Journal of Computers Vol. 34 No. 4, August 2023, pp. 215-232
doi: 10.53106/199115992023083404019

A Dynamic and Fine-Grained User Trust Evaluation Model for Micro-Segmentation Cloud Computing Environment

Chaoqun Kang[1], Erxia Li[1], Dongxiao Liu[2,3], Xinhong You[4], Xiaoyong Li[2,3*]

[1] State Grid Shanghai Energy Interaction Research Institute Co., LTD, Shanghai, China
15210800793@163.com, lierxia@epri.sgcc.cpm.cn

[2] Key Laboratory of Trustworthy Distributed Computing and Service (BUPT), Ministry of Education, Beijing, 100876, China
{liudongxiao, lixiaoyong}@bupt.edu.cn

[3] School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing, 100876, China
[4] State Grid Shandong Electric Power Research Institute, Jinan, China
youxh93@163.com

Received 1 July 2023; Revised 20 July 2023; Accepted 25 July 2023

Abstract. With the diversity and complexity of user access behaviors in the "micro-segmentation" cloud computing environment, it is no longer possible to control unauthorized access of authorized users by only relying on user identity login authentication to control user access to cloud resources. The existing trust evaluation methods can not cope with the characteristics of "micro-isolated" cloud environment, which is characterized by high granularity of resources, increasing number of users' access requests and rapid changes. Based on the zero-trust principle of "Never trust, al-ways verify", we propose a dynamic, fine-grained user trust evaluation model for micro-segmentation cloud computing environment, which combines multiple user trust attributes and leverages the subjective-objective approach to assign weights to trust attribute indicators to achieve dynamic scoring of users' real-time behaviors. To capture the characteristics of users' intrinsic behaviors, we use correlation analysis to identify the correlation between users' current and historical behaviors, and combine sliding windows and penalty functions to optimize the model. The massive simulation experiments demonstrate the effectiveness of the proposed dynamic and fine-grained method, which can effectively combine the intrinsic correlation of users' own access behavior and the difference of access behavior among different users.

Keywords: dynamic trust evaluation, association analysis, micro-segmentation, cloud computing

1 Introduction

Cloud computing is an open, service-oriented, and utility-oriented dynamic computing paradigm [1, 2], which has given rise to a large number of new business models, and technologies such as telecommuting [3] and resource storage virtualization [4]. With further growth of business, concepts such as insider threats [5, 6], security and privacy of sensitive data, and zero trust [7, 8] have come back into the limelight. The massive end devices and extensive access to users in the cloud environment have increased the exposure of the network, posing a serious challenge to the traditional protection system characterized by boundary segmentation. "Micro-segmentation" [9, 10] is a security zone created in data centers and cloud deployments, where systems and software are virtualized through re-source virtualization technology and made available to the public as cloud services on the Internet, which can be accessed by anyone through reasonable means. The core capability of the "micro-segmentation" architecture focuses on the segmentation of east-west traffic. By deploying resources in containers, the borderless network is partitioned into logically tiny segments, providing maximum segmentation and segmentation of resources, making network resource security access more granular, and achieving protection of virtualized resources on the cloud.

Traditional authentication and access control [11, 12] mostly use the "once authenticated, once authorized, long term" approach to ensure normal interaction between legitimate users and the cloud system, but this security defense system and technology are still not effective in preventing threats from within, that is, the danger of malicious behavior from the authorized user level. These malicious behaviors include stealing resources stored

* Corresponding Author

in the cloud, seizing memory space, and so on. In traditional cloud computing environments, in order to address user violations, illegal operations, and other types of abnormal behavior, many researchers or practitioners use existing manual labeled data sets to evaluate user operations through machine learning, deep learning, and other methods to timely discover abnormal user behavior, but this method requires the use of the corresponding environment of user behavior data sets, which has certain timeliness. This makes it difficult to migrate machine learning or deep learning models across different cloud ring mirrors. At the same time, rule-based trust evaluation enforces strict restrictions on users and is not adaptive, and this black-and-white list-like approach is very unfriendly to diverse access methods (different networks or devices) and benign user misuse (wrong password, etc.).

Trust evaluation for user behavior is the core technology to solve the security problem in the cloud environment, which provides a good model to improve the security of the cloud system, it scores the user's access behavior through a series of trust evaluation indicators, and users with low trust will be bound to some extent. In the cloud computing environment under "micro-segmentation" architecture, dynamic trust evaluation of users is a challenging task for three main reasons: (1) users' operating habits gradually become more random, and access behavior characteristics more diverse, which is difficult to extract and analyze. (2) The evaluation only from the user's access features is too single and does not consider the intrinsic relationship between the features. (3) The assignment of trust evaluation indexes is highly subjective and lacks certain scientific rigor. Therefore, how to analyze and evaluate the massive user access behavior, ensure the security of cloud resources in a complex and dynamic environment, dynamically and fine-grained trust evaluation of users, and how scientifically assign the trust evaluation indexes has become a crucial issue.

Zero Trust is a resource protection-focused and trust-based cybersecurity paradigm in which access subjects are never implicitly granted based solely on location information, but must be continuously assessed and privileges assigned based on comprehensive trust. Based on the principle of "never trust, always verify", this paper proposes a dynamic and fine-grained end-user trust evaluation model for "micro-segmentation" cloud environment, combined with serial correlation analysis and a combination of subjective and objective trust assignment methods. The dynamic and fine-grained scoring of each user access request is based on sliding window adaptive extraction of recent user access behavior features, which enables continuous user trust records and guarantees secure access to guest resources in the "micro-segmentation" cloud environment. The main contributions of this paper can be summarized as follows:

- A dynamic and fine-grained user trust evaluation model based on zero trust for micro-segmentation cloud environments is proposed.
- A trust scoring method which combines subjective and objective trust attribute assignment and intrinsic association behavior analysis is proposed.
- The massive simulation experimental results demonstrated the effectiveness of the proposed dynamic and fine-grained trust evaluation model.

The organization of this paper is as follows. Section 2 introduces the existing research proposals at home and abroad. Section 3 describes the proposed dynamic and fine-grained trust evaluation model. Section 4 describes the experiments and analysis. Section 5 summarizes the paper and describes the future work.

2 Related Work

Micro-segmentation technology is implemented through container-based virtualization technology represented by Docker [13]. Traditional virtualization technologies, such as Xen, KVM and VMware [14, 15], achieve complete virtualization of the system by encapsulating the software layer, emulating the full privileged instructions of the system and the complete underlying hardware environment [16, 17]. The virtualization implemented by Docker is known as OS-level virtualization [18] or container-based virtualization, which packages applications with all dependencies into lightweight containers that share the system kernel of the host. In this paper, we study the user trust evaluation problem in a "micro-segmentation" cloud environment, which is characterized by a high degree of granularity of resources and poses a more serious challenge for trust evaluation and access control, and we will analyze both user behavior analysis and trust evaluation methods.

2.1 User Behavior Analysis

In the "micro-segmentation" cloud environment, since the user's behavior is related to personal habits, their operation habits gradually become more random and their access behavior characteristics are more diverse, which makes it difficult to extract and analyze and establish evaluation user indicators. Mujawar et al. [19] proposed a method to evaluate the trustworthiness of cloud service providers based on their behavior and user feedback; by considering various service quality attributes to calculate behavioral trust values and using different parameters of service level agreements to calculate feedback trust value, which has implications for scoring user trust. Ladekar et al. [20] used the Apriori algorithm to extract meaningful user browsing behavior information through Web log data for mining and complete user authentication by comparing user historical behavior information. This method helps to mine and analyze the user's access behavior patterns but has a large time overhead. Khilar et al. [21] solved the problem of registering members with multiple user names in a cloud computing environment by incorporating user identifiers and mac addresses into the user behavior trust evaluation process and providing information on user behavior trust values. Also, the method considers the relationship between new and past behaviors in terms of user behavior analysis, which is useful for identifying malicious users and negative behaviors and can prevent the recurrence of malicious behaviors. Zhang et al. [22] proposed a trust model based on a double-blind anonymous evaluation to anonymously match cloud service providers and users based on user requirements. It can be used to effectively deal with some malicious attacks aimed at distorting trust evaluation and stopping users from colluding to deceive. Raid et al. [23] proposed a hierarchical access control scheme with dynamic revocation threshold vectors that can manage users based on final trust level and network threat level, enabling dynamic hierarchical metrics for users under multi-authority systems. Zheng et al. [24] used a neural network clustering approach to detect abnormal user behavior, which ensures that the feature information is not overfitted in the clustering analysis and similarity calculation, and can effectively identify users as well as assess the trustworthiness of their behavior. This scheme has higher detection speed and clustering accuracy than traditional schemes and is more suitable for establishing mutual trust between users and clouds in mobile cloud environments.

2.2 User Trust Evaluation

The trust evaluation provides a comprehensive score based on the user's access behavior characteristics to provide a basis for access control of the system. The scoring needs to influence the evaluation algorithm of trustworthiness metrics and the selection of evaluation indicators with full consideration so that the trust scoring can be more dynamically adapted to the user behavior changes and avoid the problem of too much subjectivity. Many works use statistical methods for trust calculation [25], such as Dempster-Shafer theory (DST) [26, 27], and Bayesian inference (BI) [28, 29]. Yang et al. [30] constructed a dynamic access control mechanism based on short-term tokens and user trust, which is based on a deep convolutional neural network model to analyze user behavior and combine user trustworthiness set the token update method to achieve dynamic access authorization, which is good for improving system security. Paul et al. [31] proposed a trust-based access control model in a cloud environment, which uses role-based access control policies for users and modifies them based on their trust level to achieve dynamic trust evaluation. Hosseini et al. [32] used decision trees, logistic regression, plain Bayesian, and various other machine learning methods of trust evaluation strategies to predict the trust values of users and resources. Yang et al. [33] proposed an integrated model-based trust evaluation method for user behavior, namely ordered logit regression, by combining the advantages of statistical methods and hierarchical analysis. The advantage of this method is that it is beneficial for cloud service providers to quickly and accurately evaluate the trustworthiness of user behavior, but the disadvantage is that the analysis of massive and complex data is not ideal. Li et al. [34] proposed a dynamic trust evaluation method based on cloud user behavior, which is based on the analysis of historical user behavior data and reflects the basic laws of user behavior by entropy method, which to a certain extent weakens the subjectivity of evaluation results, improves the recognition rate of abnormal user behavior, and realizes the dynamism of trust evaluation.

3 Dynamic and Fine-Grained Trust Evaluation Model

This chapter introduces the trust evaluation model for users and the methods used in it. The model achieves trust

evaluation for users by extracting their access behavior sequences to achieve secure cloud resource access. It is specifically divided into three parts: trust evaluation model establishment, trust attribute index selection, trust weight assignment, and trust score calculation.

3.1 The Architecture of the Proposed Model

In a "micro-segmentation" cloud environment, resources are divided and protected in a fine-grained manner. As shown in Fig. 1, each resource is deployed in a docker container, and proxy software is deployed in docker to protect and control the resources within docker. The trust evaluation model is deployed in the trust node, which is distributed according to the number and importance of virtual resources, and the node and the virtual resources in its control form a trust domain. At the boundary of the "micro-segmentation" cloud environment, there is a trust center that identifies the resource type of the access object (i.e., user) request and enables triage of access requests. Each trust node evaluates the scores of access requests in its control and serves as an important basis for access control by the trust center.

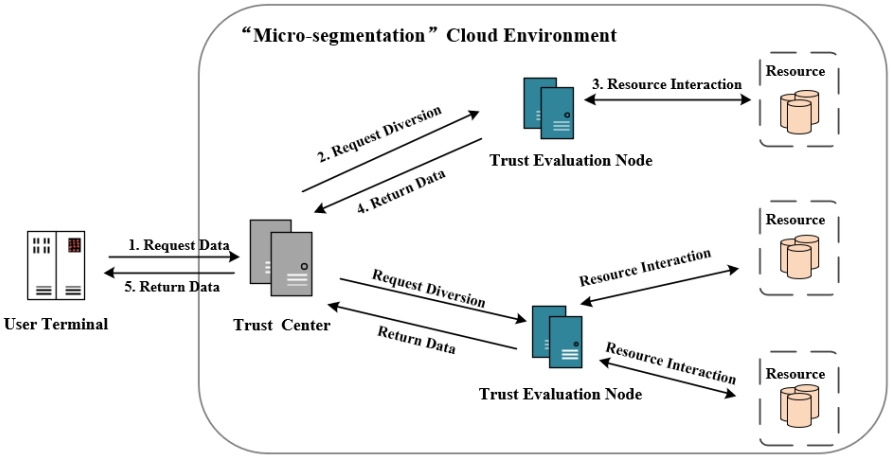


Fig. 1. The resource request flow of users in a "micro-segmentation" cloud environment.

The traditional predefined static trust scoring model can no longer meet the security requirements of the above environment "micro-segmentation" cloud environment. Thus, we combine the mentioned sequence correlation analysis, subjective and objective trust assignment methods, and trust calculation methods to design a trust evaluation model for end-user behavior in the "micro-segmentation" cloud environment, which can adaptively score the user terminal dynamically and fine-grained for user access behavior characteristics continuously.

The dynamic and fine-grained trust evaluation model is shown in Fig. 2, which consists of five modules: the user behavior sequence processing module, association rule matching module, dynamic assignment of trust attribute weights module, time decay and parameter correction module and trust score aggregation module. As shown in Algorithm 1, the trust evaluation model will combine real-time and historical access sequences for trust score calculation, and a complete trust evaluation process is as follows:

(1) The trust center receives a request from the access object (i.e., user), triages the request according to the type of resource access, and transmits it to the corresponding trust evaluation node.

(2) The trust evaluation node receives a user access request from the trust center, converts it into a sequence of real-time access behaviors according to the trust attributes, determines the access subject of the sequence, and adds the sequence to the historical access behavior database of the user.

(3) The sliding window of the history access database advances one frame, reads the user's history behavior sequence, and forms a form for the history behavior trust calculation module of the trust evaluation model.

(4) The historical behavior trust calculation module uses the improved FP-Growth [35] algorithm to calculate the set of frequent item set and calculates the trust score of the current access sequence based on the confidence level.

(5) The real-time behavioral trust calculation module scores the trust attributes based on the trust calculation function and combines them into sequences. Then, the weights obtained by the combination of AHP (Analytic Hierarchy Process) [36] and CRITIC (Criteria Importance Through Intercriteria Correlation) [37] are multiplied to obtain the trust score of real-time access behavior.

(6) The time decay and parameter correction module decay the trust score according to the user login interval and also performs a subtraction correction of the trust score according to the type and importance of the resource.

(7) The trust evaluation model adds up the scores obtained from the three trust calculation modules to get the comprehensive trust score of the user's current visit.

(8) The trust evaluation node decides whether to pass the resource data to the trust center and finally to the user based on the comprehensive trust score of that user.

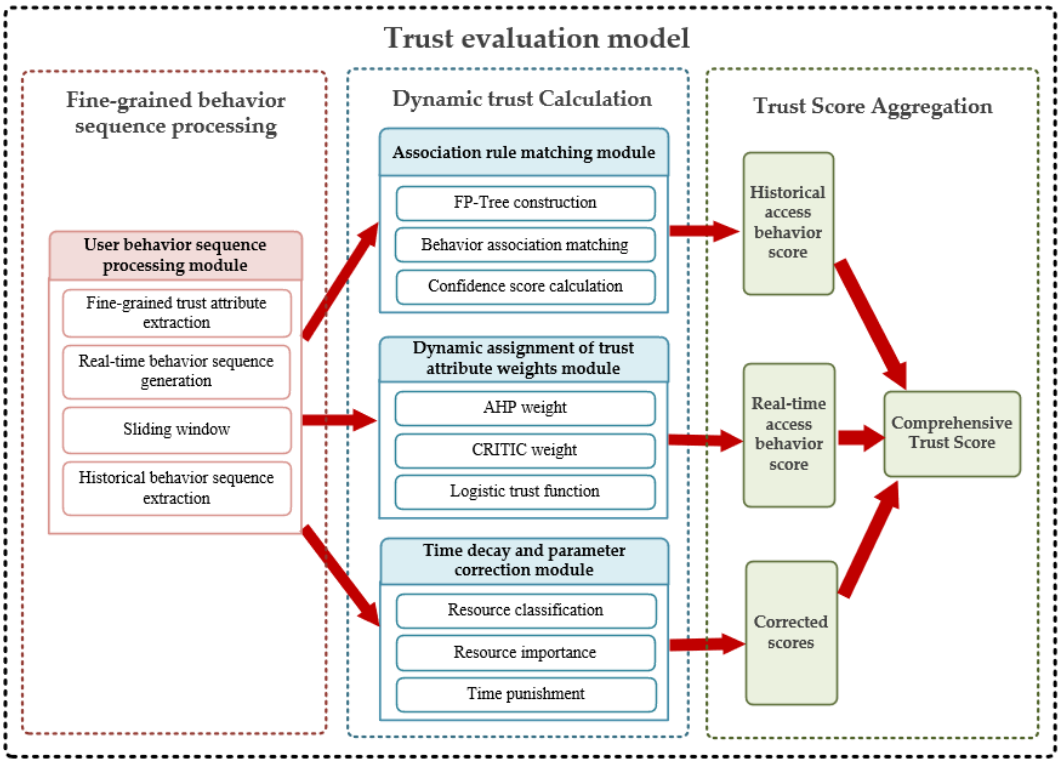


Fig. 2. The dynamic and fine-grained trust evaluation model

Algorithm 1. Dynamic trust evaluation and control algorithm for users

Input: user access request r.

Initialization: sliding window size k, two-by-two comparison judgment matrix $(A_{ij})_{n\times n}$ trust calculation function f, trust growth rate parameter α, trust path length parameter β, real-time behavioral trust weights θ_1, historical behavior trust weights θ_2,correction parameter θ_3.

Output: users' trust evaluation scores

For each user access request r arrives at Trust center **do**
  Extract the requested resource and transmit r to Trust node;
  For each Trust node receives r **do**
    Extract the trust attribute and structural behavior sequence I_n;
    Extract user's historical behavior sequences based on sliding window size s
    // **Real-time behavioral trust score calculation**
    Calculation of weights w_s using Equation (2);
    Calculation of weights w_O using Equation (8);
    Calculation of G_t using Equation (12);

9: Calculation of G_r using Equation (13).
// **Historical behavioral trust calculation**
10: Load the FP tree or building FP trees using the FP-Growth algorithm;
11: Construct frequent item set with the accessed resource as a suffix;
12: Match the association of this user access sequence with the frequent item set;
13: Calculate of the confidence degree as G_h using Equation (14).
// **User comprehensive trust score calculation**
14: Calculate of the user comprehensive trust score using Equation (10).
15: **end For**
16: Trust node records the current user trust score;
17: Trust nodes decide whether to deliver resource data based on the trust score.
18: **end For**
19: **If** Trust center receives the resource data **then**
20: **Return** resource data to the user.
21: **end If**

3.2 The Calculation of Fine-Grained Trust Indicators

The establishment of trust indicators is crucial in the trust evaluation work, and the user's behavioral indicators are the original basis of trust evaluation, and the goodness of the indicators will directly affect the effectiveness and practicality of the trust evaluation model. The trust evaluation of users in a cloud environment mainly relies on access behavior. According to life experience, each person has his or her own relatively stable behavior pattern in a normal state, and the same is true for users accessing cloud resources. Users with normal behavior history will also have relatively stable operation habits and will perform normal activities within the standard authority range after normal login. In this paper, we combine the characteristics of virtualized resources on the cloud and user terminal access, and select trust indicators from four perspectives: user login process, access time to resources, resource usage expectation and user historical behavior, as shown in Table 1. In the trust evaluation, we consider a user is trustworthy when the specified operation, specified access frequency, specified data volume, and specified request frequency match the expected range under this user role, and the trust indicators give the matching degree of user access behavior at the corresponding indicator points to support the trust scoring. In the table, IP_{login}/ MAC_{login} is the number of IP/MAC addresses in the database that match the IP address used by the user to current login, IP_{all}/ MAC_{all} is the number of all IP/MAC addresses recorded in the database. Th_{IP} and Th_{MAC} frequency threshold set by the administrator. NF_{login} is the number of authentication failures for current user during the current login. $NF_{average}$ is the historical average number of authentication failures for all users. T_{login} is the time interval of current login for current user. T_{common} is the common login time interval for current user.

Table 1. The fine-grained trust indicators

Category	Indicator name	The content of trust indicators
The evaluation of login process	IP address	$IP_{login}/ IP_{all} < TH_{IP}$
	Mac address	$MAC_{login}/ MAC_{all} < Th_{MAC}$
	Login certification	$NF_{login} < NF_{average}$
The evaluation of login time	Login time	$T_{login} \in T_{common}$
The evaluation of resource utilization	The name of resource	Label of the resource stored in the cloud environment
	The classification of resource	The classification of resources, and related to the user role, interaction mode
	The importance of resource	The relative importance of the resource in the cloud environment
	The confidence of behavioral associations	The association of each trust attribute with the accessed resource when the user accesses the resource
The evaluation of user historical behavior	Trust score	The historical score value of user behavior
	Behavior list	List of user history access behaviors

The Calculation of Fine-Grained Trust Indicators. Authentication failures are common among users in various scenarios, such as inaccurate input, or forgetting login credentials. The frequency of such failures varies,

and prior research [38] indicates that the average failure rate of user prelogin authentication ranges from 20% to 30%. Furthermore, the client terminal of cloud services is often used in diverse and unrestricted environments, allowing cloud terminal users to log in through different devices and networks, including public WiFi, which can compromise the security of the terminal. In the event that a user's account and password are compromised, either by being stored on a portable device or stolen by others, the stored cloud resources can be easily accessed by different devices in different locations using the real user's name.

We assume that each user keeps their activity within a certain rigid range every day. Although the sources of requested addresses may vary greatly from user to user, e.g., office-based users usually use fixed IP addresses and fixed devices, and frequent travelers usually use different networks and devices, the number of their different IP and MAC address values over a certain period of time remains relatively constant.

The Evaluation of Login Time. Users usually have a regular work schedule. For example, in the production environment of many enterprises, a "9:00 to 17:00" work pattern is adopted, where Employee A usually obtains the data of the day's work in the morning, Leader B will check the work reports submitted by employees in the afternoon of every Friday, and Employee C will routinely maintain and check the resources every night. The request time shows a certain regularity, and the login time overlaps or partially overlaps with the high frequency access time in their history. Therefore, in the design of the login time index, we divide the time into four segments: 23:00-6:00, 6:00-12:00, 12:00-17:00, 17:00-23:00, according to the time of user access behavior, the length of time, the distribution pattern of the access system, and the actual life situation, to ensure that the "time slice" as fine-grained as possible while not allowing this division to become complex and trivial.

The Evaluation of Resource Utilization. In the "micro-segmentation" cloud environment, the ultimate goal of trust evaluation is to control whether users are allowed to access fine-grained resources, so the types of resources accessed by users (e.g., browsing resources or interactive resources) and the importance of the accessed resources need to be recorded. At the same time, users' resource access behavior will show some regularity, and they will always access relatively fixed types of resources with some relatively fixed access methods. For example, user D will typically connect to the WiFi network at home in the morning and use his laptop to access log resources on the cloud. In this scenario, the behavioral access sequence consisting of access time, IP address, MAC address, and resource type will form an association. In trust evaluation, if an entity always achieves the desired goal, the association of individual trust attributes appears frequently, it forms an expectation that the access to that user is trusted. At the level of user behavior, different users have different "access expectations", which are always related to the user's role. Therefore, the consideration of this association can be an important indicator of user behavior.

The Evaluation of User Historical Behavior. In the trust evaluation model, the database records the users' historical resource access data, behavior sequences, and trust scores. The user's historical trust behavior records reflect the changes in the user's access behavior, and the historical trust score reflects the user's association with his historical behavior. The trust evaluation of user access behavior is a dynamic evaluation process based on historical scores. To ensure a more scientific trust score, the trust score of users with low trust values grows slowly, while the trust score of users who have accumulated a certain trust value can rise rapidly because of their good behavior. Therefore, it is necessary to consider incorporating the user's historical trust evaluation records into the trust factors of cloud terminal user behavior.

3.3 Dynamic Trust Weight Allocation

Trust evaluation is a multicriteria comprehensive evaluation problem, which requires scientific weight of each trust indicator. In this paper, based on the characteristics of the access request sequence, a combination of subjective and objective methods is adopted to assign weights to trust scores. For an access behavior sequence I_n with K trust attributes, the weight of the trust attribute can be obtained by multiplying subjective weight w_s and objective weight w_o.

Subjective Weight Calculation. The subjective weights are calculated using the AHP method, which obtains the weights of relative importance by objectively providing a quantitative description of the importance of the same hierarchical element compared two by two. This quantitative description is an expert scoring model based on ex-

perience. The subjective weights are calculated using the canonical column averaging method of the Hierarchical Analysis Method for weight assignment in the following steps:

(1) For a two-by-two comparison judgment matrix $A = (A_{ij})^{n\times n}$, normalize each columnvector of A. The elements w_{ij} of the *i-th* row and *j-th* column is converted to $\tilde{w}_{ij}$ $\tilde{w}_{ij}$ can be calculated as:

$$\tilde{w}_{ij} = a_{ij} / \sum_{i=1}^{n} a_{ij}, \tag{1}$$

where n is the number of trust attributes, a_{ij} is the relative importance of the *i-th* and *j-th* attributes.

(2) Summing $\tilde{w}_{ij}$ by rows and normalizing. Subjective weight w_s can be calculated as:

$$w_s = \sum_{j=1}^{n} \tilde{w}_{ij} / \sum_{i=1}^{n} \tilde{w}_i. \tag{2}$$

Objective Weight Calculation. The objective weights were calculated using the CRITIC method, based on the two concepts of contrast intensity and conflicting characteristics of the evaluated indicators. The contrast strength indicates the data variation within the indicators in all behavioral sequences. Indicators with larger differences provide more information through which the index can better distinguish behavior sequences. Therefore, indicators with higher contrast intensity should be given more weight. Conflict characteristics indicate the correlation between indicators and are used to reduce the impact of coupling within indicators. Since the trust indicators used are often interrelated, they are of importance in trust evaluation. The steps for objective weight assignment using CRITIC are as follows:

(1) All the metrics in the user behavior sequence I_n are processed in a way that the larger the metric value is, the better it is, and the values of all corresponding *k-th* metrics between users are normalized. I_k^i can be calculated as:

$$I_k^i = (\tilde{I}_k^i - \tilde{I}_k^{\min}) / (\tilde{I}_k^{\max} - \tilde{I}_k^{\min}), \tag{3}$$

where $\tilde{I}_k^i$ is the original value of *i*th user in *k*th index, $\tilde{I}_k^{\max}$ is the maximum value among all users in *k*th index, $\tilde{I}_k^{\min}$ is the minimum value among all users in *k-th* index.

(2) Use the standard deviation to express the contrast of the index. S_k can be calculated as:

$$S_k = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(I_k^i - \overline{I}_k)^2}, \tag{4}$$

where N is the number of the users, I_k^i is the normalized value of *i-th* user in *k-th* index, $\overline{I}_k$ is the average value of user in *k*th index, S_k is the contrast of the *k-th* index.

(3) The correlation $r_{j,k}$ can be calculated as:

$$r_{j,k} = \frac{\sum_{i=1}^{N}(I_j^i - \overline{I}_j)(I_j^i - \overline{I}_j)}{\sqrt{\sum_{i=1}^{N}(I_j^i - \overline{I}_j)^2 \cdot \sum_{i=1}^{N}(I_k^i - \overline{I}_k)^2}}, \tag{5}$$

where $r_{j,k}$ is the correlation degree between different indicators (*j-th* and *k-th*).

(4) Calculate the conflicting degree R_k using the following formula:

$$R_k = \sum_{j=1}^{K}(1 - r_{j,k}), \tag{6}$$

where K is the number of indicator, R_k is the conflicting degree of the *k-th* indicator with other indicators.

(5) Calculate the amount of information C_k using the following formula:

$$C_k = R_k \cdot S_k, \tag{7}$$

(6) Calculate the objective weights for each indicator w_o using the following formula:

$$w_o = C_k / \sum_{j=1}^{K} C_j. \tag{8}$$

3.4 The Calculation of Comprehensive Trust Score

In this paper, the target object of trust computation is a sequence of user's behaviors for accessing cloud resources, which is a collection of trust attributes composed in a single access cycle and can be expressed as follows:

$$I_n = <P_1, P_2, ..., P_n>, \tag{9}$$

where P_i is the trust attribute of the user for the i-th visit, derived from the trust indicator in Table 1, and n is the length of the behavior sequence. Typically, the 1, 2, …, n −1 attributes are the behavioral trust attributes of the user, and the n-th attribute is the name of the resource on the cloud environment. And the category of trust attributes is the same between different users. The trust evaluation score consists of three parts: real-time behavioral trust calculation score, historical behavioral association trust calculation score, and correction score. The score of real-time operation is divided into trust attributes, and the core basis of scoring each trust attribute is composed of the frequency and deviation degree of abnormal behavior within that attribute. The historical trust score, on the other hand, consists of a confidence score based on a correlation analysis algorithm, while the modified score is a fine-tuning of the trust score based on the resource type and importance level. The combined trust value of the user is caculated as:

$$S_{Trust} = \theta_1 \times G_r(x) + \theta_2 \times G_h(x) - \theta_3, \tag{10}$$

where θ_1 and θ_2 are the weights of the real-time trust score and the historical trust score respectively, and θ_3 is a correction parameter based on the penalty function or attributes such as resource type and importance.

Real-time Behavioral Trust Calculation. The user's real-time behavioral trust score is obtained by multiplying the trust scores of each trust attribute with the weights and then accumulating them. The scores of each trust attribute are obtained by the trust calculation function, and we choose the logistic function as the trust calculation function. Consider the process of a new user accessing a cloud resource, the user trust score should start from zero and increase slowly with the accumulation of good visits. The overall curve of the logistic function is S-shaped and is widely used in the real world for modeling nonlinear growth with high accuracy and matching the "slow-fast-slow" decay trend, which is calculated as

$$f_n(x) = \frac{1}{1 + e^{(\alpha x + \beta)}}, \tag{11}$$

where α and β are hyperparameters that can affect the trust score growth rate, x represents the user's access trust attributes, and the value of $f_n(x)$ represents the current trust score of the user. In the trust evaluation process of real-time operation, a corresponding trust calculation function is established for all trust attributes of each user. The dynamic trust score calculation process can be regarded as the process of moving points on the function, and the value of the function is the trust scoring score. When the user's behavior is normal, the evaluation point moves one step to the right and the trust score increases slowly, and when abnormal behavior occurs, the evaluation point moves several steps to the left. The trust attribute score $G_t(x)$ is calculated as

$$G_t(x) = \sum I_n \times f_n(x), \tag{12}$$

where $f_n(x)$ is trust calculation function in Equations (11), I_n is sequence of user's behaviors, each trust attribute in the sequence has a compute function. The integrated trust score for real-time behavior $G_r(x)$ is calculated as

$$G_r(x) = w_s \times w_o \times G_t(x), \tag{13}$$

where w_s is subjective weight in Equations (2), w_o is objective weight in Equations (8).

The logistic function-based trust calculation increases the sensitivity of the model to malicious or abnormal user behavior, which can defend against potential user malicious behavior. For example, a malicious user increases his trust value by normal access behavior over a long period of time. In the event of an attack, when the model detects a user's malicious behavior once, the trust score drops rapidly, and the next attack of that malicious user must continue to accumulate trust value, and the persistent attack is stopped, and the model provides sufficient time for defense auditing.

Historical Behavioral Trust Calculation. The historical behavior trust score is obtained by mining users' historical access behavior and calculating the confidence level through the association analysis method. Usually, users' resource access behaviors show certain patterns, and they always access relatively fixed types of resources with some relatively fixed access methods. The core of historical behavior correlation trust calculation is to calculate the relationship between each trust attribute in the user's access behavior sequence and the accessed resources, where the trust attributes have been given in Table 1 of 3.1, including access time, IP address, MAC address, etc. The historical behavior association trust score and confidence degree are calculated as shown in Equation 14.

$$G_h(x) = Confidence(behavior \rightarrow resource), \tag{14}$$

$$Confidence(behavior \rightarrow resourse) = \frac{P(behavior \bigcup resource)}{P(behavior)}, \tag{15}$$

where $P(behavior)$ is the probability that this sequence of behavior appears in the historical behavior database, $P(resource)$ is the probability of accessing the resource.

Association analysis technique is to mine the macro statistical features of massive data so that valuable information can be found efficiently in massive data. The core of association analysis is to mine frequent item set, and the Apriori algorithm is the classical algorithm in association analysis, which needs to scan the database several times before generating frequent pattern complete sets while generating a large number of candidate frequent sets, but the time and space complexity of the algorithm is large, and I/O is a big bottleneck, which is not applicable in the "micro-segmentation" cloud computing environment discussed in this paper.

FP-Growth is an improved Apriori algorithm. The efficiency of the algorithm is improved by defining an FP tree in which the association information of the item set is compressed to form a set of conditional databases, and the dataset only needs to be scanned twice regardless of the amount of data. At the same time, the use of a tree structure for behavior sequences facilitates subsequent node updates and storage. The user behavior sequences designed in this paper are of equal length and have the same trust attribute categories inside. The traditional FP-Growth algorithm mines all frequent item set, which is still not applicable in this environment, and there may still be a waste of resources in the process of scanning the dataset.

As shown in Algorithm 2, inspired by the FP-Growth algorithm, we improve the confidence calculation method by combining the features of behavior sequences, and the steps to calculate the trust score associated with the user's historical behavior for this visit are as follows:

(1) Use a sliding window to divide and scan the user behavior databases into transaction dataset, calculate the support degree of all trust attributes, and remove the behavior sequence where the trust element with support degree below the threshold is located, and build the item header table according to the first trust attribute (resource name) in the remaining sequence.

(2) Scan the transaction dataset for the second time and create the root node of the FP tree as item 0 and mark it as "null". In the process of traversing the data set, each sequence is inserted from the root of the tree in turn, and the creation or update of the parent node, child list and item header table is completed in the process to complete the construction of the FP tree.

(3) Depend on the conditional pattern base (containing the set of prefix paths that appear with the suffix pattern in the FP tree), and construct the frequent n-item set with a certain item as the suffix.

(4) Match the sequence of this user access with the frequent item set, and count the values of other nodes in the same subtree as the leaf node where the resource is located, and calculate the confidence as trust score.

Sliding Window and Penalty Mechanism. Based on the concept of "Never trust, always verify" in zero trust, every visit of the access subject will be untrusted by default, so the trust evaluation needs to calculate the trust value for each visit. Usually, we can use an authentication mechanism to ensure the trustworthiness of cloud platform users, but the operator behind the same user identity can be anyone who knows the user's authentication information. Therefore, if a user does not show any behavior for a long time, it is difficult to determine whether the user behind his identity is trustworthy when he logs in to the cloud platform to use the resources next time. An authenticated and long-regulated user with high operational privileges can pose a serious threat to cloud resources if that identity is stolen by other malicious users. If the integration of historical trust records relies only on simple averaging, it can lead to trust abuse by malicious users who intentionally accumulate a large number of trusted records.

Algorithm 2. An improved FP algorithm for calculating confidence

Input: user access request r, minimum support threshold *min_sup*, A collection of user history sequences $x(n)$, sliding window size k.
Output: user confidence score

Use sliding window to divide $x(n)$ into transaction dataset;
For each user access request $x(i)$ in transaction dataset **do**
  Calculate support degree of all trust attributes in $x(i)$;
  If support degree of any trust attribute is below *min_sup* **then**
    Remove behavior sequence.
  end If
  Build item header table according to first trust attribute (resource name).
end For
Create root node of FP tree as item 0 and mark it as "null";
For each user access request $x(i)$ in transaction dataset **do**
  Insert sequence $x(i)$ from root of tree;
  Create or update parent node, child list, and item header table.
end For
For each set of prefix paths that appear with suffix pattern in FP tree **do**
  Construct frequent item set with certain item as suffix.
end For
Associative matching of r with frequent item set;
Count the values of other nodes in the same subtree as the leaf node;
Calculate of confidence score using **Equation** (15).
Return user confidence score

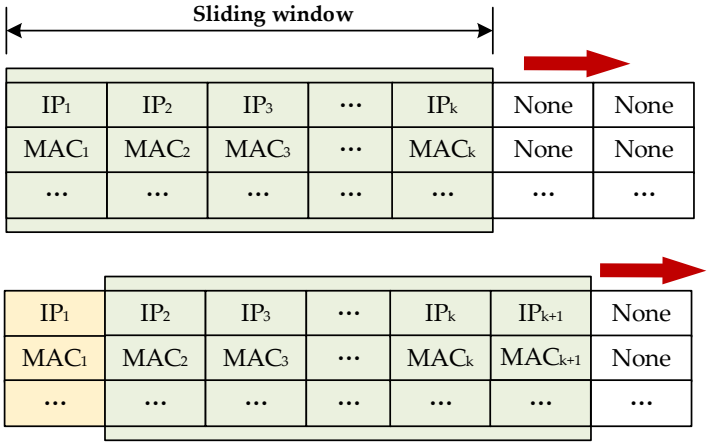


Fig. 3. The sliding window model is used to obtain the user history behavior sequence in trust evaluation

In this paper, we introduce a sliding window to make the trust evaluation process more time-sensitive and reliable, and k is the maximum length of the sliding window. As shown in Fig. 3, each user has a table of access behavior records for trust evaluation, and when the number of records in the table exceeds the maximum window, the old data beyond that window will lose its reference value as expired records and will be deleted from the data table. Only the data located in the sliding window is used to calculate the trust value, and the window is moved by 1 unit per interaction.

Based on the idea that a user who has not logged into the cloud platform for a long time to use cloud resources will be considered as a new user when requesting cloud resources again, the user trust value is modified based on the time interval between the user's two accesses to cloud services, and the type and importance of the requested resources as shown in equation 15,

$$\theta_3 = \gamma \times e^{\delta t}, \tag{16}$$

where γ is a correction factor based on the resource type and importance, taking values in the range of [0, 1], γ is the time difference between two user logins, and δ is the hyperparameter. Since the trust score cannot be negative, when using θ_3 for score correction, need to ensure that it is less than the current trust value. Here we need to emphasize that in some articles, the penalty mechanism is corrected for the weight of the trust score. That is: $S_{Trust} = \theta_3 \times S_{Trust}$, but this penalty mechanism is heavily dependent on the time interval between two user logins, and if θ_3 is used as the weight parameter for the correction. The trust score may show an exponential trend, which will make the weight of other trust attributes lower, so θ_3 in the proposed method only makes subtractive correction to the trust score.

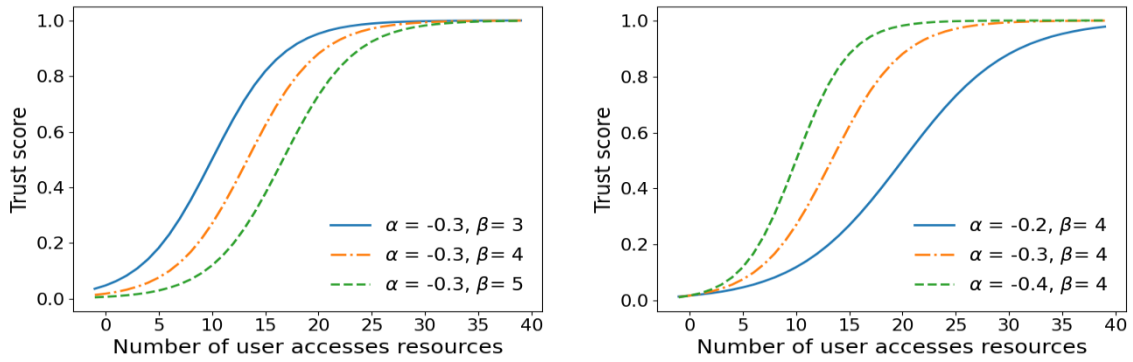
4 Experiments and Analysis

In the previous section, we designed a user behavior-based trust evaluation model for the "micro-segmentation" cloud computing environment, in which a user's access is considered as a set of behavior sequences, and each module of the model scores the user's access behavior sequences and aggregates them into a comprehensive trust value. In this section, the details of the parameters of the above model and the experimental results are discussed. All experiments are simulated in Python, and user access behavior data was generated by simulation. The experimental setup consists of an Intel(R) Core(TM) CPU model with Corei7-10780H, 2.20GHz, DDR4 memory, and 32G memory capacity.

4.1 Sensitivity Analysis for Real-time Behavioral Trust Calculation

The scoring function is used for real-time access sequence trust calculation of users, and each fine-grained divided trust attribute has a trust calculation function, so its design and the selection of parameters directly affect the trust evaluation. Since the growth of the trust score is a process from 0 to 1, we selected the logistic function as the trust calculation function, which is a function that conforms to the law of nature with an S-shaped curve, and it is reasonable to use this function for user's trust score in the cloud environment.

Fig. 4 analyzes the influence of hyperparameters α and β on the trust calculation function. By fixing the value of the parameter β to 4, different α reflects the trust growth rate of users in the middle stage, and the larger the absolute value of α the faster the growth. After fixing α, β reflects the movement of the function in the horizontal direction, and the appropriate selection of β ensures the process of accumulating the trust of users in the early stage. In the early stage of user interaction with cloud resources, there is a process of understanding the user in the trusted node, and the too-fast growth of the user trust score is not appropriate. In the middle of the interaction, since the user has accumulated a certain amount of trust, normal access behavior allows the user's trust score to grow rapidly. At the end of the visit, since there is a certain upper limit of trust value, the growth of the trust score can gradually slow down and eventually converge to the threshold value. Thus, the "slow, fast, slow" form of the user's trust score can be highly compatible with the logistic function.

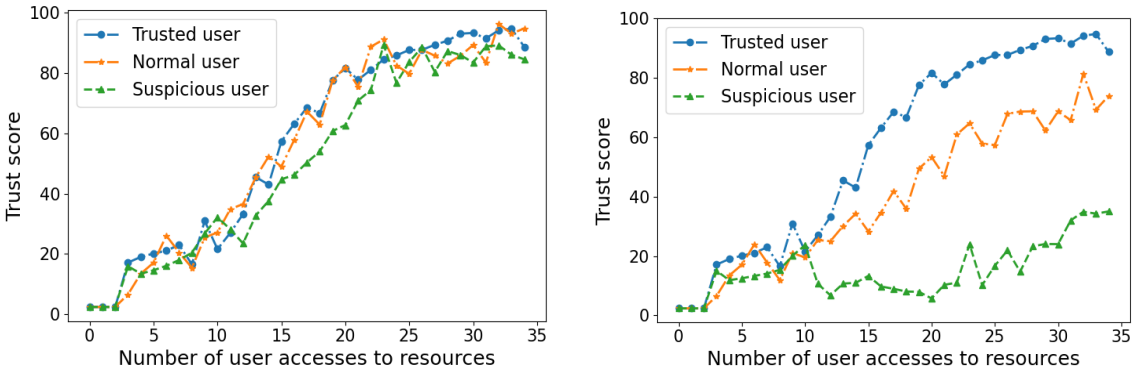


(a) β influences the initial growth of trust scores (The larger the value of β, the longer the path that users need to take to increase their trust score.)

(b) α influences the rate of growth of trust scores (The larger the value of α, the faster the user trust score grows.)

Fig. 4. Sensitivity of the of hyperparameters α and β to trust calculation function

Since the process of calculating real-time trust score can be regarded as a point sliding on the trust calculation function, the function shifts to the right when the user's behavior meets the trust expectation and vice versa to the left. Fig. 4 shows the change of trust score for 3 users accessing resources under different constraints, where the values of θ_1, θ_2, θ_3 are 0.5, 0.5 and 0 respectively. When the model believes that the user has misbehaved, the function will move multiple frames to the left, resulting in a significant decrease in the user's trust score, and this design will make the cumulative trust attack by malicious users more costly.



(a) More lenient thresholds for trust attributes do not differentiate between different user access behaviors

(b) Stricter thresholds for trust attributes clearly differentiate between different user access behaviors

Fig. 5. Influence of trust calculation function on user trust evaluation

Fig. 5 describes the influence of trust calculation function on user trust evaluation. A comparison of Fig. 5(a) and Fig. 5(b) shows that the higher constraint limits the trust score accumulation of potentially suspicious users. This constraint is a constraint on the trust attributes, set by the cloud resource manager, which can be reflected in the list of commonly used IPs and MAC addresses, the number of password entry errors, and so on. In the case of low constraint, the trust score sequences of trusted and suspicious users are very similar, and the gap is within 15 points. In the case of high constraint, the gap between trust score sequences of trusted and suspicious users is significantly widened, exceeding 50 points.

4.2 The Influence of Historical Behavior Trust Calculation on User Trust Score

Historical behavioral trust computation aims to mine the intrinsic association of users' behavior in accessing cloud resources. Since we design user behavior sequences of equal length and the same trust attribute categories inside, traditional Apriori and FP-Growth algorithms will mine all frequent item set and there is a waste of time efficiency. We have made some improvements to some parts of the FP-Growth algorithm and validated them. Based on the scalability of the model we designed, the manager of the cloud resource can extend the model with any number of trust attribute category attributes and trust attribute features according to the actual situation. The core of the association mining algorithm is the construction of frequent item set. The sliding window controls the amount of data read in by the algorithm, and Table 2 shows the time overhead of each algorithm at different amounts of data. We fixed the feature dimension of each trust attribute to be 20 and the maximum value of the sliding window s to be 1000, which means that the number of sequences of user access requests (data volume) for constructing the frequent item set is 1000. Each experiment was repeated 10,000 times to obtain the mean and variance, and it can be seen that FP-Growth has a significant advantage over the Apriori algorithm in terms of time overhead, with a difference of more than 200%. Meanwhile, our improved method reduces more than 10% in time overhead compared to FP-Growth.

Table 2. Comparison of the time overhead of association analysis algorithms for different data volume scenarios

Association analysis algorithm	Indicator	The relative complexity of trust data volume scenarios				
		20%	40%	60%	80%	100%
Apriori	mean ± std (*ms*)	5.47 ± 0.20	15.31 ± 0.61	29.20 ± 1.07	49.29 ± 2.52	70.21 ± 2.91
FP-Growth	mean ± std (*ms*)	0.98 ± 0.17	1.91 ± 0.26	2.69 ± 0.23	3.53 ± 0.27	4.42 ± 0.65
Ours	mean ± std (*ms*)	0.79 ± 0.15	1.51 ± 0.19	2.22 ± 0.23	2.92 ± 0.27	3.58 ± 0.35

For cloud resource managers, a more fine-grained trust attribute feature dimension is likely to be more practical. Table 3 shows the time overhead comparison of each algorithm for different trust attribute complexity, where the sliding window k has a value of 500 and the maximum number of features is 100. Fig. 6 shows more clearly the comparison between our approach and the FP-Growth algorithm. grows, the time overhead of our algorithm gradually decreases, and in the case of 100 features, the overhead of our time decreases by 18%.

Table 3. Comparison of the time overhead of association analysis algorithms for different trust attribute complexity cases

Association analysis algorithm	Indicator	The relative complexity of trust data volume scenarios				
		20%	40%	60%	80%	100%
Apriori	mean ± std (*s*)	0.97 ± 0.06	1.22 ± 0.10	1.37 ± 0.13	1.60 ± 0.14	1.97 ± 0.21
FP-Growth	mean ± std (*s*)	0.05 ± 0.00	0.06 ± 0.01	0.07 ± 0.01	0.07 ± 0.01	0.08 ± 0.01
Ours	mean ± std (*s*)	0.05 ± 0.01	0.05 ± 0.01	0.06 ± 0.02	0.06 ± 0.02	0.07 ± 0.01

Since users' resource access behavior presents a certain regularity, an innovative point of the model designed in this paper is to introduce association analysis to mine users' intrinsic behavioral association features and realize trust calculation of users' historical behavior. The trust score calculation in this part is independent of real-time behavior, and relatively regular access patterns in a certain time range indirectly prove the user's identity, forming a complement to the real-time trust score.

We emphasize that there is a fundamental difference between real-time behavior evaluation and historical behavior evaluation. The real-time behavior score is scored based on trust attributes, which better reflects the trustworthiness of the operation. Historical behavior is an intrinsic association of the user's access behavior, and the confidence score obtained by association analysis reflects the degree of association of that behavior sequence with the user's real identity from the side. In Fig. 7, we show the role of the historical behavior trust parameter when the user's behavior sequence does not change. As the weight of θ_2 becomes higher, the user's trust score fluctuates more, and the trust scores of trusted and suspicious users are further distinguished, reaching a gap of about 20 points.

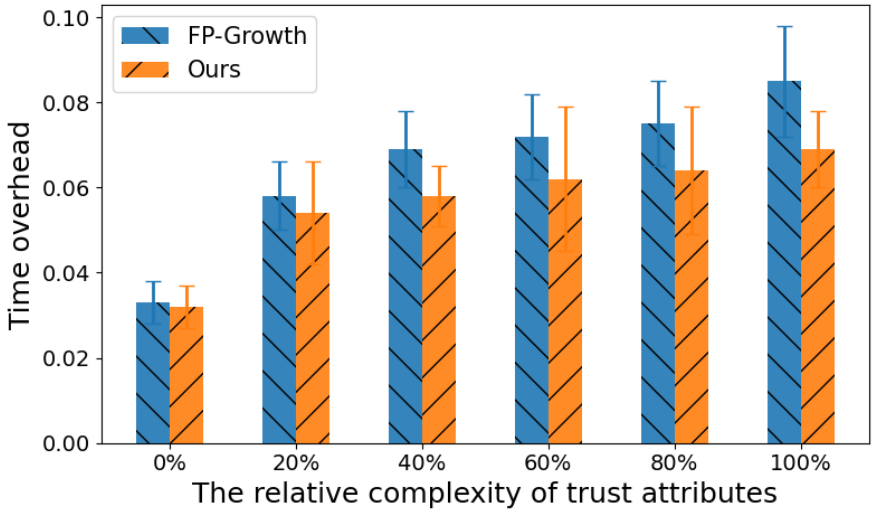
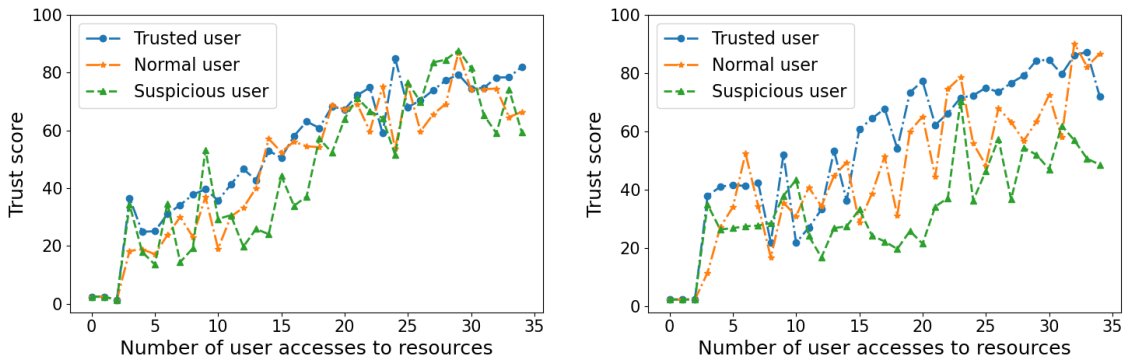


Fig. 6. Comparison of the time overhead of association analysis algorithms for different trust attribute complexity cases



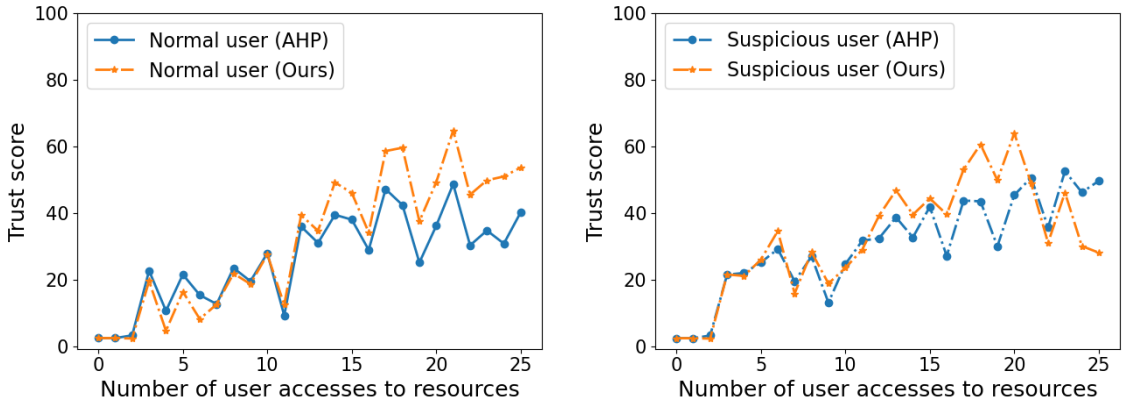
(a) Higher real-time trust score weight with $\theta_1 = 0.8$, $\theta_2 = 0.2$, $\theta_3=0$

(b) Higher historical trust score weight with $\theta_1 = 0.5$, $\theta_2 = 0.5$, $\theta_3=0$

Fig. 7. The influence of historical behavior trust calculation on user trust score, users with abnormal access behavior are distinguished

4.3 The Influence of Dynamic Weight Allocation on User Trust Score

Finally, we analyze the influence of different weight allocation methods. In Fig. 8, due to the dynamic allocation of trust attribute weight, the trust score of normal users gradually increases, while that of suspicious users gradually decreases after weight adjustment. Our example illustrates the advantage of the CRITIC method in weight distribu-tion. Although the weight of AHP is obtained according to objective comparison of trust attributes, its essence is still expert rating, which is a subjective weight allocation method. However, the CRITIC method is based on two concepts of comparison intensity and conflict characteristics of evaluation indicators. By adding different indicators and reducing internal coupling, the comprehensive weight allocation method is based on the original characteristics of data, which is more objective.



(a) Dynamic weight allocation improves the trust scores of normal users

(b) Dynamic weight allocation reduces the trust score of suspicious users

Fig. 8. Change of users' trust scores under different weight allocation methods

Fig. 9 shows the change in a single user's trust score after using the CRITIC method. In the interval of [0, 25] and [25, 40], it can be seen that the weight of trust attribute changes significantly in the middle period after the accumulation in the early period, thus affecting the change of score. The introduction of CRITIC method makes the user trust evaluation process dynamic.

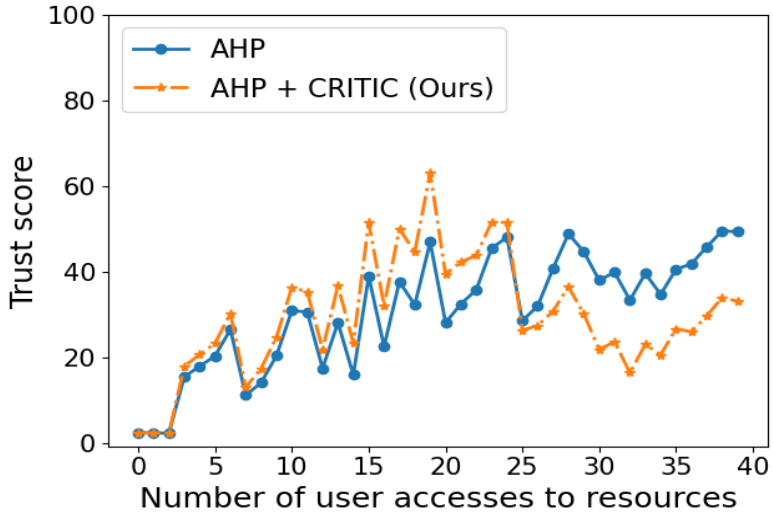


Fig. 9. Trust score of a single user under different weight allocation methods

5 Conclusion and Future Work

To address the special characteristics of increased user access requests and diverse and complex access behaviors in the "micro-segmentation" cloud computing environment, this paper proposes a dynamic and fine-grained end-user trust evaluation model, which adaptively extracts the recent access behavior characteristics of users based on sliding windows, and performs dynamic and fine-grained scoring for each user access request, In this paper, we propose a dynamic and fine-grained end-user trust evaluation model, which adaptively extracts the

characteristics of users' recent access behavior based on a sliding window, and performs a dynamic and fine-grained scoring for each user access request to achieve a continuous user trust record and provide secure access to guest resources in the "micro-segmentation" cloud environment. In this model, the trust evaluation score consists of three parts: real-time behavior trust score, historical behavior associated trust calculation score, and correction score. The user's real-time behavioral trust score is obtained by multiplying the trust scores and weights of each trust attribute. The historical behavior trust calculation score is obtained by mining the user's historical access behavior and calculating the confidence level through the method of correlation analysis. The correction score is a subtractive correction to the user trust value based on the time interval between the user's two accesses to the cloud service, and the type and importance of the requested resources. In the trust evaluation process, the classical sliding window and penalty mechanism are introduced to ensure the model is more credible.

To validate the proposed method, we conducted simulation experiments in a Python environment and analyzed the parameters and sensitivity of the proposed model in detail. The results show that the proposed dynamic and fine-grained model has more than 10% improvement over the traditional method in the time overhead method in the environment set in the paper. The organic combination of the trust evaluation model with CRITIC assignment and correlation analysis can effectively evaluate user behavior. At the same time, the model we designed has good scalability and can be used by users to adjust the parameters according to their actual situation. Future research can conduct experiments in more realistic scenarios to continuously optimize our model. Since the model proposed in this paper is a lightweight trust evaluation model based on traditional methods, it is also a future research direction to compare the accuracy and overhead with trust evaluation methods using machine learning.

Acknowledgement

This work is supported by the project "Key technologies research on security protection for distribution cloud master station and intelligent terminal" (5400-202116144A-0-0-00) of the State Grid Corporation of China.

References

[1] M.N. Birje, P.S. Challagidad, R.H. Goudar, M.T. Tapale, Cloud computing review: concepts, technology, challenges and security, International Journal of Cloud Computing 6(1)(2017) 32-57.
[2] A. Rashid, A. Chaturvedi, Cloud computing characteristics and services: a brief review, International Journal of Computer Sciences and Engineering 7(2)(2019) 421-426.
[3] K. Okereafor, P. Manny, Understanding cybersecurity challenges of telecommuting and video conferencing applications in the COVID-19 pandemic, International Journal in IT& Engineering (IJITE) 8(6)(2020) 13-23.
[4] A. Singh, M. Korupolu, D. Mohapatra, Server-storage virtualization: integration and load balancing in data centers, in: Proc. SC'08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing, 2008.
[5] W.R. Claycomb, A. Nicoll, Insider threats to cloud computing: Directions for new research challenges, in: Proc. 2012 IEEE 36th annual computer software and applications conference, 2012.
[6] C.H.S.S. Prasad, B.P. Yadav, S. Mohmmad, M. Gopal, K. Mahender, Study of threats associated with cloud infrastructure systems, IOP Conference Series: Materials Science and Engineering 981(2020) 022055.
[7] S. Rose, O. Borchert, S. Mitchell, S. Connelly, Zero trust architecture, NIST special publication 800-207, https://doi.org/10.6028/NIST.SP.800-207.
[8] C. Buck, C. Olenberger, A. Schweizer, F. Völter, T. Eymann, Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero-trust, Computers & Security 110(2021) 102436.
[9] D. Klein, Micro-segmentation: securing complex cloud environments, Network Security 2019(3)(2019) 6-10.
[10] N. Sheikh, M. Pawar, V. Lawrence, Zero trust using network micro segmentation, in: Proc. IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2021.
[11] Snyder, Formal models of capability-based protection systems, IEEE Transactions on Computers C-30(3)(1981) 172-181.
[12] R.S. Sandhu, Role-based access control, Advances in computers 46(1998) 237-286.
[13] D. Bernstein, Containers and cloud: From lxc to docker to kubernetes, IEEE cloud computing 1(3)(2014) 81-84.
[14] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, l. Pratt, A. Warfield, Xen and the art of virtualization, ACM SIGOPS operating systems review 37(5)(2003) 164-177.
[15] I. Habib, Virtualization with KVM, Linux Journal 2008(166)(2008) 8.
[16] R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni, F.C.M. Martins, A.V. Anderson, S.M. Bennett, A. Kagi, F.H. Leung, L. Smith, Intel virtualization technology, Computer 38(5)(2005) 48-56.

[17] K. Adams, O. Agesen, A comparison of software and hardware techniques for x86 virtualization, ACM Sigplan Notices 41(11)(2006) 2-13.
[18] G. Vallee, T. Naughton, C. Engelmann, H. Ong, S.L. Scott, System-level virtualization for high performance computing, in: Proc. 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing (PDP 2008), 2008.
[19] T.N. Mujawar, L.B. Bhajantri, Behavior and feedback based trust computation in cloud environment, Journal of King Saud University-Computer and Information Sciences 34(8)(2022) 4956-4967.
[20] A. Ladekar, P. Pawar, D. Raikar, J. Chaudhari, Web log based analysis of user's browsing behavior, International Journal of Computer Applications 115(11)(2015) 5-8.
[21] H. Das, R. Barik, H. Dubey, D. Roy (Eds.), Cloud Computing for Geospatial Big Data Analytics: Intelligent Edge, Fog and Mist Computing, Springer, Cham, 2019 (pp. 55-79).
[22] P. Zhang, M. Zhou, Y. Kong, A double-blind anonymous evaluation-based trust model in cloud computing environments, IEEE Transactions on Systems, Man, and Cybernetics: Systems 51(3)(2021) 1805-1816.
[23] K. Riad, T. Huang, L. Ke, A dynamic and hierarchical access control for IoT in multi-authority cloud storage, Journal of Network and Computer Applications 160(2020) 102633.
[24] R. Zheng, J. Chen, M. Zhang, J. Zhu, Q. Wu, User abnormal behavior analysis based on neural network clustering, The Journal of China Universities of Posts and Telecommunications 23(3)(2016) 29-36, 44.
[25] S. Wang, Y. Zhang, A credit-based dynamical evaluation method for the smart configuration of manufacturing services under Industrial Internet of Things, Journal of Intelligent Manufacturing 32(4)(2021) 1091-1115.
[26] B. Shayesteh, V. Hakami, A. Akbari, A trust management scheme for IoT-enabled environmental health/accessibility monitoring services, International Journal of Information Security 19(1)(2020) 93-110.
[27] C. Wang, IoT anomaly detection method in intelligent manufacturing industry based on trusted evaluation, The International Journal of Advanced Manufacturing Technology 107(3-4)(2020) 993-1005.
[28] H. Wang, D. Yang, Q. Yu, Y. Tao, Integrating modified cuckoo algorithm and creditability evaluation for QoS-aware service composition, Knowledge-Based Systems 140(2018) 64-81.
[29] Y. Guo, X.J. Yang, Modeling and predicting trust dynamics in human–robot teaming: A Bayesian inference approach, International Journal of Social Robotics 13(8)(2021) 1899-1909.
[30] K. Yang, L. Zhao, X. Yu, K. Cheng, J. Ma, Research on Dynamic Access Control Mechanism Based on Short-term Token and User Trust, in: Proc. 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS), 2022.
[31] N.R. Paul, D.P. Raj, Enhanced Trust Based Access Control for Multi-Cloud Environment, Computers, Materials & Continua 69(3)(2021) 3079-3093.
[32] S.B. Hosseini, A. Shojaee, N. Agheli, A new method for evaluating cloud computing user behavior trust, in: Proc. 2015 7th Conference on Information and Knowledge Technology (IKT), 2015.
[33] R. Yang, X. Yu, Research on Way of Evaluating Cloud End User Behavior's Credibility Based on the Methodology of Multilevel Fuzzy Comprehensive Evaluation, in: Proc. of the 6th International Conference on Software and Computer Applications, 2017.
[34] L. Tian, J. Li, Z. Wu, Trust evaluation of web user behavior with weight optimal balance, Journal of Beijing University of Posts and Telecommunications 39(6)(2016) 99-103, 30.
[35] C. Borgelt, An Implementation of the FP-growth Algorithm, in: Proc. of the 1st international workshop on open source data mining: frequent pattern mining implementations, 2005.
[36] O.S. Vaidya, S. Kumar, Analytic hierarchy process: An overview of applications, European Journal of operational research 169(1)(2006) 1-29.
[37] D. Diakoulaki, G. Mavrotas, L. Papayannakis, Determining objective weights in multiple criteria problems: The critic method, Computers & Operations Research 22(7)(1995) 763-770.
[38] R. Yang, X. Yu, Research on building the credibility evaluation's indicator system of cloud end user's behavior, in: Proc. 2017 ieee 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids), 2017.