Web Service Access Control Based on Browser Fingerprint Detection

Liu Hui, He Xudong*, Gao Fan, Wang KaiLun and Yuan Enze

Beijing Jiaotong University, China
E-mail: hexudong@bjtu.edu.cn
**Corresponding Author*

Received 25 May 2021; Accepted 08 June 2021;
Publication 28 April 2021

Abstract

Web services have covered all areas of social life, and various browsers have become necessary software on computers and mobile phones, and they are also the entrances to Web services. All kinds of threats to web data security continue to appear, so web services and browsers have become the focus of security. In response to the requirements of Web service for access entity identification and data access control, this paper proposes a multi-dimensional browser fingerprint detection method based on adversarial learning, and designs a Web service access control framework combined with browser fingerprint detection. Through the joint use of multi-dimensional browser features, adversarial learning is used to improve the accuracy and robustness of browser fingerprint detection; a cross-server and browser-side Web service access control framework is established by creating tags for Web data resources and access entities. Based on the mapping relationship between browser fingerprint detection entities and data resources, fine-grained hierarchical data access control is realized. Through experiments and analysis, the browser fingerprint detection method proposed in this paper is superior to existing machine learning detection methods in terms of accuracy and robustness. Based on the adversarial learning method, good detection

Journal of Web Engineering, Vol. 20_5, 1587–1622.
doi: 10.13052/jwe1540-9589.20512

results can be obtained in the case of a small number of user samples. At the same time, the open source data set is further used to verify the advantages of the method in this paper. The Web service access control framework can satisfy the requirements of Web data security control, is an effective supplement to user identification technology, and is implementable.

Keywords: Access control, adversarial learning, browser fingerprint, web service.

1 Introduction

With the rapid development of information technology, according to the 47th "Statistical Report on Internet Development in China"[1], as of December 2020, the number of Internet users in my country has reached 989 million, and the Internet penetration rate is 70.4%. Various application scenarios are described. Web services are evolving rapidly, covering almost every application scenario. Corresponding to the development of web services, browsers are used as the entrances to web services to interact with various important service systems to realize various network services, such as web browsing, information management, and online shopping. Browsers have become a hot spot for major Internet companies. For web-based information systems, the web service backend stores massive amounts of polymorphic data, involving information in social, commercial, and personal fields. The web data access is realized through the browser, and the browser and the web system build a data storage access Closed-loop system. Currently, governments and enterprises usually have multiple Web systems for different businesses. In order to facilitate user management and service access, single sign on access to multiple Web systems has become the mainstream. By adopting this implementation scheme, on the one hand, the system development and deployment becomes convenient; on the other hand, the unified entrance makes the identity management of the system simple and efficient. When accessing services through a browser, the user name/password has become the main authentication method. In order to enhance the authentication security, additional devices (such as USB Key, fingerprint collection devices, etc.) can be used to form a multi-factor authentication scheme.

Under such a service model, the security problems of the Web system continue to appear, especially the data in the Web system faces security risks such as unauthorized access and leakage. The browser is the entrance of the service, and attacks on the Web system through the browser have become

one of the main security threats. When the user accesses the service through the browser, there will be password leakage, access hijacking, etc., which will lead to the occurrence of user identity fraud; at the same time, existing Web services have coarse granularity of access control, which does not meet the requirements of fine-grained access control of data in the network environment. Coarse-grained role control cannot meet the requirements of precise control of hierarchical data access requests, so various situations of exceeding the scope of authority appear.

It is common in Web services to use browser-related information (such as cookies, etc.) for entity identification to implement access control. However, this solution using single browser information has the problems of insufficient concealment and insufficient accuracy. Cookies also have the risk of privacy leakage.

At the same time, some Web service security methods available on the server side, such as OAuth [2], SAML protocol [3] and HTTP authentication methods, etc., due to poor compatibility, custom selection settings may lead to insecure implementation and not applicable to apps , Easy to tamper and other issues, need to further propose new supplementary programs.

Therefore, this article uses browser fingerprints to improve entity authentication and identification. In the face of browser configuration changes, it enhances the robustness of fingerprint identification algorithms under conditions of diversity and variability, and solves the problem of fine-grained access control. The main work of this paper includes: proposing a multi-dimensional browser fingerprint detection method based on adversarial learning to solve the problem of access entity identification and weak robustness of existing fingerprint recognition algorithms; on this basis, combined with browser fingerprint detection, design a cross-server and browser-side Web service access control framework to establish a mapping relationship between access entities and data resource subjects. Based on data tags and custom access control strategies, achieve fine-grained hierarchical data access control.

The content structure of the paper is arranged as follows: Section 1 is the introduction; Section 2 introduces the research status and existing problems of browser fingerprinting and access control; Section 3 designs the Web service access control framework and browser fingerprint detection method based on adversarial learning; Section 4 uses the self-collected data set to perform browser fingerprint access control detection; section 5 uses the open source data set to verify the validity of this method; finally, section 6 summarizes the work of this paper.

2 Research Background

The existing browser authentication methods are more about user identification in the application layer. The method proposed in this paper can add system layer and hardware layer for authentication on the basis of the application layer, which can effectively supplement the existing security authentication methods and further improve the security of Web services.

In order to realize the access control of Web service resources, this paper mainly adopts the browser fingerprint and access control technology, and describes the relevant research status.

2.1 Browser Fingerprint Related Work

Browser fingerprints are used to distinguish browsers by collecting browser characteristics. It is mainly used to identify visiting entities in Web services to provide recommendation services based on visit history and visit enhanced authentication.

Browser fingerprints were proposed in a paper published by the Electronic Frontier Foundation in 2010. Eckersley [4] collected 47,0161 browser fingerprint samples and found that browser fingerprints can show the diversity of devices, using HTTP headers or Java The fingerprint of the browser constructed by features such as data is unique. Generally, browser fingerprints don't need to store information such as cookies, but use browser software and hardware parameters as features. The existing browser fingerprint is a string representation that can uniquely identify the current browser. Pierre Laperdrix [5] divides the use of browser fingerprints into two categories: destructive and constructive. Here we mainly focus on the positive role of constructive, such as FaizKhademi [6] puts forward a method of FPGuard to fingerprint detection and prevention of browser, at run time by collecting related to fingerprint activities of fine-grained data, to analyze it, fingerprints can be changed each time a user access to the web browser, thus protecting privacy data were not easily leak. Torres et al. [7] use FP-Block's solution to separate Web identities, and browsers will present different fingerprints when connecting to different sites to prevent tracking. In contrast, Fiore et al.'s [8] research creates similar fingerprints based on previous data, and counters malicious detection with forged consistent fingerprints.

On the other hand, browser fingerprints can be used to uniquely identify Web services and online behaviors, and can be accurately detected when performing service access behavior recognition to distinguish browser access permissions.In terms of browser fingerprint features, Cao [9] proposed the

cross-browser Fingerprinting technology. On the basis of the traditional browser fingerprint feature selection, the operating system and hardware attributes are added, such as video card and CPU, etc., and the WebGL feature is also proposed to realize the computer recognition under the scenario where users switch browsers. It realizes the recognition of the computer when the user switches the browser. Vastel [10] created the FP-STALKER link browser fingerprint evolution method, which is based on rules and machine learning methods to compare whether fingerprints originate from the same browser. There are improvements in fingerprint recognition for changes, but improper feature selection can also lead to deviations in the experiment. Picasso [11] proposed to use specific graphics primitives in the Canvas API to detect the browser and operating system of the device. Rochet [12] used deep learning to create a personalized model for each device. After training 2,000 canvas images, they extracted features to build a binary classification model for users, and set a threshold. The browser score must be higher than a specific threshold. Able to accept or reject the connection. In terms of browser fingerprint applications, SecurAuth [13] as a provider of adaptive access control solutions, has successfully constructed a heuristic-based identity verification system through device fingerprint recognition, which also uses browser fingerprints as part of its multi-factor authentication process.

It can be seen from the above summary that browser fingerprints have been studied and applied to multi-factor authentication of access control solutions, but at the same time, browser fingerprint changes caused by browser software updates and configuration changes pose challenges for browser identification. Therefore, the robustness of the browser fingerprint detection method has become a key issue. Existing solutions also try to predict browser fingerprints through machine learning algorithms, but they cannot effectively solve the impact of fingerprint changes.

2.2 Access Control Related Work

Access control originated in the 1970s, mainly to solve the authorization problem caused by data sharing. Discretionary Access Control (DAC) means that the owner of the subject has the power to grant others access to the subject. Mandatory Access Control (MAC) and DAC is used in conjunction with the security control of the US military, but it has the phenomenon of malicious disclosure of information by users, and the integrity of the information cannot be guaranteed.In 1995, Ferraiolo [14] introduced “role” into the management model, linking “user-role-permission”, and proposed a

role-based access control model (Role-based Access Control, RBAC), which enhanced stability and Reduce management overhead.

In order to further solve the authorization problem, Sandhu [15] introduced authorization management permissions and authorization management roles on the basis of the existing RBAC96 model, and proposed a role-based access control management model, showing better flexibility and adaptability. Liu Wu [16] associate roles with trust, and propose a trust-based access control model TRBAC based on Blaze trust management, which refines the relationship between authority and trust management and improves the security of user authorization. Silva [17] proposed a method of access control, SarBAC, which dynamically adjusts business processes, and mitigates internal threats by dynamically assigning roles to users to perform tasks. This method uses business process execution tracking and Markov [18] model to establish confidence intervals for key measurable attributes of user behavior, thereby identifying and adaptively reducing users who are malicious or abusing access rights.In the same year, Yavari [19] proposed a role-based disclosure control technology suitable for IoT applications, and reduced the size of data before confusion by combining sensitive data with context, so as to reduce the occurrence of unauthorized access and sensitive information leakage in the IoT.

Aftab [20] proposed a hybrid access control model based on separation of authority and responsibility, which dynamically improves the performance of the RBAC model from automatic authority and role assignment, and adds homomorphic attributes on the basis of RBAC to support flexible and fine-grained access control. Maesa et al. [21] proposed an access control method created and implemented based on blockchain technology to allow distributed transfer of resource access rights between users. And this scheme allows distributed auditing to prevent one party from fraudulently denying the rights granted by the executable policy. It can effectively solve the problem of fine-grained access control in a dynamic large-scale environment.Yang et al. [22] proposed an adaptive access control system that can only delete duplicate data under the medical background, and can use attribute-based encryption and cross-domain access control strategies to realize information sharing.

Judging from the above existing work, the pure role based access control model can no longer meet the needs of users and the needs of complex application scenarios. A lot of work is based on RBAC combined with requirements to improve. In the problem of fine-grained data resource access control for Web access entities studied in this paper, RBAC cannot solve the fine-grained hierarchical data access control. Therefore, this paper proposes

the two-level association based on "user-identity-authority" and "user-label-classification". The two-level association between the user tag and data tag realizes the mapping relationship between entities and data resources.

3 Implementation of Web Service Access Control Based on Browser Fingerprint Detection

The recognition of browser fingerprints is often researched for one or more characteristics. For example, Englehardt and others [23] use the canvas element to detect the font set of user equipment; Rola and others [24] use timing side-channel attacks to detect browser extensions, etc., so as to perform user Recognition. Or by converting fingerprint features into hash strings for static identification, the method has strong limitations and poor practicability, and does not consider the complete attribute list of fingerprint identification, so the following work is introduced.

Section 3.1 presents a web service access control framework based on browser fingerprint detection (abbreviated as: web service access control framework), explains its composition and workflow in detail, proposes and analyzes an access control model based on identity and tags. In Section 3.2, according to the insufficient number of data sets, a browser fingerprint detection method based on adversarial learning is designed.

3.1 Web Service Access Control Framework

Web services include server side and browser side, and the server backend stores accessible data. As shown in Figure 1, the Web service access control framework includes: a detection information module and a behavior recognition module on the browser side; a security management module on the server side and an access control module based on identity and tags. The detection information module on the browser side has fingerprint extraction, token storage, and identification service interface functions for interaction with the server, the behavior recognition module has dynamic fingerprint analysis, behavior analysis, and behavior identification interface functions for interaction with the server. The security management module of the server has the functions of abnormal behavior detection, behavior identification interface and identification service interface; the access control module has the functions of browser fingerprint detection, identity and tag-based access control and data tagging.

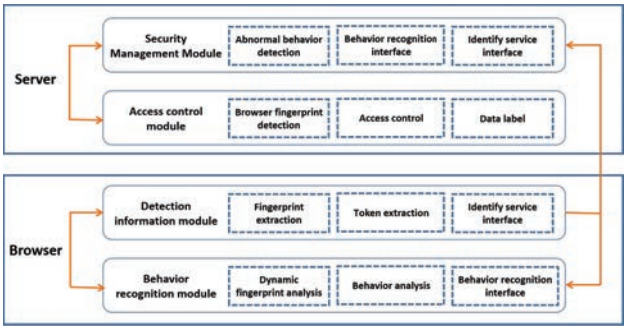

Figure 1 Web service access control process.

Under this framework, both the browser side and the server side have fingerprint analysis and access behavior analysis functions. Usually, the browser side adopts a lightweight analysis method, and the server side will perform training and learning mode to continuously update the model to improve the recognition and detection capabilities. The main workflow is described as follows:

(1) When the browser makes a request to the server to initiate a web service, the detection information module first initiates an authentication request to the identification service interface of the server. After the identification service interface uses the fingerprint detection function of the server access control module to identify the browser (static) fingerprint, Token is distributed to the browser, and the browser stores the Token. The Token will be used as a credential when the browser accesses the server-side Web service interface.

(2) The behavior recognition module on the browser side uses the collected dynamic fingerprints and related access behavior analysis to support abnormal behavior detection on the server side when accessing the server Web service interface after having the Token, so as to find abnormal access behaviors.

(3) The access control module of the server combines the fingerprint detection results of the browser and the data label function to set the data classification label for the data, and the access control model based on the identity and label realizes the control of the browser access request.

The Web access control framework proposed in this paper is implementable. Through pure front-end native Js to achieve browser fingerprint data acquisition, the collection overhead is small during the experiment,

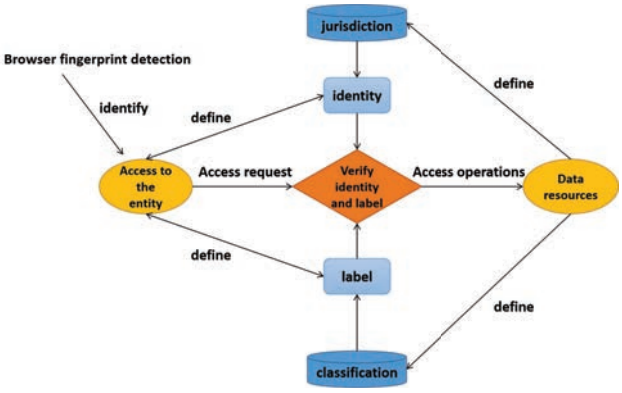

Figure 2 Access control model based on identity and label.

and the detection information module and behavior recognition module corresponding to the browser side can be implemented by browser plugins.

3.1.1 Access control model based on identity and label

Because the resource access requests sent from the browser are diverse, including different access request types, for example, requests have different permission levels or belong to different companies or organizations, and data resources are also classified by different levels. Therefore, a model that supports hierarchical data access control is needed to meet application access control requirements. For this reason, this section designs an access control model based on identity and tags, so that only authorized users can access the specified hierarchical data.

As shown in Figure 2, when a browser sends a Web service as a request, the subject or object of access control is a server resource access request initiated by a Web browser user or a Web browser user, while the object of access control is a variety of accessible data resources stored on the Web server.

(1) Model formal definition
First, give the definition of basic concepts, as follows:

(1) Visiting entity (user): Refers to the visitor in the Web system, this paper refers to the browser, the browser collection is denoted as $U = u_i|i \in N*$.
(2) Data resource: Refers to the data resource that can be accessed in the Web system, the data resource collection is denoted as $D = d_i|i \in N*$.

(3) Session: Refers to the executive body of the access entity in the Web system. A user can correspond to multiple sessions, and the session set is denoted as $S = su_i, sd_i | i \in N*$,where sui is the access entity session and sdi is the data resource session.
(4) Permission: Refers to operations on digital resources, such as reading, writing, modifying, etc. The set of permissions is recorded as $P = p_i | i \in N*$.
(5) Classification: Refers to the classification of the importance of data resources, such as important, medium, etc.And the classification set is recorded as $C = c_i | i \in N*$.
(6) Identity: Refers to the identity corresponding to the access entity, such as core personnel, ordinary personnel, etc. The set of identities is recorded as $ID = id_i | i \in N*$.
(7) Label: Refers to the classification identification corresponding to the data resource or access entity, such as important, medium, etc., and corresponding to the classification set. The label set is recorded as $L = l_i | i \in N*$.

Definition 3.1. Accessed entity identification refers to the use of browser fingerprint detection for entity identification, and the identification set is expressed as $Recg = recg_i | i \in N*$.

Definition 3.2. Access request refers to the access request made by the accessing entity to the data resources in the Web system. The set of access requests is expressed as $R = ri = |iN * \langle uU, dDrecgRecgidIDlL$.

Definition 3.3. The access operation refers to the access operation of the access entity to the data resource. The set of access operations is expressed as

$$O = o_i = \langle p, c, d \rangle | i \in N*, p \in P, d \in D, c \in C.$$

Secondly, the access control model based on identity and label includes the following components:

(1) Access entities, data resources, sessions, identities, permissions, tags, and classifications (see above for definitions).
(2) Access request.
(3) Access operation.
(4) The access control model based on identity and label verifies the identity and label of the visiting entity, as well as the data label to determine whether access to data resources is allowed.

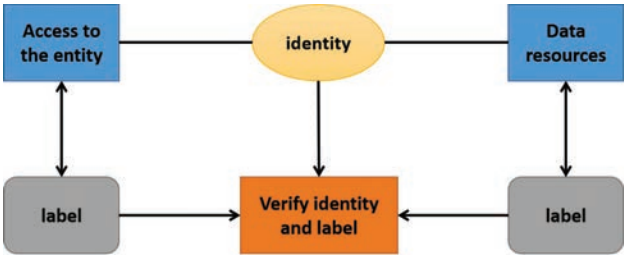

Figure 3 Model workflow.

Different access entity requests and different data resources have different hierarchical tags, through which different access control requirements can be directly reflected in the data resource tags. In the access control model based on identity and tags, any access entity (user request) and access object (data resource) have a complete set of tags. The label is set by the access entity based on identity, grants the session after the access request is authenticated, and supports access entity label modifications based on identity changes. Tags are granted when data resources are created, and tag modification is also supported.

(2) Model workflow

The access control model based on identity and label includes: label initialization grant module, identity and label verification module, label modification module and identity modification module. The workflow is as Figure 3.

Labels respectively grant access to entities and data resources. The label of the visiting entity is set by the identity of the visiting entity corresponding to the default label. As shown in the figure above, the label of the visiting entity can modify the default setting; this label is used to determine whether the visiting entity has corresponding access operation authority to the visiting object. The data label is set and allowed to be modified during the creation of data resources, which corresponds to the classification requirements of data resources.

Initialize the label module to complete the default label setting and label modification of the access entity. The label modification module completes label modification when the access entity or data resource is sent and changed; only by changing the label type of the access entity or data resource, the access control requirements can be met, and direct operations on the access subject and object can be avoided. The identity change module completes the identity modification of the visiting entity, and at the same time

cooperates with the label change module to complete the label modification. The identity and label verification module completes the identity and label verification of access entities and data resources to determine whether the session has access operation authority. The authentication strategy of the identity and label authentication module is the core of the basic access control model; when the Web system needs to strictly abide by the principle of least privilege or the information system strictly controls the information in the system can only flow unidirectionally from low-level objects to high-level objects, the decision strategy can refer to BLP model, that is, only when the tag level of the access subject (user request) is higher than or equal to the tag level of the data resource, the access requests of the access entity for the data resource is allowed and the access operation can be performed. Otherwise, the user's requests are rejected.

3.1.2 Web service access control model security analysis

The Web service access control model based on identity and label is designed to meet the requirements of fine-grained data and hierarchical access control studied in this paper. Compared with the existing access control schemes, it has the following advantages from the perspective of security.

(1) Security protection of user data: Browser fingerprint detection is introduced to identify the browser that initiates the access request in the access control. Combined with browser fingerprint, the security of Web service access is enhanced in the absence of user perception, and the integrity of data and the authenticity of data sources are guaranteed at the same time.

(2) The fine-grained access control: based on the identity and access control model is combined with labels, based on the thought of roles and access control based on "user identity – permission" and "user – label – classification" of two layers of associations, by the user tag and data match mapping relationship between entities and the data resources, in the access request authenticated access entity identity and labels, hierarchical data access control. Only when the user's fingerprint data meets the match, can the target resource be accessed to ensure that the falsified and tampered user information can be identified.

(3) Portability: This framework is divided into the browser side and the server side, with expansibility and flexibility. Analysis and detection functions can be deployed on the browser side, and can also be implemented on the server side, which can meet the application environment of different security requirements.

(4) Adaptability: The access control model proposed in this paper also has simple modification of labels, which can adapt to changes in the grading of access entities and data resources and has low modification overhead.
(5) Anti-leak and Malicious Sharing: Our access control mode is an effective supplement to the existing authentication mode. When the user's sensitive information is leaked or shared maliciously, our model can guarantee that the attacker can only obtain invalid information and has no access control authority.

Web Service access control model proposed in this paper can effectively deal with the network system of information manipulation and unauthorized access, as well as the XSS (Cross site scripting) and DDOS (Distributed Denial of Service) and intrusion detection threat model has a good effect. It can prevent a lot of redundant data occupy resources, so that they can identify legitimate users and respond in a timely manner.

3.2 Implementation of Fingerprint Detection Algorithm based on Adversarial Network

In the Web service access framework and access control model in the previous section, browser fingerprint detection realizes the identification of access entities, correlates the identity and label in the access request according to the identification results, and then performs the verification function to decide whether the requested data resources can be accessed.

However, the data collection of browser fingerprints involves privacy issues, so compared with traditional experiments, it has the characteristics of less data and weak algorithm robustness. In order to solve this phenomenon, this section designs a multi-dimensional browser fingerprint detection method based on adversarial learning to enrich the experimental algorithm and expand the data. Figure 4 and Algorithm 1 is a simple process of the fingerprint detection enhancement method based on adversarial learning.

Algorithm 1 The fingerprint detection enhancement method based on adversarial learning

Input: Browser fingerprint data;

Output: Model parameters;

1: Fingerprint feature generation and preprocessing;
2: Classify using classification model;
3: Use generative adversarial networks to train data and generate new data $\min_G \max_D V(D,G) = E_{x\sim p_{data(x)}}[logD(x)] + E_{z\sim p_{z(z)}}[log(1-D(G(z)))]$
4: Use the classification model to train the newly generated data;

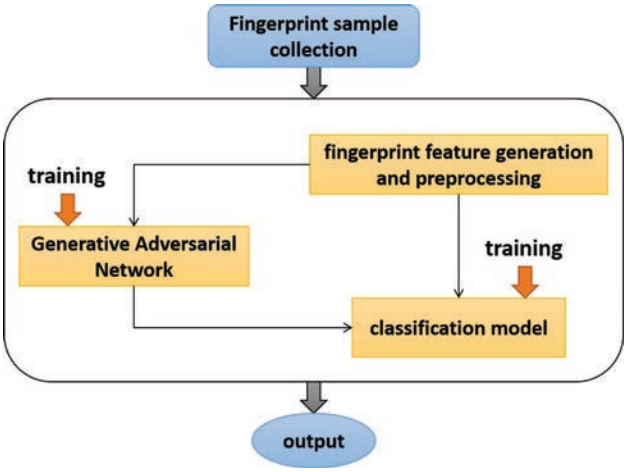

Figure 4 Multi-dimensional browser fingerprint detection method based on adversarial learning.

The method used to enhance browser fingerprint detection mainly uses the combination of static and dynamic browser fingerprints in feature selection to generate browser 34-dimensional fingerprint features; the generated browser multi-dimensional fingerprint features are preprocessed and transformed to generate 28* 28 gray-scale images; in order to solve the lack of sample data and improve the robustness of the algorithm, the adversarial learning (GAN) is introduced for data expansion; finally, a variety of deep learning models (CNN model, hollow convolution model, residual network model) are used Perform classification to achieve browser fingerprint detection.

The details of related algorithms and models are introduced as follows.

3.2.1 Fingerprint feature preprocessing process

The experimental data set is collected from the real environment browser, the feature dimension of the training data set is 34, and the data volume is 84. Since there is a character string type feature (hash value) in the feature, the type characteristics need to be numerically processed. The processing is shown in Algorithm 2.

In addition, due to the different meanings of different feature fields, there are too large differences in the range of different feature values (different dimensions). In order to avoid overfitting, normalization operations are required. The specific steps are shown in Algorithm 3.

Algorithm 2 String feature processing algorithm pseudo code

Input: feature String characteristicsString feature processing algorithm;
Output: Numerical features;
1: Get the string length;
2: Get every character in the string;
3: Characters use Python built-in ord function to convert characters to corresponding ASCII values;

Algorithm 3 Normalization processing algorithm pseudo code

Input: Feature Original feature;
Output: Normalized features;
1: Use standardized methods to normalize all features;
2: Value zoom;
3: Modulo calculation (0-255);

Figure 5 Original grayscale image.

Figure 6 Image classification and preservation after preprocessing.

After normalization, the value of the characteristic data is between 0–255.

On this basis, the experimental image data is generated. Considering that the feature dimension is limited, so data expansion operations are required. This expansion uses zero-padding. Finally, the feature dimension is expanded to a size of 28*28 and converted to a grayscale image. The effect is shown in Figure 5.

All data is classified according to labels, as shown in Figure 6.

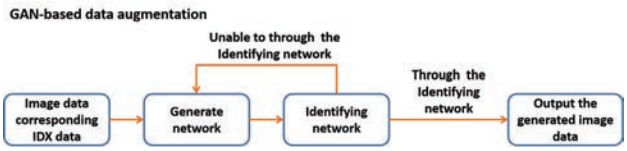

Figure 7 Data expansion based on GAN.

The generated image data needs to be converted to idx data, and the generated idx data is used as the input of the GAN model.

3.2.2 Gan (generative adversarial network) data expansion method

Since the real data collected this time is small, training data expansion is needed. This expansion uses the GAN network. The GAN network was proposed by Ian Goodfellow and others [25] in 2014, and further described in the "Generative Adversarial Networks" [26] of 2020. It consists of two parts: a generative network and a discriminant network. The generation network randomly samples from the latent space [27] as input, and its output needs to imitate the real samples in the training set as much as possible. The input of the discriminating network is the real sample or the output of the generated network. Its purpose is to distinguish the output of the generated network from the real sample as far as possible. However, the generated network should deceive and discriminate the network as much as possible. The final aim of the two networks is to make the discriminant network unable to judge whether the output result of the generated network is real or not.

The overall structure of this GAN network is shown in Figure 7.

As shown in Figure 8 is a picture data generated by GAN. This picture has passed the GAN discriminant network, indicating that it has a high similarity to the original picture. In the absence of training data in real environment, it can be used as training data to participate in training and testing of the model. At the same time, the use of GAN network can expand the experimental data on the one hand, and can also enhance the robustness of the classification model.

3.2.3 Classification model introduction

In the experiment, the CNN model, the Dilated convolution model and the residual network model are mainly used for comparison, and the experimental results of these three models are analyzed to verify the effectiveness of the

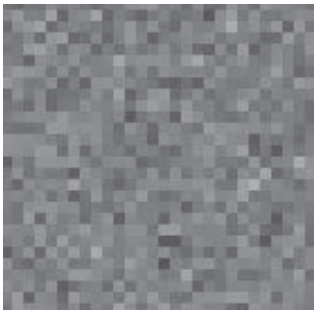

Figure 8 Grayscale image data generated by GAN.

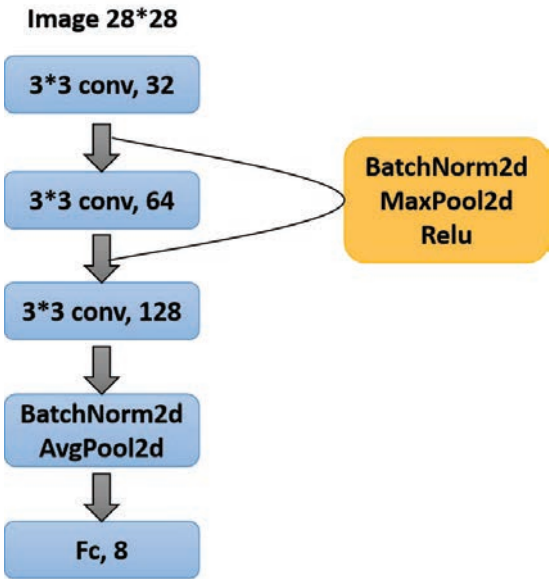

Figure 9 CNN model.

residual network-GAN method proposed in this paper.Specific experimental results will be described in the next section.

CNN Model

The CNN model used in this experiment contains three convolutional layers. The size of the convolution kernel of the first layer is 3*3, and the input is a 1-channel grayscale image with a size of 28*28. The output is 32 channels, and the picture size is 28*28.

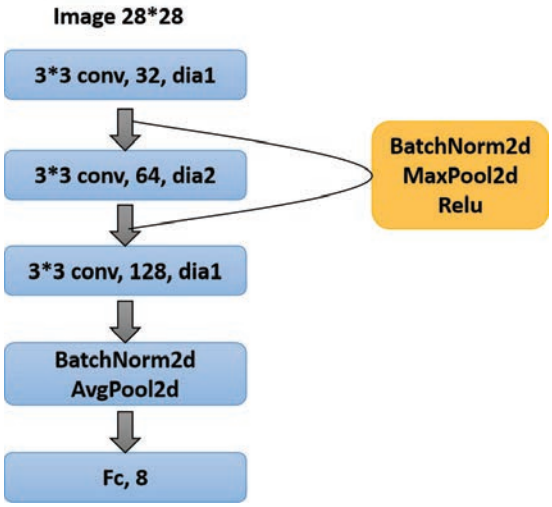

Figure 10 Dilated convolution model.

After the first layer of convolution, use BatchNorm2d to regularize the samples, and then use the ReLU function as the activation function. Then proceed to maximize pooling, and the processed image size is 13*13.

Then comes the second layer of convolution, the size of the second layer of convolution kernel is 3*3, and the image size of the 32-channel input is 13*13. The output is 64 channels, and the picture size is 13*13.

After the second layer of convolution, use BatchNorm2d to regularize the samples, and then use the ReLU function as the activation function. Then proceed to maximize pooling, and the processed image size is 6*6.

Finally, the third layer of convolution, the size of the third layer of convolution kernel is 3*3, and the image size of the 32-channel input is 6*6. The output is 128 channels, and the picture size is 6*6.

After the third layer of convolution, use BatchNorm2d to regularize the samples, and then use the ReLU function as the activation function. Then perform the average pooling process, and the processed image size is 1*1.

Then the results are mapped into categories through linear transformation.

Dilated Convolution Model

The dilated convolution model is based on the CNN model, and the dilated rate can be set on the basis of the CNN model.

The hollow convolution model used in this experiment contains three convolution layers. The size of the convolution kernel of the first layer is

3*3, and the input is a 1-channel grayscale image with a size of 28*28. The void ratio of the first layer is 1, which is a normal convolution. The output is 32 channels, and the picture size is 28*28.

After the first layer of convolution, use BatchNorm2d to regularize the samples, and then use the ReLU function as the activation function. Then proceed to maximize pooling. The processed image size is 13*13.

Then comes the second layer of convolution, the size of the second layer of convolution kernel is 3*3, and the image size of the 32-channel input is 13*13. The dilated rate of the second layer is 2, the output is 64 channels, and the picture size is 6*6.

After the second layer of convolution, use BatchNorm2d to regularize the samples, and then use the ReLU function as the activation function. Then proceed to maximize pooling. The processed image size is 2*2.

Finally, the third layer of convolution, the size of the third layer of convolution kernel is 3*3, and the image size of the 32-channel input is 2*2. The dilated ratio of the third layer is 1, which is a normal convolution. The output is 128 channels, and the picture size is 2*2.

After the third layer of convolution, use BatchNorm2d to regularize the samples, and then use the ReLU function as the activation function. Then perform the average pooling process. The processed image size is 1*1.

Then the results are mapped into categories through linear transformation.

Residual Network Model

Figure 11 shows the model of the residual network [28]. The standard input size of the residual network is 32*32. Due to the limited size of the

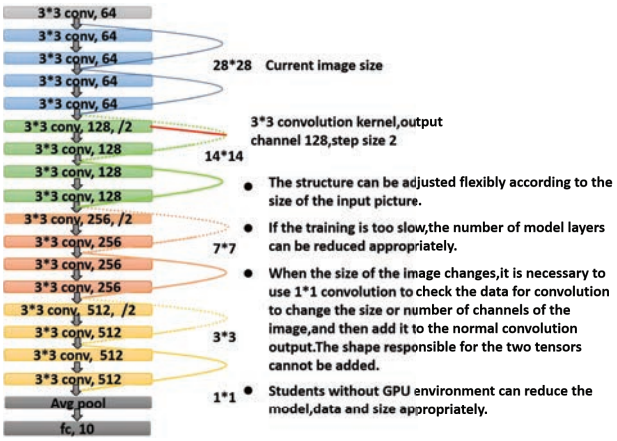

Figure 11 Residual network model.

experimental pictures in this experiment, the input size is 28*28, and the corresponding sizes are 28*28, 14*14, 7*7, 3*3, 1*1.

4 Experiment and Analysis

This section will conduct an experimental analysis of the multi-dimensional browser fingerprint detection method based on adversarial learning, and give the implementation route. The following will introduce in detail from the perspective of data set, feature processing, algorithm, etc., and finally analyze the experimental results.

4.1 Lab Environment

This experiment uses Pycharm 2020.2.1 professional (configure Pytorch environment); browser feature data set (txt_to_csv-utf8(84)); operating system is Win10 X64 system; computer is Inter i7 processor, 16G memory.

4.2 Dataset

4.2.1 Experimental data collection

The experimental data collection in this paper is to use the open source Fingerprintjs2 [29] to obtain the middle information of the user's browser, and further preprocess the collected information such as numerical processing for subsequent model training.

Fingerprintjs2 is a browser fingerprint collector implemented through pure front-end native Js. By obtaining multi-dimensional information in the browser (partially converted to String through base64), it finally generates md5, which is used for the user's unique identification code on the device.

4.2.2 Data characteristics

This step is to extract data from the collected information, and save the extracted data features as key-value pairs in the form of¡key : value¿, a total of 34 groups. Each key is a feature category of the browser, and 'value' is the value under that feature category, such as ¡language : zh-CN¿, which means that the language used by the browser is simplified Chinese.

The collected feature categories include: userAgent, webdriver, language, colorDepth, deviceMemory, pixelRatio, screenResolution, availableScreen Resolution, timezone, timezoneOffset, hardwareConcurrency, sessionStorage, localStorage, indexedDb, addBehavior, openDatabase, cpuClass,

platform, doNotTrack, plugins, canvas, webgl, webglVendorAndRenderer, adBlock, hasLiedLanguages, hasLiedResolution, hasLiedOs, hasLiedBrowser, touchSupport, fonts, fontsFlash, audio, enumerateDevices.

The main features are explained as follows:

(1) UserAgent: It is mainly composed of three parts. The first part is Mozilla/5.0. Due to the history of browser development, everyone will fill in this part by default. The second part is the platform part, which can be composed of multiple strings, such as the corresponding Windows NT10.0 It is win10, Win64; X64 means that the operating system is 64-bit, and other different operating systems have their own corresponding strings. The third part of the engine version is mainly KHTML and Gecko, and the fourth part of the browser version is related to each browser.
(2) Plugins: It shows the plugins installed by the current browser. The plugins installed by different users are likely to be different, so the number of plugins can be used as a criterion.
(3) Canvas: The same canvas element drawing operation, on different operating systems and different browsers, because different browsers use different graphics processing engines, different image export options, different default compression levels, etc., the resulting image content is also not exactly the same. Even with the same drawing operation, the CRC of the generated picture data is different.
(4) WebglGL: The principle is similar to Canvas, except that WebglGL renders 3D stereoscopic images.
(5) WebGLVendorAndRenderer: Return WebGL information, including the manufacturer of the graphics card, the model of the graphics card, etc.
(6) Audio: The subtle differences in host or browser hardware or software lead to differences in audio signal processing. The same type of browser on the same machine produces the same audio output, and the audio output produced by different machines or different browsers will be different.

4.2.3 Data preprocessing

For the collected browser data, there are mainly the following seven processing methods.

(1) If the collected data is in numerical form, no further processing will be done on it. This category includes fingerprint, colorDepth, deviceMemory, pixelRatio, hardwareConcurrency, timzoneOffset, cpuclass,

doNotTrack, audio. For this category, if the value acquisition fails that is, not available, it is recorded as 0.

(2) If the collected data is only True and False, record False as 0 and True as 1. This category includes sessionStorage, localStorage, indexedDb, addBeahvior, openDatabase, adBlock, hasLiedLanguages, hasLiedResolution, hasLiedOs, hasLiedBrowser.

(3) If the value of value can be enumerated and the number of enumeration is far less than the number of data, then the enumeration is numerically processed. This category includes language, timezone, platform, and torchSupport.

(4) If this key contains more information, divide it into multiple columns, and enumerate and digitize each column separately. This category includes userAagent, webglVendorAndRenderer.

(5) Data that has failed to collect or has no effect on model training should be excluded. This category includes webdriver and fontshash.

(6) Perform some other processing on the collected data. In this category, scrrenResolution multiplies the acquired data, availableScreenResolution multiplies the acquired data. Plugins change value to the number of plugins. Because canvas and webgl are both base64 encoded strings, we restore them first, and obtain the last piece, a 32-bit CRC check code, as our value. Fonts take the first 10 fonts as hash, and enumerateDevice as hash processing.

(7) Add a label column at the end of each piece of data. The value is from 0 to 7 corresponding to 7 browsers. The specific corresponding relationship is: 0 means chromium browser, 1 means IE browser, 2 means Edge browser, 3 Represents Firefox browser, 4 represents UC browser, 5 represents 360 browser, 6 represents Sogou browser, and 7 represents chrome browser.

4.3 Experimental Results and Analysis

In this section, there show three type of experiments. Firstly, it uses the residual network to classify the original data feature set. Secondly, there uses the residual network to classify the expanded data of the original data, which data is obtained by GAN from the original data. Thirdly, it compares the results by using three different methods.

Figure 12 is a model that uses only the residual network to classify the original data feature set collected by the experiment, and the following results are obtained. It can be seen from the figure that loss and acc are very

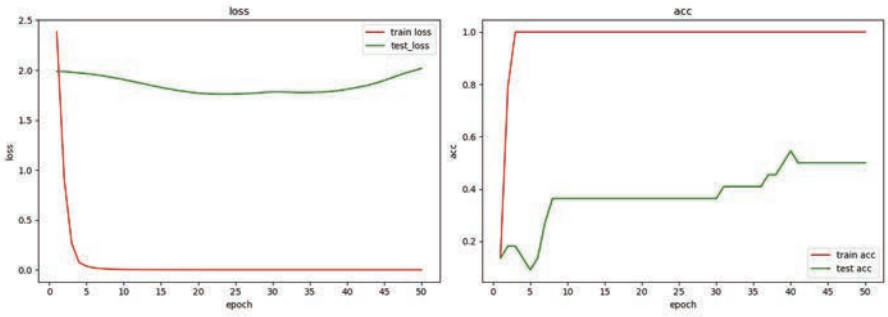

Figure 12 Residual network-source data.

unsatisfactory in the test set, and acc is in the [0.2–0.4] interval. The main reason for the analysis is that the number of data samples collected from real browsers in this experiment is too small to be adjusted to excellent model.

On the basis of the above, the fingerprint detection method based on adversarial learning is used to expand the data and enhance the robustness of the algorithm. The process is as follows:

First of all, the original data is expanded using GAN to generate 20 times the expanded data of the original data. Second, use the residual network model to classify the original data collected in the experiment and the expanded data feature set again, and get the result shown in Figure 13. In this experimental result, it can be seen that the acc of the test set is around 0.8. It can be seen from the experimental results that the detection method based on the residual network-GAN model is significantly improved compared to the single residual network model. The detection model based on adversarial learning proposed in this paper can obtain ideal detection results with a small number of samples. It also shows that this model can improve the robustness of the detection algorithm. Under the same original sample conditions, the detection model based on adversarial learning proposed in this paper is superior to other network model algorithms.

Figure 14 is the comparison result of experiments on the expanded fingerprint data using the residual network-GAN model, the convolutional network-GAN model and the dilated convolution-GAN model in the detection algorithm model based on adversarial learning proposed in this paper.

It can be seen from the comparison results that the residual network-GAN model shows a clear downward trend in the loss function index. Although there are large fluctuations in the test set due to the limited number of samples, it is compared with the convolutional network-GAN model and the

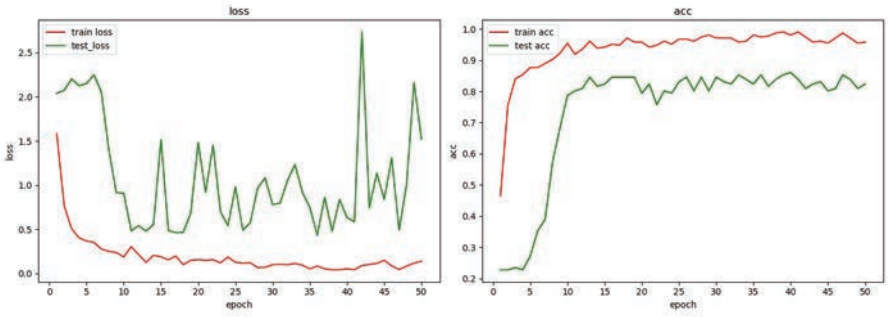

Figure 13 Residual network-GAN extended data (20 times the expanded data of the original data).

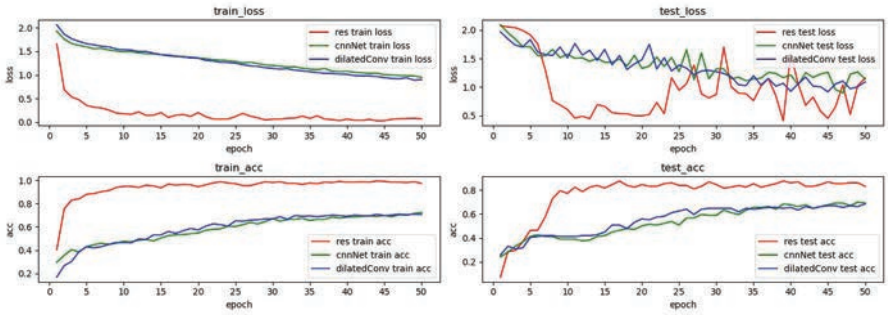

Figure 14 Experimental comparison of residual network, convolutional layer network, and dilated convolutional network.

dilated convolution-GAN model performs well. At the same time, the residual network-GAN model is also significantly better than other models in the acc index, and shows a higher accuracy rate in both the training set and the test set.

Through the above three experiments, the effectiveness of the multi-dimensional browser fingerprint detection method based on adversarial learning proposed in this paper is demonstrated. It can obtain ideal detection results with very few samples; at the same time, this method significantly improves the robustness of the detection algorithm. Because the data set generated by the GAN network can be regarded as the data in the original browser evolution set, and can be correctly classified through the trained classification network, the detection algorithm is robust in detecting the changed browser characteristics.

Although the above experiments have achieved the expected results, the number of fingerprint samples collected and other aspects need to be optimized, and the effectiveness of this work needs to be further verified.

5 Method Validation and Analysis

Because the data set samples used in the experiment in Section 4 are too few, the open source data set is further used to verify the method proposed in this paper. The following will introduce the new data set selection, feature processing and experimental verification in detail, and finally analyze the experimental results.

5.1 Lab Environment

This experiment still uses the above experimental environment, using Pycharm 2020.2.1 professional (configure Pytorch environment); the browser feature data set is replaced with (new_fpdata(7316).csv); the operating system is Win10 X64 system; the computer is Inter i7 processor, 16G memory.

5.2 Dataset

5.2.1 Experimental data collection

The data set for this comparative experiment was derived from open source information on GitHub, totaling 150,000 pieces of data. At the same time, due to the need to label the browser types in the experiment, 7316 pieces of data meeting the requirements were extracted for the subsequent steps.

5.2.2 Data characteristics

This step continues the previous operation, extracting data features and saving them as key-value pairs in the form of ¡key: value¿, a total of 30 groups. The collected feature categories include: id, acceptHttp, hostHttp, addressHttp, userAgentHttp(UserAgent_bw, UserAgent_os,UserAgent_is_pc), connectionHttp, encodingHttp, new_plugin_num, platformJS, cookiesJS, dntJS, timezoneJS, resolutionJS, localJS, sessionJS, IEDataJS, resolutionFlash, languageFlash, adBlock, vendorWebGLJS, rendererWebGLJS, octaneScore, sunspiderTime, pluginsJSHashed, canvasJSHashed, webGLJsHashed, fontsFlashHashed, osDetailed.

5.2.3 Data preprocessing

For the unprocessed original data set, the following seven processing methods are used.

(1) If the collected data is in numerical form, no further processing will be done on it. This category includes id, timezoneJS, addressHttp, and adBlock. For this category, if the value acquisition fails, that is, not available, it is recorded as 0.
(2) If the collected data is only True and False, record no as 0 and yes as 1. This category includes cookiesJS, dntJS, localJS, sessionJS, IEDataJS, adBlock.
(3) If the value of value is enumerable and the number of enumerations is far less than the number of data items, the enumeration is numerically processed. This category includes acceptHttp, hostHttp, connectionHttp, encodingHttp, platformJS, resolutionFlash, and osDetailed.
(4) If this key contains more information, divide it into multiple columns, and enumerate and digitize each column separately. This category has userAgentHttp, which is parsed into UserAgent_bw, UserAgent_os, UserAgent_is_pc.
(5) For data that fails to collect or has no effect on model training, zero-padded operations are performed on them. This category includes octaneScore and sunflowerTime.
(6) Perform some other processing on the collected data. This type of resolutionFlash multiplies the acquired data. resolutionJS multiplies the acquired data. pluginsJS changes the value to the number of plugins, which is new_plugin_num. languageFlash, vendorWebGLJS and rendererWebGLJS use LabelEncoder() to encode and convert them into digital label form.
(7) Add a label column at the end of each piece of data. The values are from 1 to 11 corresponding to 10 major browsers. The specific correspondence is: 1: chrom browser, 2: Firefox browser, 3: Chromium browser Browser, 4: Iceweasel browser, 5: Iron browser, 6: Opera browser, 7: Vivaldi browser, 8: Dragon browser, 9: Maxthon browser, 10: Pale Moon (2 Variant) browser. Finally, in order to facilitate the comparative experiment, the data of the first four browsers are retained for subsequent operations.

5.3 Experimental Results and Analysis

Because the amount of previously collected data is too small, the experimental results are quite different. Therefore, after processing the large sample

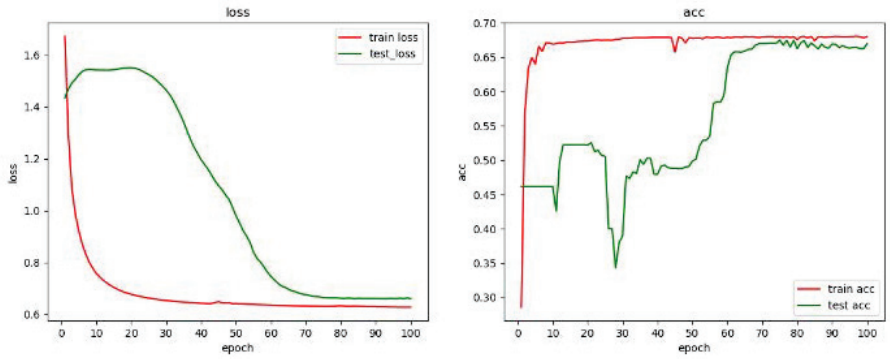

Figure 15 Residual network-source data.

data set, the residual network is used for classification, and the results are as Figure 15.

In the newly adopted data set, due to the uneven number of different types of browsers, the number of Chrom browsers and Firefox browsers with labels 1 and 2 is much higher than that of Chromium browser and Iceweasel browsers with labels 3 and 4, so The result of classification using residual network fluctuates too much. Based on this phenomenon, the multi-dimensional browser fingerprint recognition method based on adversarial learning proposed in this article is still used.

First, use GAN to expand the browsers with tags 3 and 4, and expand it to 20 times the original. Then, the experiment uses the residual network model to classify the data after GAN again, as shown in Figure 16, the results are smoothly fluctuating, and the performance of loss and acc has been partially improved compared to before.

The experimental results show that the detection method based on the residual network-GAN model can obtain better results than the single residual network model in the case of insufficient samples, reducing the fluctuation phenomenon caused by the small number of samples, which is further evidence the points raised above. This model can improve the robustness of the detection algorithm. Under the same original sample conditions, the algorithm of detection model based on adversarial learning is better than other network model algorithms.

Figure 17 shows the results of experiments on the new data set after using the residual network-GAN model, the convolutional network-GAN model and the dilated convolution-GAN model respectively. It can be seen

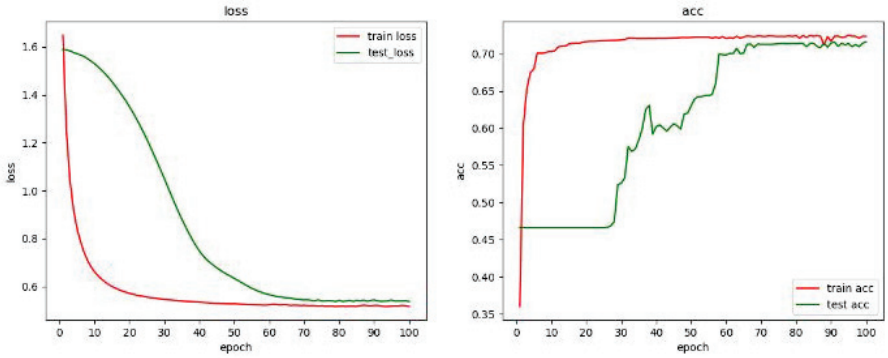

Figure 16 Residual network-GAN extended data.

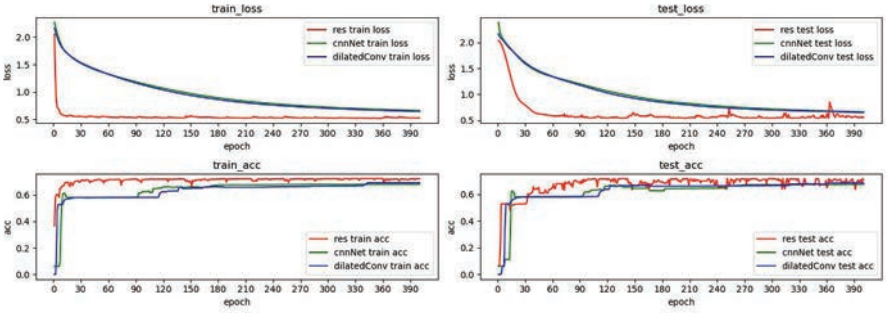

Figure 17 Residual network-GAN extended data.

from the comparison results that due to the sufficient number of samples, the fluctuations of the loss function indicators of each comparison model are significantly reduced, showing a smoother decline curve. The decline trend of the residual network-GAN model is significantly better than the convolutional network-GAN model and the dilated convolution-GAN model. At the same time, in the experimental results of the acc indicator, the gap between the three comparison models is relatively small, but the accuracy of the residual network-GAN model is still higher than other models.

The above comparative experiments show that the method proposed in this paper still has better results after replacing the data set. In the case of a small number of samples or large differences in the number of samples, the use of GAN networks can still improve the effectiveness and robustness of the detection algorithm.

6 Conclusion

Aiming at the potential security problems in Web service access, especially the requirements for hierarchical data and fine-grained data access control in Web services with high security requirements, this paper proposes a cross-server and browser-side Web combined with browser fingerprint detection. Service access framework, and designed two of the key models an access control model based on identity and tags and a multi-dimensional browser fingerprint detection method based on adversarial learning. Through experiments and analysis, the robustness of browser fingerprint detection is solved. And through the matching of users and data tags to achieve the mapping relationship between entities and data resources and other issues. At the same time, in order to verify the effectiveness of the model, this paper uses other open source data set, and experiments after processing them to prove the effectiveness of the method. The work of this paper is suitable for Web service access control requirements in high security scenarios, and is an effective supplement to the existing authentication methods, and it is feasible.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities 2019YJS033 and the Research Funds of NSTEC under Grant K20GY500010.

References

[1] C. Administration, "Cyberspace administration of china," http://www.cac.gov.cn/2021-02/03/c_1613923423079314.htm.

[2] H. Wang, Y. Zhang, J. Li, and D. Gu, "The achilles heel of oauth: A multi-platform study of oauth-based authentication," in *Proceedings of the 32nd Annual Conference on Computer Security Applications*, ser. ACSAC '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 167–176. [Online]. Available: https://doi.org/10.1145/2991079.2991105

[3] A. Armando, R. Carbone, L. Compagna, J. Cuéllar, and L. Tobarra, "Formal analysis of SAML 2.0 web browser single sign-on: breaking the saml-based single sign-on for google apps," in *Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering, FMSE 2008, Alexandria, VA, USA, October 27, 2008*, V. Shmatikov, Ed. ACM,

2008, pp. 1–10. [Online]. Available: https://doi.org/10.1145/1456396.1456397

[4] D. Perito, C. Castelluccia, M. A. Kâafar, and P. Manils, "How unique and traceable are usernames?" *CoRR*, vol. abs/1101.5578, 2011. [Online]. Available: http://arxiv.org/abs/1101.5578

[5] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine, "Browser fingerprinting: A survey," *CoRR*, vol. abs/1905.01051, 2019. [Online]. Available: http://arxiv.org/abs/1905.01051

[6] A. FaizKhademi, M. Zulkernine, and K. Weldemariam, "Fpguard: Detection and prevention of browser fingerprinting," in *Data and Applications Security and Privacy XXIX - 29th Annual IFIP WG 11.3 Working Conference, DBSec 2015, Fairfax, VA, USA, July 13-15, 2015, Proceedings*, ser. Lecture Notes in Computer Science, P. Samarati, Ed., vol. 9149. Springer, 2015, pp. 293–308. [Online]. Available: https://doi.org/10.1007/978-3-319-20810-7_21

[7] C. F. Torres, H. L. Jonker, and S. Mauw, "Fp-block: Usable web privacy by controlling browser fingerprinting," in *Computer Security – ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II*, ser. Lecture Notes in Computer Science, G. Pernul, P. Y. A. Ryan, and E. R. Weippl, Eds., vol. 9327. Springer, 2015, pp. 3–19. [Online]. Available: https://doi.org/10.1007/978-3-319-24177-7_1

[8] U. Fiore, A. Castiglione, A. D. Santis, and F. Palmieri, "Countering browser fingerprinting techniques: Constructing a fake profile with google chrome," in *17th International Conference on Network-Based Information Systems, NBiS 2014, Salerno, Italy, September 10-12, 2014*, L. Barolli, F. Xhafa, M. Takizawa, T. Enokido, A. Castiglione, and A. D. Santis, Eds. IEEE Computer Society, 2014, pp. 355–360. [Online]. Available: https://doi.org/10.1109/NBiS.2014.102

[9] Y. Cao, S. Li, and E. Wijmans, "(cross-)browser fingerprinting via OS and hardware level features," in *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, February 26–March 1, 2017*. The Internet Society, 2017. [Online]. Available: https://www.ndss-symposium.org/ndss2017/ndss-2017-programme/cross-browser-fingerprinting-os-and-hardware-level-features/

[10] A. Vastel, P. Laperdrix, W. Rudametkin, and R. Rouvoy, "FP-STALKER: tracking browser fingerprint evolutions," in *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21–23 May 2018, San Francisco, California, USA*. IEEE Computer Society,

2018, pp. 728–741. [Online]. Available: https://doi.org/10.1109/SP.2018.00008

[11] E. Bursztein, A. Malyshev, T. Pietraszek, and K. Thomas, "Picasso: Lightweight device class fingerprinting for web clients," in *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices, SPSM@CCS 2016, Vienna, Austria, October 24, 2016*, L. Lu and M. Mannan, Eds. ACM, 2016, pp. 93–102. [Online]. Available: http://dl.acm.org/citation.cfm?id=2994467

[12] F. Rochet, K. Efthymiadis, F. Koeune, and O. Pereira, "SWAT: seamless web authentication technology," in *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13–17, 2019*, L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, and L. Zia, Eds. ACM, 2019, pp. 1579–1589. [Online]. Available: https://doi.org/10.1145/3308558.3313637

[13] secureauth, "Device and browser fingerprinting," https://docs.secureauth.com/pages/viewpage.action?pageId=40045162.

[14] D. Ferraiolo, J. Cugini, and D. Kuhn, "Role-based access control: features and motivations," 01 1995.

[15] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996. [Online]. Available: https://doi.org/10.1109/2.485845

[16] W. Liu, H. Duan, H. Zhang, P. Ren, and J. Wu, "Trbac: Trust based access control model," *Jisuanji Yanjiu yu Fazhan/Computer Research and Development*, vol. 48, no. 8, pp. 1414–1420, 2011, discretionary access control; Dynamic access control; Mandatory access control; Role-based Access Control; Secure operating system; Trust; Trust computing; Trust-based access control.

[17] C. E. da Silva, J. D. S. da Silva, C. Paterson, and R. Calinescu, "Self-adaptive role-based access control for business processes," in *12th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2017, Buenos Aires, Argentina, May 22–23, 2017*. IEEE Computer Society, 2017, pp. 193–203. [Online]. Available: https://doi.org/10.1109/SEAMS.2017.13

[18] W. K. Grassmann, "Markov modelling," in *Proceedings of the 15th Conference on Winter Simulation – Volume 2*, ser. WSC '83. IEEE Press, 1983, pp. 613–619.

[19] A. Yavari, A. S. Panah, D. Georgakopoulos, P. P. Jayaraman, and R. G. van Schyndel, "Scalable role-based data disclosure control for the internet of things," in *37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017, Atlanta, GA, USA, June 5–8, 2017*, K. Lee and L. Liu, Eds. IEEE Computer Society, 2017, pp. 2226–2233. [Online]. Available: https://doi.org/10.1109/ICDCS.2017.307

[20] M. U. Aftab, Z. Qin, N. W. Hundera, O. Ariyo, Zakria, N. T. Son, and T. V. Dinh, "Permission-based separation of duty in dynamic role-based access control model," *Symmetry*, vol. 11, no. 5, p. 669, 2019. [Online]. Available: https://doi.org/10.3390/sym11050669

[21] D. D. F. Maesa, P. Mori, and L. Ricci, "Blockchain based access control," in *Distributed Applications and Interoperable Systems – 17th IFIP WG 6.1 International Conference, DAIS 2017, Held as Part of the 12th International Federated Conference on Distributed Computing Techniques, DisCoTec 2017, Neuchâtel, Switzerland, June 19–22, 2017, Proceedings*, ser. Lecture Notes in Computer Science, L. Y. Chen and H. P. Reiser, Eds., vol. 10320. Springer, 2017, pp. 206–220. [Online]. Available: https://doi.org/10.1007/978-3-319-59665-5_15

[22] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart iot-based healthcare big data storage and self-adaptive access control system," *Inf. Sci.*, vol. 479, pp. 567–592, 2019. [Online]. Available: https://doi.org/10.1016/j.ins.2018.02.005

[23] S. Englehardt and A. Narayanan, "Online tracking: A 1-million-site measurement and analysis," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24–28, 2016*, E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, Eds. ACM, 2016, pp. 1388–1401. [Online]. Available: https://doi.org/10.1145/2976749.2978313

[24] I. Sánchez-Rola, I. Santos, and D. Balzarotti, "Extension breakdown: Security analysis of browsers extension resources control policies," in *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16–18, 2017*, E. Kirda and T. Ristenpart, Eds. USENIX Association, 2017, pp. 679–694. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/sanchez-rola

[25] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December*

8–13 2014, Montreal, Quebec, Canada, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020. [Online]. Available: https://doi.org/10.1145/3422622

[27] F. Monay and D. Gatica-Perez, "On image auto-annotation with latent space models," in *Proceedings of the Eleventh ACM International Conference on Multimedia, Berkeley, CA, USA, November 2–8, 2003*, L. A. Rowe, H. M. Vin, T. Plagemann, P. J. Shenoy, and J. R. Smith, Eds. ACM, 2003, pp. 275–278. [Online]. Available: https://doi.org/10.1145/957013.957070

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27–30, 2016*. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[29] github, "fingerprintjs," https://github.com/fingerprintjs/fingerprintjs.

Biographies

Liu Hui received his B.Sc. degrees in Computer Science and Technology from Hunan University, China; M.Sc. degree in Computer Science and Technology from Huazhong University of Science and Technology, China; Now, Liu Hui is a Ph.D. candidate in Beijing Jiaotong University, China; His research field of centers on information security.

He Xudong received his B.Sc. degrees in Dalian Jiaotong University, China; He Xudong is a Ph.D. candidate in Computer Technology from Beijing Jiaotong University, China; His main research field are Web security, Internet of Things security and blockchain.

Gao Fan received her B.Sc. degrees in computer science and Technology from Shandong University of Technology, China; M.Sc. degree in Computer Technology from Beijing Jiaotong University, China; Her main research field are Web security and browser fingerprint detection.

Wang KaiLun received his B.Sc. degrees in Beijing University of Technology. He is currently studying for a master's degree in the school of computing and information technology of Beijing Jiaotong University. His current research interests include Internet of Things and blockchain.

Enze Yuan received his B.E. degree in information security from Beijing Jiaotong University, Beijing, China, in 2020, He is currently pursuing the MA.Eng degree in cyberspace security at Beijing Jiaotong University. His research interests include Internet of things and blockchain.