and fine-grained IoT device identification.

The main contributions of this paper are summarized as follows:

- An IoT device identification method for cross-layer protocol feature fingerprinting is proposed. The key features are automatically extracted using a causal inference method, which better trades off identification accuracy and labor cost.
- The Stacking method is used to reduce the model variance and is experimentally demonstrated to achieve 96.3% and 97.7% identification accuracy under HTTP/TCP and SSH/TCP protocol clusters, respectively.

The remainder of the article is organized as follows: Section 2 presents the work related to device identification as well as causal inference. Section 3 introduces the framework of the device identification method proposed in this paper. Section 4 describes the collection of data and the data pre-processing methods designed in this paper. Section 5 describes the method of selecting key features of devices and the design of the device classifier in this paper. Section 6 shows the experimental evaluation. Section 7 concludes the work in this paper.

2. Related Work

The related work will be discussed separately in terms of causal reasoning and device identification.

2.1. Causal Inference

Researchers have proposed many kinds of causal inference algorithms, which can be divided into two categories: constraint-based methods and scoring-based methods. Spirtes and Clark proposed the PC algorithm [5] and the Fast Causal Inference (FCI) algorithm [6], which are constraint-based causal methods. Constraint-based methods are mainly used for causal inference on data under causal Markov assumptions and fidelity assumptions. Both the PC algorithm and FCI algorithm are based on conditional independent tests, where the data are transformed into a complete undirected graph followed by censored undirected graph edges, and finally, the causal direction is determined. The PC algorithm assumes that there are no latent variables available, and the FCI algorithm can also perform causal learning on potential confounders. Scoring-based methods use a scoring function to evaluate the goodness of a causal network and find a causal network structure that fits the data best according to the observed data. Typical search algorithms include the GES algorithm [7], HC algorithm [8], etc. In addition, the design of scoring methods is also crucial, and typical scoring includes the AIC criterion [9], MDL criterion [10], etc. The

computation time of scoring-based causal inference algorithms is greatly affected by the dimensionality of the data and the amount of data, and its causal inference performance decreases when the number of data dimensions is high.

2.2. Device Identification

Researchers in cyberspace security have carried out research work related to device identification and have achieved a series of research results. The current IoT device identification methods can be broadly divided into two categories: passive identification and proactive identification.

Passive identification: Passive identification mainly focuses on the analysis of traffic characteristics during the normal operation of the device, through which the heterogeneity of the device is reflected, and thus, the device information is identified. The traffic characteristics of devices mainly contain behavioral characteristics of communication traffic as well as the usage of communication protocols and protocol header information, etc. Miettinen et al. [11] focus on device type identification in small networks and reduce the computational overhead of identification. Thangavelu et al. [12], on the other hand, generated the mean variance of protocols such as DNS, TLS, HTTP, etc., through statistical analysis. Marchal et. al [13] listened to the communication behavior of the device, extracted features from it, and combined it with machine learning methods to identify the device. Aneja et al. [14] used a passive identification method to generate device fingerprints, which selected the message reach interval as a feature to generate the inter-arrival time (IAT) graph as a device fingerprint, and the identification accuracy was only 86.7%. The classification model based on traffic features is fast and accurate, but due to the complexity of IoT device categories, the cost of manually collecting and calibrating traffic features is high, and sufficient training data are not available. For similar devices, it is difficult to distinguish their traffic behavioral features, and usually, the granularity of such devices can only reach the device-type level, and fine-grained information such as device model and firmware version is difficult to identify.

Proactive identification: Banner-based device identification is a well-established proactive identification method. The banner-based identification method uses techniques such as regular matching and natural language processing to extract fields such as type, brand, and model directly from the collected text data. ARE [15] uses the Google search engine to expand text information and automatically generate text rules to label and identify IoT devices. IotTracker [16], on the other hand, integrates the DOM tree structure features of device landing page and unstructured text features of FTP ports and achieves different types of device identification by matching text feature libraries. The banner-based identification method can directly and explicitly give the brand and model of the device, but there are also some disadvantages: the banner features are textual information, which has a larger storage overhead compared with numerical features; the identification process relies heavily on the explicit brand and model fields in the banner information, and when the explicit brand and model information does not exist in the banner, the accurate identification cannot be completed; a large text feature comparison library needs to be built, and each time, it is necessary to establish a large text feature comparison library, and each time a device is identified, the entire feature comparison library needs to be traversed, which has a large time overhead. Because banners are added to the descriptions at the manufacturer's discretion, their identification capability and identification applicability are very limited, but because of the simplicity of their acquisition, the more popular IoT device identification engines, such as Shodan [3], Censys [17], and Zoomeye [18], now use this technique.

Prior to this paper, there were also methods using multilayer network protocols for device identification [19], but their methods only considered correlation between fields and ignored causality, which led to the selection of redundant features for device identification. To solve these problems, this paper needs to find an IoT device identification method that can take into account time, labor cost, and feature selection causality.

3. Framework for IoT Device Identification

In this paper, we design a fine-grained device identification method based on causal inference, which can be divided into three stages:

1. **Data collection.** Firstly, based on the normal workflow of IoT devices, the packet sender is used to construct different transport layer and application layer protocol request data messages to send to the target device, and the target device receives the request and returns the transport layer and application layer response messages, and this paper uses the traffic listener to capture these communication messages.
2. **Data preprocessing and feature selection.** Data preprocessing is performed on the collected transport and application layer fields, and the normalization method is used to preprocess for numeric fields, and for text-based fields, the text features are first embedded into n-dimensional vectors using the Doc2Vec [20] method, followed by normalization. After obtaining the normalized data for the causal network between each field, the device identification target is obtained using the PC causal inference algorithm [5], and then, the key features that directly affect the device identification target (device type, device model) are obtained from the causal network.
3. **Device identification.** The key features of the devices are classified to achieve device identification. In this paper, a two-layer Stacking [21] approach is used to combine multiple classifiers in order to reduce the variance of the model and improve the overall identification performance.

Figure 1 illustrates the framework of the causal inference-based approach for IoT device identification.

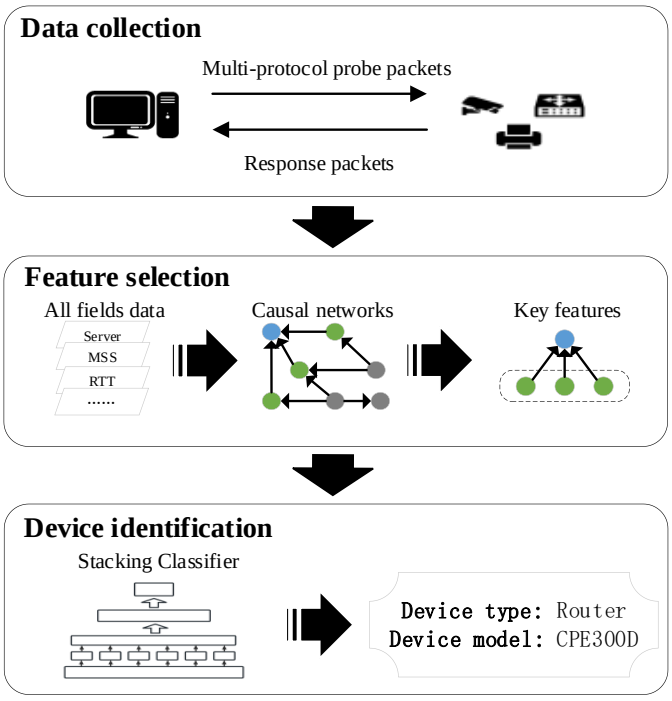


Figure 1. Framework for device identification.

4. Data Acquisition

In the network communication of IoT devices, the values of certain fields in the communication protocol can differ greatly between different devices. Among the seven-layer OSI network architecture, the messages of the transport and application layers are the most informative and closely related to the device attributes. In this paper, we choose to use the telegrams of the transport and application layers of IoT devices for device identification.

4.1. Selection of Protocols

For transport layer protocols, most of the IoT devices use two kinds of transport layer protocols, TCP and UDP. The header structure of UDP is relatively simple compared to the header structure of TCP, which carries more rich device characteristics and is conducive to device identification. In this paper, we choose TCP as the transport layer protocol.

The fields of the application layer protocol contain specific device information, so it can play an important role in device identification. In this paper, some application layer protocol contents can be used as features for device identification. In order to reduce the time consumption and network overhead as much as possible without reducing the identification accuracy and precision, the application layer protocol selected in this paper needs to meet two conditions: (1) In order to cover more IoT devices, the protocol should have a high usage rate among IoT devices. (2) The protocol should have rich fields and be distinguishable among different devices. In this paper, we counted the protocols used by IoT devices for network communication and found that the more common application layer protocols are HTTP, FTP, SSH, Telnet, UPnP, and so on. For the UPnP protocol, a certain number of IoT devices support this protocol, but UPnP needs to be opened manually and has some security risks, especially in some network environments with high security requirements, which leads to some restrictions on its practical application. Although the FTP and Telnet protocols also have a high usage rate in IoT devices, they are relatively simple, their communication messages contain fewer fields, and the fields are not well differentiated. If they are used for device identification, the feature space will be insufficient, so that the identification granularity of the device model level cannot be achieved. The HTTP and SSH protocols satisfy the two protocol selection conditions proposed in this paper, so HTTP and SSH are the two application layer protocols chosen for IoT device identification.

4.2. Rules for Data Acquisition

In order to collect as much device-related data as possible in a short period of time, for each device, this paper sends an application layer request packet to the device to be tested on the basis of establishing a TCP connection once, so as to obtain response packets from the device.

The TCP packets sent by the IoT device during the establishment and disconnection phases of the TCP connection have greater value. Figure 2 shows the process of establishing and disconnecting the TCP connection between the host and the IoT device, and this paper collects the SYNACK packets sent by the IoT device during the establishment phase and the FINACK and FIN packets sent by the device during the disconnection phase.

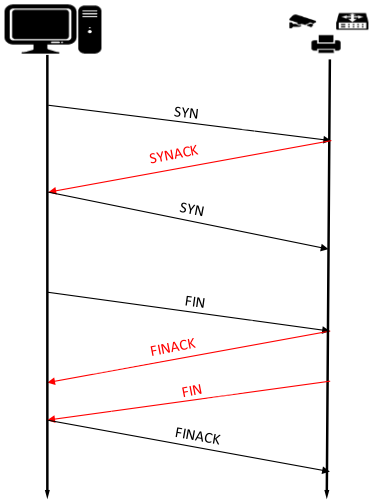


Figure 2. Acquisition of TCP packets.

After a TCP connection is successfully established with the IoT device, request packets for the application layer protocol are sent to the IoT device, and the application layer responses from the IoT device are collected. For the HTTP protocol, only one application layer response packet needs to be collected to collect all the field data, while the SSH protocol needs to collect the Request Reply packet, Key Exchange Init packet, and Key Exchange Reply packet to collect all the fields needed in this paper. The collection process is shown in Figure 3.

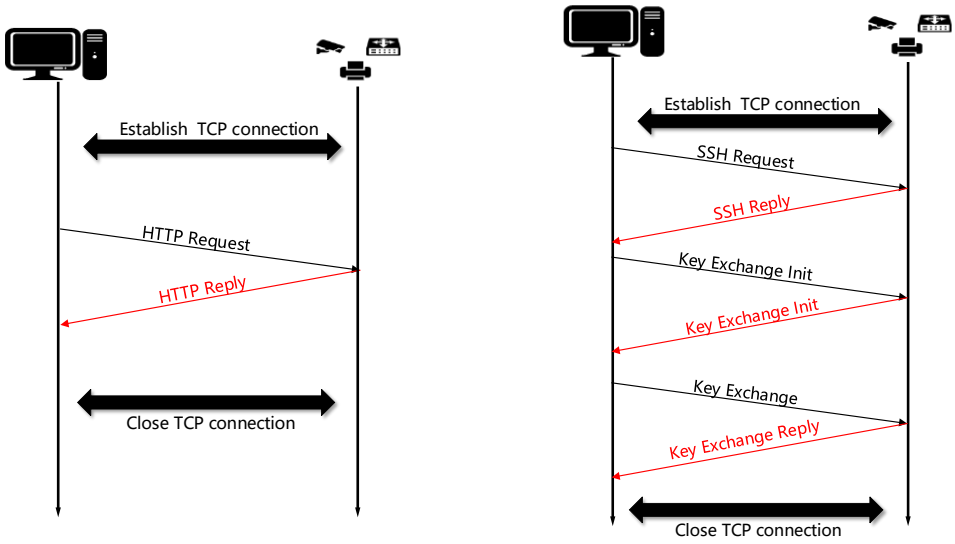


Figure 3. Acquisition of application layer packets. (**a**) HTTP protocol; (**b**) SSH protocol.

In the process of establishing TCP connections, performing application layer interactions, and disconnecting TCP connections, a total of three TCP packets as well as one application layer packet are collected in this paper for an HTTP/TCP IoT device, and three TCP packets as well as three application layer packets are collected in this paper for an SSH/TCP device.

4.3. Data Preprocessing

The fields of the transport layer protocols are all numeric, and this paper normalizes the numeric fields using the following normalization method:

$$x' = \frac{x - x_{avg}}{x_{\max} - x_{\min}} \tag{1}$$

When normalizing a field value, x' is the final normalization result for a device, x_{avg} is the average value of such fields, x_{min} is the minimum value in the set of such field values, and x_{max} is the maximum value in the set of such fields.

Most of the field features of application layer protocols are text-based, and in order to unify the data form and facilitate further work, this paper designs a method to transform textual data into numerical data: first, removing useless symbolic information in the text such as ".", "-" and other useless symbolic information, followed by embedding the text features into n-dimensional vectors using the Doc2Vec [22] tool, and finally, normalizing them. The fields of the transport and application layer protocols are combined to form the initial set of fields for the device.

5. Feature Extraction and Device Identification

5.1. Method for Feature Selection

After collecting the initial set of fields of IoT devices, this paper needs to select the device features from them for device identification. Traditional feature selection methods can be divided into two categories: manual feature selection methods based on domain-specific expertise and general feature selection methods based on data statistics. The former requires researchers to understand in detail the role of each alternative field for each protocol and also requires extensive experiments to verify the validity of the selected fields, which is very costly when applied to IoT devices with multiple different protocols. The latter approach selects features from a statistical perspective based on conditions such as variance and information entropy of the data and can be applied to different specialized fields. However, the statistical-based feature selection method only finds correlations between fields without considering causality, and it is easy to select redundant features that are only associated with identification targets without causality, thus affecting classification performance.

The causal inference method can automatically discover causal relationships among fields in the data, not just correlations, to construct a complete causal network. In this paper, we use the method of causal inference to find out the causal direction of each field in the initial field set with the device identification result in the normalized data to construct a complete causal network and then select the fields in the causal network that have a direct causal relationship with the identification result as the device key features.

The experimental data in this paper include the communication protocol fields of IoT devices across layers, which have the characteristics of high-dimensional data, and the PC algorithm [11] has good performance in high-dimensional data, so this paper chooses to use the PC algorithm for causal inference. Figure 4 shows the causal feature selection method designed in this paper.

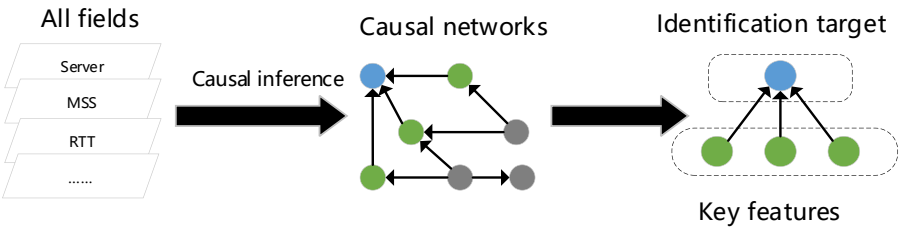


Figure 4. Method for feature selection.

5.2. Device Identification

In this paper, the key features selected by the causal feature selection method are classified to achieve device identification. In the experiments, this paper finds that different machine learning classifiers have large performance disparities in classification for different devices. For example, the accuracy of a DT classifier for S7-1500 PLC and DCS-2121 router is only 82.4% and 79.3%, respectively, while the accuracy for most other devices identification is around 90%, i.e., the model variance of individual classification models is large.

In order to reduce the model variance to improve the overall identification accuracy as well as the accuracy, this paper uses the Stacking [21] integrated classifier. The Stacking method is a hierarchical model integration framework, which is generally divided into two layers: the first layer is a number of primary models, called L1-Learners, trained with the training set; the second layer is a high-level model, which is trained with L1-Learners for the test set and the output values of the data as the output values for the high-level model, called the L2-Learner. To avoid overfitting, the L2-Learner needs to use simpler classifiers and different data for the two layers in one training. As shown in Figure 5, the L1-Learners of the Stacking classifier in this paper consist of six distinct classifiers, including a support vector machine (SVM) [23], decision tree classifier (DT) [24], k-nearest neighbor (k-NN) [25], linear discriminant classifier (LDA) [26], plain Bayesian classifier [22], and one-dimensional

convolutional neural network [27]. The L2-Learner of the Stacking classifier uses Logistic Regression [28].

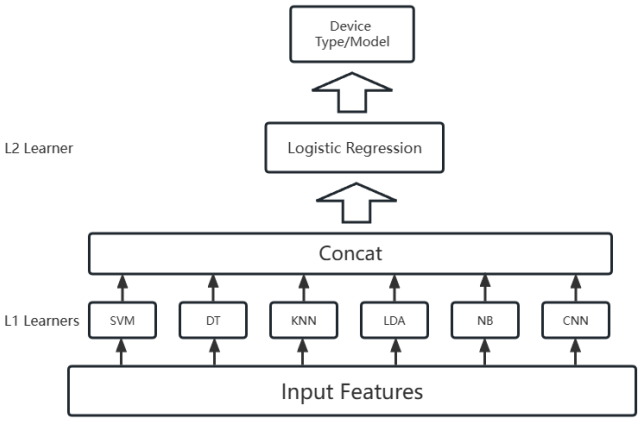


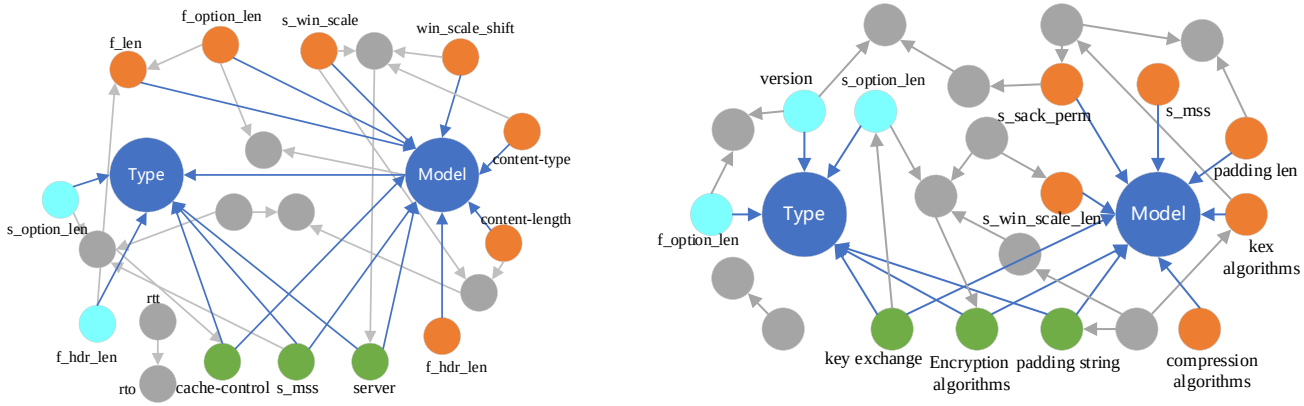
Figure 5. Stacking classification model.

6. Experiments and Evaluation

In this section, we will evaluate the multi-protocol device identification method based on the causal feature selection method proposed in this paper through experiments. By calling the Censys API, we obtained 15,000 IoT devices with their IP addresses, involving 3 device types (webcam, printer, and router) and 25 device models. In this paper, we will use these data for experimental evaluation and comparison with existing device identification methods to validate the identification capability of the proposed method in this paper.

6.1. Evaluation of Feature Selection

In this paper, data collection was performed for all experimental devices, and after data pre-processing using the PC algorithm [11], the causal network between each field and the device identification target (device type, device model) under the HTTP/TCP cluster and SSH/TCP protocol cluster is shown in Figure 6, where the blue dots indicate the device identification target (type and model), the orange dots indicate the fields with a direct causal relationship with the device model, the light blue dots indicate the fields with direct causal relationship with the device type, the green dots indicate the fields with causal relationship with both the device type and the device model, and the gray dots indicate the fields without a direct causal relationship with the device identification target.



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

Figure 6. Causal network of communication protocol fields for IoT devices. (**a**) HTTP protocol; (**b**) SSH protocol.

The causal network obtained from the experiments in this paper is largely consistent with the a priori knowledge of IoT devices as well as communication protocols in this paper. For example, MSS (Maximum Segment Size) is an option defined by the TCP protocol for the maximum length of data that each message segment can carry when the sender and receiver negotiate communication during TCP connection establishment. Option_len is the length of the option part of the TCP message, and these two fields will directly affect the length of the TCP message, so the causal network RTT (Round Trip Time) is the difference between the time when the data are sent and the time when the acknowledgement is received. Although the RTT algorithm may be different for different devices, RTT is more affected by network fluctuations, geographical location, and other objective conditions. Therefore, the RTT field has no effect on the device identification. RTO (Retransmission Time-Out) is the waiting time for TCP retransmission; if no ACK is received within the RTO time, the previous TCP message will be retransmitted; this field is calculated from the RTT. Reflected in the causal network, there is an edge where the RTT points to the RTO, neither of which has a causal relationship with the device identification target.

In this paper, the fields with a direct causal relationship with the device identification target are obtained from the causal network as the key features for device identification. Table 1 shows the key features of both device type and device model granularity selected in this paper for HTTP/TCP and SSH/TCP protocol clusters.

Table 1. Key features of device identification.

Identification Target	HTTP/TCP Features	SSH/TCP Features
Device Type	server cache-control f_hdr_len s_mss s_option_len	version padding string key exchange code encryption algorithms f_option_len s_option_len
Device Model	F_len f_option_len s_mss s_sack_len s_win_scale s_win_scale_shift server cache-control content-type content-length www-authenticate	f_option_len s_mss s_sack_perm s_win_scale_len padding length padding string key exchange code kex algorithms encryption algorithms compression algorithms

In order to evaluate the effectiveness of the features selected by the feature selection method in this paper, the features selected by the causal feature selection method are compared with the traditional statistical-based feature selection algorithms Percentile [29], FWE [30], RFE [31], variance threshold [32], and Chi2 [33] and manually selected features [19]. The features selected under HTTP/TCP protocols were used in comparison experiments using the same classifier and training methods. The features selected by each method are shown in Table 2.

Figure 7a shows the ROC curves of the seven feature selection methods for device type identification, and Figure 7b shows the ROC curves for device model identification. From the ROC curves of the seven feature selection methods, it can be judged that all seven methods achieve excellent classification performance in recognizing device types. Since other feature selection strategies can only select based on the correlation between fields, while the feature selection method proposed in this paper performs feature selection from the perspective of causality, which can avoid selecting redundant features compared with

the correlation-based feature selection, the method in this paper is significantly better in identifying finer-grained equipment models.

Table 2. Features selected by different methods.

Methods	Features
Percentile	server, www-authenticate, contenttype, s_mss, f_len, s_rtt, s_len, f_nop
FWE	server, www-authenticate, contentlength, cache-control, s_len, f_len, f_rtt, s_nop
RFE	www-authenticate, content-type, s_win_scale_len, s_rtt, s_len, f_nop, f_len, s_option_len
Variance threshold	www-authenticate, content-type, s_win_scale_len, s_rtt, s_len, f_nop, f_len, s_option_len
Chi2	server, content-length, www-authenticate, cache-control, s_mss, s_len, f_hdr_len, f_nop
Manual selection	server, contenttype, contentlength, cache-control, ave-segment, ave-win

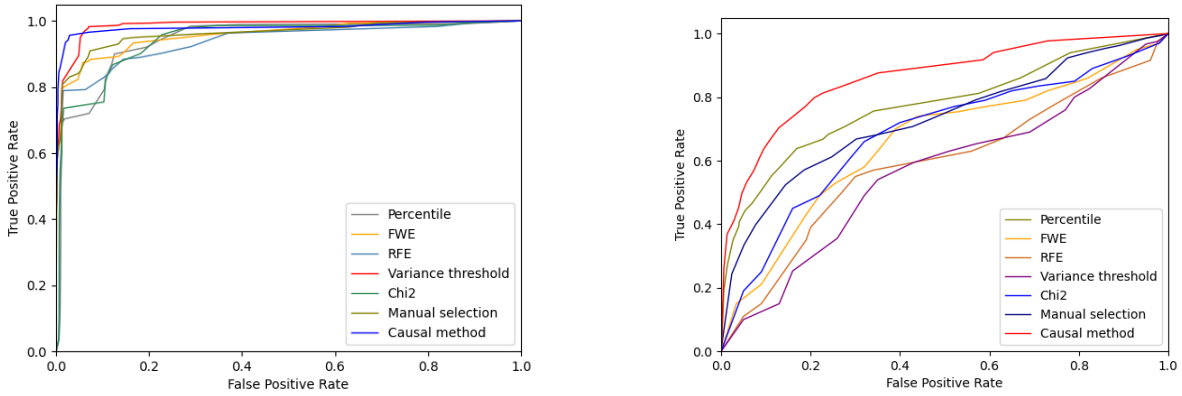


Figure 7. ROC comparison of feature selection methods. (**a**) Device Type Identification ROC; (**b**) Device Model Identification ROC.

To further quantitatively evaluate the effectiveness of the feature selection method in this paper, the selected features are evaluated using consistency and variance quantification methods.

The consistency of a feature is the stability of a feature among similar devices, and this paper uses information entropy to measure the consistency of a field:

$$H(Y) = -\sum_{i=1}^{m} p_i \log_2 p_i \tag{2}$$

$$C = \frac{1}{n} \sum_{i=1}^{n} H(Y_i) \tag{3}$$

Y represents the value of a field in the same kind of equipment, including $\{y_1, y_2, \ldots y_m\}$, the number of samples of such equipment is m, and the percentage of the value y_i is p_i. If the distribution of the elements of the set Y is more concentrated, the information entropy is smaller; if the distribution of the value of this field is more disordered, the information entropy is larger. After obtaining the information entropy of all the n devices in this field, the information entropy of the n devices in this field is averaged, the final result C represents the discrete situation of this field, and the consistency of the field is better when the value of C is smaller.

Differentiation is the degree to which a feature can be distinguished between different devices. It represents the ability of a feature to distinguish between different devices. In this paper, we use the optimized Euclidean distance to represent the differentiation of this field:

First, we calculate the proportion of samples with the same value in a field among different devices among all samples. $W = \frac{x_{same}}{x_{all}}$, x_{same} denotes the number of samples with the same value in a field as different devices, and x_{all} denotes the number of all samples.

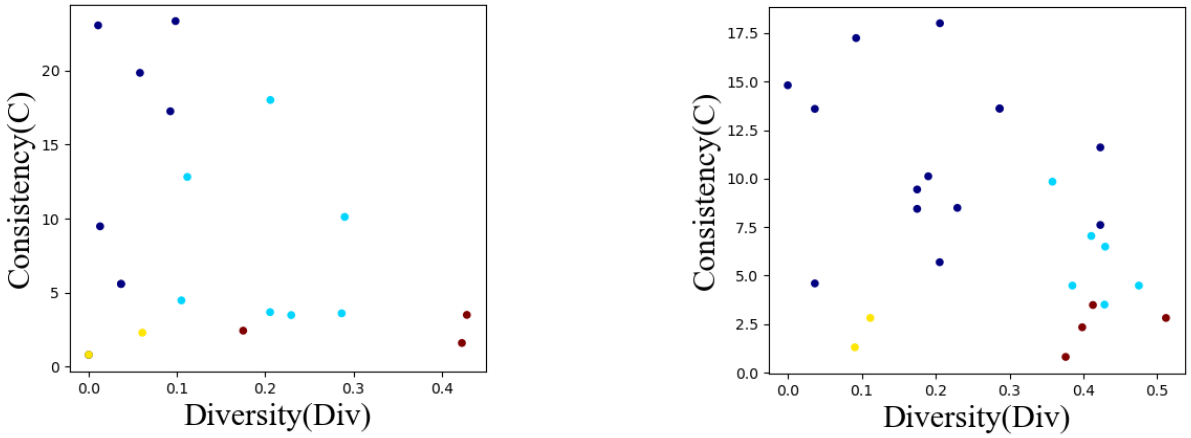
The normalized Euclidean distance D of all devices in this field is then calculated. x_i and y_i are the values taken by any two different devices in a given field.

$$D = \sum_{i=1}^{m} \sqrt{x_i^2 - y_i^2} \tag{4}$$

In order to reduce the impact of data randomness, if the more repeated points this field detects, the higher the case of different devices appearing with the same value, the actual distance calculated between different devices should be smaller, so this paper uses $1 - W$ (non-overlapping ratio) and the Euclidean distance D multiplied to obtain the quantitative index of field discrepancy Div; the larger the Div value, the better the field discrepancy.

$$Div = \frac{(1 - W) \sum_{j=1}^{L} D_j}{L} \tag{5}$$

In this paper, the consistency quantification C and the difference quantification Div of all fields in the initial field set are derived, as shown in Figure 8, with the horizontal axis denoting Div and the vertical axis denoting C. The dark blue dots are the unselected fields, the brown points are the features common to the identified device types and device models, the yellow dots are the features for the identified device types, and the light blue dots are the features for the identified device models. It can be seen that the selection of features in identifying device types will be more inclined to demonstrate good consistency to find similarities between similar devices; in identifying fine-grained targets such as device models, it will be more inclined to feature differences to distinguish different devices; and features common to identifying device types and device models have good performance in both consistency and difference dimensions at the same time.



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

Figure 8. Consistency and discrepancy of fields. (**a**) HTTP/TCP protocol; (**b**) SSH/TCP protocol.

6.2. Classification Performance Evaluation

To reduce the variance of the model, this paper uses a two-layer Stacking classifier to classify the selected features to achieve device identification. In order to demonstrate the effect of the Stacking classifier, this paper uses Stacking and 6 other classifiers to classify 15,000 devices at 2 granularities of device type and device model. In order to avoid overfitting, one training of the two layers should use data that do not contain each other. An intuitive approach is to divide the training data into two mutually exclusive parts, A and B, and use A to train L1-Learners, and then train the L2-Learner with the output of the L1-Learners on B. Although this approach avoids the same data being used by two layers of Learner in one training, the number of training samples in each layer is only half of the total sample size, and insufficient samples may easily lead to overfitting. The

approach in this paper is similar to 10-fold cross-validation. The data set is divided into 10 mutually exclusive subsets of similar size, and 10 models are trained for each L1-Learner. The outputs of the 10 models on their respective verification sets (1/10 of the data not involved in the training) are combined to form the input of the next layer with the same number of original samples, which ensures the number of training samples and avoids using the same data for both layers in one training.

Figure 9 shows the results of classifying device models using the Stacking method with six distinct classifiers in this paper. The x-axis indicates 25 different device models, and the y-axis indicates the accuracy of the classifier for that model. Figure 10 shows the accuracy, recall, and inter-class accuracy variance of different classifiers. From the perspective of accuracy and recall, SVM, DT, CNN, and Stacking classifiers all achieved significant device identification performance. However, from the perspective of variance and the identification accuracy of each device model in Figure 9, the accuracy of SVM, DT, and CNN fluctuates widely among different device models with large variance, and the identification stability of these models has a large gap compared to the Stacking model. The inter-class performance of Stacking is more stable, and it can be concluded that the Stacking method significantly reduces the variance of the model and achieves better classification performance.

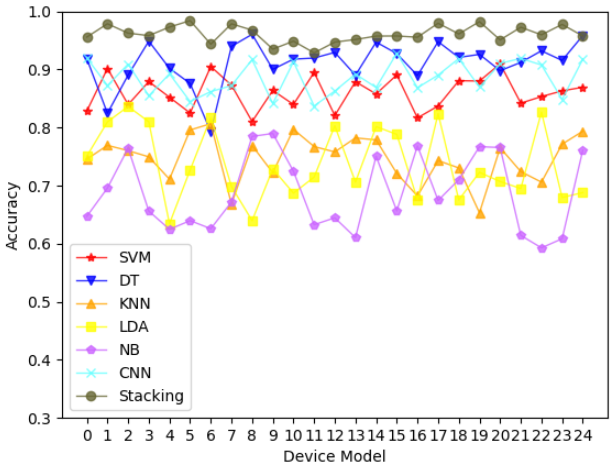


Figure 9. Accuracy of each classifier for each device model.

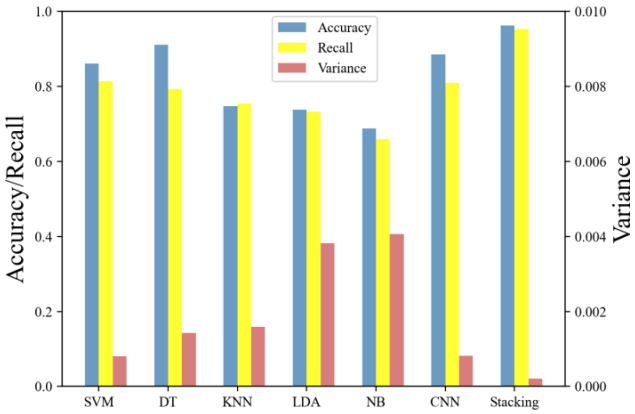


Figure 10. Accuracy, recall, and variance of each classifier.

6.3. Comparison with Existing Methods

In this section, we will compare our method with existing proactive identification methods. In addition to accuracy and recall, the compared items also include time consumption. The time consumption of the device identification process is another important indicator of the performance of an identification method. In order to verify the timeliness

and accuracy of the multi-protocol device identification method based on the causal feature selection method proposed in this paper, the identification accuracy, recall, and average time consumption of 15,000 devices are counted and compared with the manual feature selection [19], banner-based method [15], and Nmap [4]. Figure 11 shows the comparison of accuracy and time consumption of each method. The *X*-axis shows each identification method, the blue and yellow bars indicate the identification accuracy and recall rate, respectively, corresponding to the left *Y*-axis, and the red bars indicate the time consumption of that method, corresponding to the right *Y*-axis. Table 3 shows the specific numbers for each method of comparison. From the time efficiency dimension, Nmap needs to send the largest number of messages and consumes the longest time, and the banner-based method needs to send more detection messages than both the manual feature extraction method and the method in this paper, so it also takes a longer time. From the perspective of accuracy and recall, the recall rate is low because the banner-based method is more dependent on the device vendor's settings and cannot identify when the device banner does not have key information. Due to the use of the Stacking model, the method in this paper requires more models to be trained, but all L1-Learners of the Stacking model can be trained in parallel, so the time consumption in this paper is not too different from the method in the literature [19], and since this paper selects features from the perspective of causality rather than just correlation, the Stacking method is used. The variance of the classification model in this paper is smaller, and the identification performance is higher.

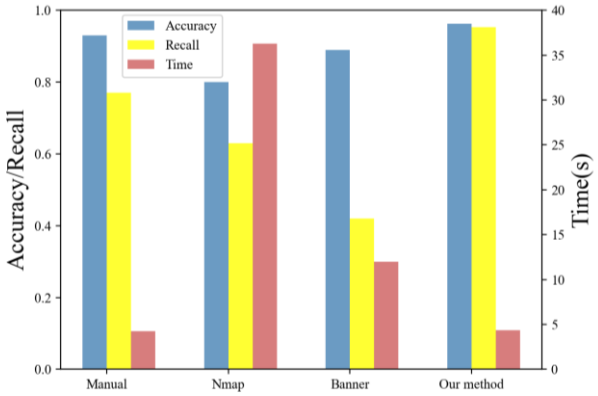


Figure 11. Accuracy, recall, and time comparison with existing methods.

Table 3. Accuracy, recall, and time comparison with existing methods.

Methods	Accuracy	Recall	Time (s)
Manual	93.1%	77.2%	4.24
Nmap	80.3%	62.9%	36.31
Banner	88.9%	42.1%	11.95
Our method	**96.3%**	**95.4%**	**4.36**

7. Conclusions

In this paper, we designed an IoT device identification method based on causal inference, which automatically extracted key features in different protocol fields from the causality perspective using the causal inference approach and achieved high-precision and fine-grained device identification. The experimental results showed that the method proposed in this paper achieved 96.3% accuracy under HTTP/TCP protocol cluster and 97.7% accuracy under SSH/TCP protocol cluster. However, the shortcoming of the proposed method is that the impact of the strength of causality on identification results is not considered in the selection of features and identification of devices. To further enhance the performance of IoT device identification, we should research a device identification

method that can take into account the strength of causal relationships in causal networks, i.e., the weights of the edges in causal networks.

Author Contributions: Conceptualization, X.W. and Y.W.; methodology, Y.W. and D.Y.; software, Y.Y. and Y.M.; validation, X.W. and Y.W.; formal analysis, D.Y.; investigation, Y.W.; writing—original draft preparation, Y.W.; writing—review and editing, X.W.; visualization, D.Y.; supervision, Y.C.; project administration, Y.C.; funding acquisition, Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Program of Shanxi Province, grant numbers 20210302123131 and 210302124395. The APC was funded by Taiyuan University of Technology.

Data Availability Statement: The data used in this article were obtained from Censys and are available at https://search.censys.io/ under license from Censys.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Pacheco, J.; Hariri, S. IoT security framework for smart cyber infrastructures. In Proceedings of the 2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W), Augsburg, Germany, 12–16 September 2016; IEEE: New York, NY, USA, 2016.
2. Li, Q.; Feng, X.; Zhao, L.; Sun, L. A framework for searching Internet-wide devices. *IEEE Netw.* **2017**, *31*, 101–107. [CrossRef]
3. Shodan Search Engine. Available online: https://shodan.io/ (accessed on 25 May 2023).
4. Duarte, F.S.L.G.; Sikansi, F.; Fatore, F.M.; Fadel, S.G.; Paulovich, F.V. Nmap: A novel neighborhood preservation space-filling algorithm. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 2063–2071. [CrossRef] [PubMed]
5. Spirtes, P.; Glymour, C.N.; Scheines, R. *Causation, Prediction and Search*; MIT Press: Cambridge, MA, USA, 2000.
6. Zhang, K.; Peters, J.; Janzing, D.; Schölkopf, B. Kernel-based conditional independence test and application in causal discovery. *arXiv* **2012**, arXiv:1202.3775.
7. Chickering, D.M. Optimal structure identification with greedy search. *J. Mach. Learn. Res.* **2002**, *3*, 507–554.
8. Tsamardinos, I.; Brown, L.E.; Aliferis, C.F. The max-min hill-climbing Bayesian network structure learning algorithm. *Mach. Learn.* **2006**, *65*, 31–78. [CrossRef]
9. Akaike, H. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*; Springer: New York, NY, USA, 1998; pp. 199–213.
10. Lam, W.; Bacchus, F. Learning Bayesian belief networks: An approach based on the MDL principle. *Comput. Intell.* **1994**, *10*, 269–293. [CrossRef]
11. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.R.; Tarkoma, S. 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), Atlanta, GA, USA, 5–8 June 2017; IEEE: New York, NY, USA, 2017.
12. Thangavelu, V.; Divakaran, D.M.; Sairam, R.; Bhunia, S.S.; Gurusamy, M. DEFT: A distributed IoT fingerprinting technique. *IEEE Internet Things J.* **2018**, *6*, 940–952. [CrossRef]
13. Marchal, S.; Miettinen, M.; Nguyen, T.D.; Sadeghi, A.-R.; Asokan, N. Audi: Toward autonomous iot device-type identification using periodic communication. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1402–1412. [CrossRef]
14. Aneja, S.; Aneja, N.; Islam, M.S. IoT device fingerprint using deep learning. In Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS), Bali, Indonesia, 1–3 November 2018; IEEE: New York, NY, USA, 2018.
15. Feng, X. Acquisitional rule-based engine for discovering internet-of-things devices. In Proceedings of the 27th USENIX Security Symposium (USENIX Security 18), Baltimore, MD, USA, 15–17 August 2018.
16. Wang, X.; Wang, Y.; Feng, X.; Zhu, H.; Sun, L.; Zou, Y. Iottracker: An enhanced engine for discovering internet-of-thing devices. In Proceedings of the 2019 IEEE 20th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Washington, DC, USA, 10–12 June 2019; IEEE: New York, NY, USA, 2019.
17. Censys. Available online: https://censys.io (accessed on 25 May 2023).
18. Zoomeye. Available online: https://zoomeye.org (accessed on 25 May 2023).
19. Yu, D.; Xin, H.; Chen, Y.; Ma, Y.; Chen, J. Cross-Layer Protocol Fingerprint for Large-Scale Fine-Grain Devices Identification. *IEEE* **2020**, *8*, 176294–176303. [CrossRef]
20. Quoc, L.; Tomas Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014; PMLR: Cambridge, MA, USA, 2014.
21. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259. [CrossRef]
22. Barnard, G.A. The foundations of statistics 1964–1974. *Bull. IMA* **1974**, *10*, 344–347.

23. Boser, B.E.; Guyon, I.M.; Vapnik, V.N. A training algorithm for optimal margin classifiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, PA, USA, 27–29 July 1992.
24. Loh, W.-Y. Classification and regression trees. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 14–23. [CrossRef]
25. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
26. Mitchell, T.M. *Machine Learning*; McGraw-Hill: New York, NY, USA, 2007; Volume 1.
27. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Part I 13; Springer International Publishing: Cham, Switzerland, 2014.
28. Copas, J.B. Regression, prediction and shrinkage. *J. R. Stat. Soc. Ser. B Methodol.* **1983**, *45*, 311–335. [CrossRef]
29. Percentile. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectPercentile.html (accessed on 25 May 2023).
30. FWE. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFwe.html (accessed on 25 May 2023).
31. RFE. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html (accessed on 25 May 2023).
32. Variance Threshold. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html (accessed on 25 May 2023).
33. Chi2. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.chi2.html (accessed on 25 May 2023).