The purpose of this paper is to build a novel trust evaluation mechanism model for edge devices of the Internet of Things (IoT). First, we use the SM9 authentication mechanism to authenticate the identity of the terminal device. Authentication includes both single-domain and cross-domain authentication of devices, reducing the time overhead of the authentication process. Then, we use the satisfaction function and the time degradation factor to improve the direct trust in the behavioural trust of Bayesian equations. Next, the weight of indirect trust of each device is calculated to improve the gray association method, and the accuracy of indirect trust is improved. Finally, the trust value of the edge device is obtained by combining the authentication result and the behaviour trust. Simulation results verify that the trust evaluation model proposed in this paper has certain advantages for the reliability, accuracy, and efficiency of trust evaluation.

1. Introduction

With the development of the Internet of Things (IoT), the traditional cloud-based authentication framework has gradually evolved to an authentication architecture based on edge computing [1]. The traditional PKI system consumes the resources of the equipment excessively [2]. However, due to the resource-constrained nature of edge devices, traditional authentication architectures cannot be applied to authentication of devices in edge computing environments. In addition, due to the strong mobility of devices in the edge computing environment, how to achieve efficient authentication when switching between different edge devices is a great challenge [3]. Therefore, low-consumption and efficient identity authentication technology is the premise of ensuring edge computing security [4]. It is necessary to design an authentication scheme for edge computing environments.

Recently, many researchers propose a dynamic device trust evaluation model for device trust in PTM’s research project [5]. The model uses a probability-weighted approach to assess the user’s trust. In the PET project [6], the scholar proposed an evaluation model based on the trust coupling of risk and feedback to calculate the trustworthiness of nodes in a P2P network during file sharing. Guo et al. [7] make a detailed analysis of how to establish a more secure and reliable execution environment in the edge computing environment, effectively assessed the trust of device nodes in the environment, and then analyzed the data security aspects in the environment. Chi et al. [8] make a detailed description of the edge computing environment from the aspects of data security and privacy protection, introduced the relationship between the trust domain and the device entity in the environment, studied the identity problems of the device entities in different trust domains, and proposed the security threats at each layer. Din et al. [9] propose a distributed federated trust algorithm based on merge splitting rules for small cell networks (SBSs). The definitions in this article describe the trust relationships between SBSs. This relationship is generated through a network of social trust, which is calculated by propagating trust by the shortest path that connects them. Haghighi et al. [10] propose a multilayer adaptive edge
computing data structure based on comprehensive trust for privacy trust and data security issues in the edge computing environment and how to improve the quality of user experience according to the needs and adjusted the allocation of tasks. Finally, they test the effectiveness of the model. However, they only use subjective evaluations between nodes and ignore objective factors. Therefore, the model has certain limitations. Yen et al. [11] propose a distributed reputation management approach that employs secure and efficient vehicle edge computing and networking. In order to ensure network security protection, they focused on reputation management, using weighted subjective logic to update the reputation data in the system. Bradbury et al. [12] propose a reliable and lightweight trust mechanism for IoT edge devices based on multisource feedback information fusion. The trust mechanism mainly evaluates direct trust with a risk model and proposes a method based on information entropy to recommend a trust computer system. This trust computer system improves the computational fandom and reliability of trust. However, the treatment of malicious nodes is not reflected in the article, resulting in low accuracy of trust assessment.

Regardless of the computing environment, the trust evaluation mechanism is constantly proposed by scholars at the forefront of academics [13]. However, there are certain problems that cannot be ignored in these model algorithms. They focus only on the outcome of the interaction and ignore other factors that influence the outcome and whether the identity is trustworthy and do not consider the time factor. These factors prevent models from effectively in a timely manner resisting malicious behaviours such as malicious spoofing [14]. In addition, these studies weighed too much weight over nonimportant recommended devices and did not conform to conventional logic [15]. Therefore, the trust evaluation mechanism is still a very important topic for scholars to study security.

In the paper, we propose the novel trust evaluation mechanism model for edge devices of the IoT. First of all, we use the SM9 authentication mechanism to authenticate the identity of the terminal device. Authentication includes both single-domain and cross-domain authentication of devices, reducing the time overhead of the authentication process. Then, we use the satisfaction function and the time degradation factor to improve the direct trust in the behavioural trust of Bayesian equations. After that, the weight of indirect trust of each device is calculated to improve the gray association method, and the accuracy of indirect trust improved. Finally, the trust value of the edge device is obtained by combining the authentication result and the behaviour trust. Simulation results verify that the trust evaluation model proposed in this paper has certain advantages for the reliability, accuracy, and efficiency of trust evaluation.

2. Authentication Model Design Based on SM9

2.1. Signature and Check Algorithm-Based SM9

In this paper, the SM9 cryptographic algorithm [16] is used to design the authentication algorithm for edge devices. The algorithm uses bilinear pair implementation and discards the PKI complex verification certificate revocation and management related operations, which is more suitable for the resource-limited edge devices. Based on the SM9 cryptographic algorithm, the signature and check algorithm is designed. The signature algorithm is shown as follows.

In addition, the check-in algorithm is designed, as shown in Algorithm 2.

2.2. Authentication Architecture

In an edge-IoT environment, the end device only needs to authenticate to the edge agent and no longer needs to send authentication requests to the cloud. As shown in Figure 1, there are three entities in the proposed authentication architecture, including cloud center, edge agent, and end device. In the traditional cloud system, cloud center nodes have a high degree of identity confidence in establishing reliable authentication technology. In the edge-IoT environment, edge agents are low-level fixed and have less mobility, which also have higher identity confidence. Because of the dynamic and mobility of terminal devices, the authentication credibility of the terminal equipment is difficult to determine. In addition, the terminal device contains the customer’s privacy data. In order to protect the user’s privacy for the terminal device, an interaction authentication mechanism is necessary. Therefore, the terminal device is mainly verified in the authentication framework of edge-IoT.

Different from the centralization of cloud computing, edge computing is a new distributed computing model. The interaction of edge terminal devices is divided into two different ways. Edge endpoints may belong to the control of the same edge layer node, that is, the edge compute node of the same PKG. In addition, if the end device has different edge layer nodes, its PKG master key is different. This will not enable mutual authentication of the terminal equipment, nor will it provide continuity and stability for the interaction of the terminal equipment. Therefore, we propose a single-domain and cross-domain edge device authentication mechanism, as shown in Figures 2 and 3.

2.3. Single-Domain Authentication Mechanism for Edge Device

In an edge computing environment, you need to authenticate yourself to the edge computing layer, whether you need edge computing layer nodes to provide services [17]. The edge compute layer node also records the authentication information for the device. The main steps in the authentication process for endpoint nodes are as follows:

Step 1. When the device $d_i$ joins the network, authentication is required at the edge compute layer. This process requires that the authentication request be sent to the edge compute layer node $s_j$ first, which in turn obtains the corresponding key and applies the key communication to the session. The request sent by the device contains the identity of the device and the edge layer node. The corresponding digital signature is calculated by using the private key of the device in the SM9 signature algorithm to sign $(h, s)$.

\[
d_i \rightarrow s_j : \text{AccessReq} \parallel d_i \parallel s_j \parallel h \parallel S.
\]
Step 1: Calculate the elements \( g = e(P_1, P_{pub}) \) in the group \( G_T \).
Step 2: Produce random numbers \( r \in [1, N - 1] \).
Step 3: Calculate the elements \( \omega \) in \( G_T \).
Step 4: Calculate \( h = H_f(M|\omega, N) \).
Step 5: Calculate \( l = (r - h) \mod N \), if \( l = 0 \), go to Step 2.
Step 6: Calculate the element \( S = [l]d_{pub} \) in \( G_T \).
Step 7: Get the digital signature \((h, S)\) of \( M \).

**Algorithm 1: Signature algorithm-based SM9.**

Step 1: Verify that the signature is valid and does not end directly.
Step 2: Calculate the elements \( g = e(P_1, P_{pub}) \) in the group \( G_T \).
Step 3: Calculate the elements \( t = g^h \) in \( G_T \).
Step 4: Calculate \( h_t = H_f(ID_a||hid, N) \).
Step 5: Calculate the elements \( P = [h_t]P_2 + P_{pub} \) in \( G_T \).
Step 6: Calculate \( G_T \) the elements \( u = e(S', P) \).

**Algorithm 2: Check algorithm-based SM9.**

Here, \( \text{AccessReq} \) denotes the authentication request for devices. \( N_j \) is a random number.

Step 2. The Edge compute layer node uses the SM9 signature algorithm to verify the signature of the request line after receiving an authentication request from the end device. Once verified, the edge compute layer node saves the identity information of the end device to the authentication list and feeds the encrypted authentication information back to the end device node. First, the value of the element \( G_t \) in the group is calculated according to equation (1). The Cipher value in the group is calculated based on the generated random number \( r \). Cipher is a redaction, where \( r \in [1, N - 1] \). Based on equation (2), the encapsulated key \( \text{Key} \) is calculated using KDF. The value of Key is the shared key for the edge layer and the end device.

\[
Q_D = [H_1(d_i||hid, N)]P_1 + P_{pub-e}
\]

(2)

\[
\text{Key} = \text{KDF}(\text{Cipher}||P_{pub-e}, P_2')||D, \text{klen}, \text{h}||S.
\]

(3)

\[
d_i \rightarrow s_i : \text{AccessRsp}||s_i || \text{Cipher}||h||S.
\]

(4)

Here, \( \text{klen} \) is the length of key. \( \text{AccessRsp} \) is a response identity of requests.

Step 3. After receiving the response package for the edge layer node, the edge end device parses the received Cipher to obtain the appropriate key \( \text{Key} \), which first determines whether Cipher belongs to the element in and does not belong to the \( G_t \). According to equation (4), calculate the \( \omega' \) in \( G_t \) and use SM9 to convert the data to a bit stream; then, calculate the key \( \text{Key} \) for the response.

Here, \( \text{AccessAck} \) is the response. \( \text{d}_{e_{al}} \) is the terminal private key.

In summary, authentication fails if the resulting key is 0. Otherwise, device authentication is successful, and then, save the device’s identity to the edge compute layer node.

2.4. Authentication of Edge Device Terminals across Domain.

Edge computing environment is a typical distributed computing model. The PKG of two different edge layer nodes may be different. Therefore, authentication keys will not be the same. In order to achieve cross-domain authentication of end equipment, the implementation process is as follows:

Step 1. Cross-domain authentication is conditional on the end device having completed authentication in the domain to which it belongs. When the device moves to another domain, the edge compute node sends a request...
for cross-domain authentication. The identity is encrypted according to the equations (5) and (6), and a new domain edge computing node is calculated as follows.

\[
C = E_{\text{key-old}}(d)|s_1|s_2), \quad (5)
\]

\[
d_1 \longrightarrow s_2 : \text{Reauth}|d_1||s_1||s_2||C||N_1. \quad (6)
\]

Here, Reauth is cross-domain requests. \(N_1\) is a random number.

Step 2. If a new edge compute node receives a cross-domain authentication request for a device, verify that there is a trust relationship with the original domain based on the identity passed over. If there is a trust relationship, step 3 is executed directly; otherwise, the new domain sends a request to the original edge compute layer node for authentication information. Once the request is accepted from \(s_2\), the digital certificate is used to authenticate the pair. The public key obtained is validated when feedback is received.

Step 3. Certification information for cross-domain end devices is derived from \(s_1\) and \(s_2\). Based on \(s_2\) and \(r_{s_2}\), the calculation \(G \cdot r_{s_2}\) is made and the information is sent to \(s_1\).

\[
s_2 \longrightarrow s_1 : \text{KeyAgree} || d_1||s_2||G \cdot r_{s_2}||C||N_2)_{pk_{s_1}}. \quad (7)
\]

Here, KeyAgree represents a negotiated request. \(N_2\) is a random number.

Step 4. Once a request is received from \(s_1\), the identity of the device is verified and the cross-domain information is received. Based on \(s_2\) and \(r_{s_2}\), the calculation \(G \cdot r_{s_2} \cdot r_{s_1}\) is made. Once the final terminal authentication information is derived, the information is fed back to IDg2, as shown in

\[
s_1 \longrightarrow s_2 : \text{KeyAck} || d_1||s_1||G \cdot r_{s_1} \cdot r_{s_2}||E_{\text{key-old}}(\text{Key})||N_3)_{pk_{s_2}}. \quad (8)
\]

To sum up, we propose the authentication of interdomain terminal devices of different edge computing layer nodes. No complex calculations and communications are required in this process, and digital authentication is simple to use. Key negotiation completes the authentication of the terminal device and the edge computing layer, which effectively reduces the calculation and overhead of the verification process.

### 3. Trust Assessment of Edge Device’s Dynamic Behaviour

#### 3.1. Direct Trust Assessment Based on Improved Bayesian Methods

Most of the current research on direct trust uses Bayesian trust-based assessment methods. The trust evaluation method is mainly calculated by evaluating the historical data of the interaction between the device node and the evaluated device node. \(r\) and \(s\) record the number of successful and failed interactions between the two devices. Since the beta distribution function better fits the reputation distribution, the statistical expectation for direct trust from \(d_i\) to \(d_j\) is as follows.

\[
C_{d_i,d_j} = E(\text{Beta}(r + 1, s + 1)) = \frac{r + 1}{r + s + 2}. \quad (9)
\]

Although equation (9) is capable of predicting the trust value of future device nodes from historical interaction data, trust is dynamic. Equation (9) does not reflect the decay of time, making it difficult to ensure the timeliness of the trust evaluation of device nodes. In addition, equation (9) ignores the interaction failure caused by abnormalities in the process of interaction of the device node, so that the results of the direct trust evaluation cannot accurately characterize the behavior of the device node.

The satisfaction function records the satisfaction of device transactions during the evaluation process. In an edge computing environment, edge device nodes affect the interaction results according to the changes in the influencing factors at that time during the interaction process. Here, influencing factors include computing power, data storage capacity, energy supply capacity, etc. Some device nodes are more concerned with transaction speed, and the other part is more concerned with the integrity of the interaction data. Therefore, calculating direct trust only from historical interaction records may result in a large false detection rate. To solve this problem, the concept of satisfaction is introduced to correct the beta density distribution function. Satisfaction is defined as shown in

\[
q(t) = \sum_{k=1}^{n} e_k \cdot \omega_k. \quad (10)
\]
Here, $c_k \in [0, 1]$ indicates the evaluation of the influencing factors of the device node on the interaction process. $\omega_k$ is the evaluation weight and $\omega_k \in [0, 1]$.

The degree of trust between nodes is dynamically variable. As an important criterion for cooperation between nodes, trust needs to select nodes to cooperate according to the prediction of trust. In addition, the trust value between the two nodes is to select the trust value updated by the node at a more recent time as the evaluation index. Thus, the time degradation factor reflects the dynamic nature of the trust value over time, as defined in

$$k_z = \frac{1}{t - t_z + 1}, k \in (0, 1).$$ \hspace{1cm} (11)

Here, $t$ is the current time. The time degradation factor calculates the direct trust between two device nodes based on the time that occurs from the interaction between the nodes. As a result, the same factor values are produced in time-close interaction records. To facilitate the use of the beta density function, this article introduces the $p$ variable. If the interaction between the two nodes is successful, $\rho = 1$; otherwise, $\rho = 0$. After the time degradation factor is introduced, the variables in the beta density function change accordingly, and its latest construction method is shown in

$$r_{new}^{(t)} = \sum_{z=1}^{n} k_z p_z,$$

$$s_{new}^{(t)} = \sum_{z=1}^{n} k_z (1 - p_z).$$ \hspace{1cm} (12)

According to the historical interaction, this paper introduces an incentive mechanism to calculate the direct trust value, where thresholds are the average of the number of interaction failures and the number of successes. If the number of failures is greater than the average, a penalty factor is introduced, and the reward factor is introduced instead, as shown

$$EP = \left\{ \begin{array}{ll}
\left( \frac{1}{1 + e^{\rho \pi(t)}} - \frac{1}{2} \right) \cdot \frac{1}{1 + e^{-\rho \pi(t)}}, & \rho > 0, \\
-\left( \frac{1}{1 + e^{\theta \pi(t)}} - \frac{1}{2} \right) \cdot \frac{1}{1 + e^{-\theta \pi(t)}}, & \theta > 0.
\end{array} \right.$$ \hspace{1cm} (13)

As can be seen from equation (17), the degree of punishment and reward of the incentive mechanism for nodes is not the same. Node trust slowly increases, and trust in penalty points quickly weakens. After the incentive mechanism is introduced in the calculation of the direct trust degree calculation, the direct trust between the nodes obtained is determined by

$$DT_{d_i, d_j}(t) = C_{d_i, d_j}(t) + EP.$$ \hspace{1cm} (14)

### 3.2. Indirect Trust Assessment Based on Improved Gray Association Analysis

Most current indirect trust calculations claim that the higher the device node’s trust in other recommended devices, the more reliable it is. However, some malicious devices will carry out malicious recommendation behavior through their own good direct trust, thereby reducing the accuracy of the recommended trust of the evaluated device. In addition, the traditional trust model has the limitation of manually weighting or subjectively weighting the trust factors provided by the recommended device. This does not accurately reflect the adaptability of the trust decision process, resulting in a misjudgment of the trust level of the device node being evaluated.

The indirect trust estimation model contains the following elements. Assume that $k$ is the recommended device nodes participating in the indirect trust assessment. Among them, the number of the evaluation device nodes common to the device nodes is $n$. Define $R : \{R_1, R_2, \ldots, R_k\}$ as the recommended device set. $C : \{C_1, C_2, \ldots, C_n\}$ is the collection of public device nodes that have a history of interaction between the collection $R$ and the evaluation node. The recommended device weight set is denoted as $W = \{w_1, w_2, \ldots, w_k\}$. Here,$\sum_{i=1}^{n} w_i = 1, w_i > 0$. Compute matrix $f = [DT_{R, C_i}]_{k \times n}$.

Here, $DT_{R, C_j}$ is the recommended device $R_i$ direct trust value for public devices $C_j$.

As a systems analysis technique, gray association analysis is a method used to analyze the degree of association of multiple factors in a system. This paper uses the gray association to quantitatively compare the direct trust of public devices and recommended devices with the maximum direct trust value of a device and analyzes the degree of association of public devices with recommended devices according to the size of the difference between devices, that is, the degree of association. The greater the degree of association, the more common device trusts the device. The more important the recommended device is in the indirect trust evaluation calculation, the higher the weight. All recommended devices participating in the indirect trust calculation are normalized and weighted accordingly. Select a reference value $X$ from $f = [DT_{R, C_i}]_{k \times n}$. Each column in the matrix to form a reference sequence is denoted as $X_0(i)$. A new decision matrix is formed with the first line reference sequence and the rest consisting of direct trust between recommended devices and public device nodes. According to the theory of gray association analysis, the correlation coefficient and correlation degree of the recommended device node and the reference sequence on the public equipment indicator are calculated by using

$$\xi_{0i} = \frac{\min \min_{x} |x_0(x) - x_i(x)| + \rho \max_{x} \max_{i} |x_0(x) - x_i(x)|}{|x_0(x) - x_i(x)| + \rho \max_{x} \max_{i} |x_0(x) - x_i(x)|},$$ \hspace{1cm} (15)

$$r_{0i} = \frac{1}{n} \sum_{x=1}^{n} \xi_{0i}(x).$$ \hspace{1cm} (16)

Here, $\rho$ is the resolution coefficient, which usually takes 0.5. The size of the affinity $r_{0i}$ reflects the importance
of the corresponding recommended device. After that, calculate the weight value $w_i = r_{oi}$ for the recommended device.

Solving weights based on traditional gray associations is susceptible to the influence of $\rho$ values, resulting in uncertainty in the weight values, which affects the evaluation of indirect trust values. Therefore, this paper improves the traditional gray association solution weight method. In this paper, drawing on the idea that gray correlation is similar, the more credible values of the evaluation device node are first selected as the reference sequence. Then, the equation (16) is used to calculate and normalize the weights, and finally, the weight value of the recommended device node in the indirect trust evaluation calculation is obtained. In the calculation process, it is not necessary for the decision-maker to set the parameters subjectively, which eliminates the subjective interference of the decision-maker and makes the calculated indirect trust value more reliable. First, it is needed to determine the reference sequence. It is more trustworthy to assess the direct trust of device nodes in the public device set. Therefore, the optimal reference sequence in the decision matrix is $DT_{d,c}(t)$. Then, we need to calculate the absolute distance between the recommended device comparison sequence and the reference sequence according to

$$D_{oi} = \sum_{x=1}^{n} (x_o(x) - x_i(x))^2.$$  \hspace{1cm} (17)

In addition, we need to find the weight value for the recommended device, which is shown as follows.

$$w_i^* = \frac{1}{1 + D_{oi}}.$$  \hspace{1cm} (18)

The final step is to normalize weights for recommended devices.

$$w_i^* = \frac{w_i^*}{\sum_{i=1}^{n} w_i^*}.$$  \hspace{1cm} (19)

The weight of the recommended device in the indirect trust evaluation is calculated based on the improved gray association. The proposed algorithm takes the direct trust of the device as the reference sequence to evaluate the device node as the comparison sequence. The weight of the recommended device is then calculated based on the distance from the reference sequence, resulting in an indirect level of trust $IT_{d,d_i}(t)$, as shown in Algorithm 3.

3.3. Behavior Trust Is Calculated and Updated. Behavioral trusts are a fusion of direct and indirect trust from the evaluated node to the evaluated node. The value of the behavior trust is calculated based on the weights of both. The weight value satisfies $w_1 + w_2 = 1$. This weight can be calculated based on the information entropy of direct trust and indirect trust.

$$TR_{d,d_i}(t) = w_1 \cdot DT_{d,d_i}(t) + w_2 \cdot IT_{d,d_i}(t).$$  \hspace{1cm} (20)

In an edge computing environment, the calculated behavioral trust value is not immutable. In the case of abnormal device nodes, rapid or unpredictable situations may occur during the interaction of two nodes. This makes the computed evaluated nodes unreliable. Therefore, the generated behavior trust value needs to be dynamically updated in conjunction with the previous interaction.

$$BT_{d,d_i}(t) = \alpha_t \cdot TR_{d,d_i}(t - 1) + (1 - \alpha_t) TR_{d,d_i}(t).$$  \hspace{1cm} (21)

Here, $\alpha_t$ refers to the trust dynamic update factor, which is calculated as follows.

$$\alpha_t = \begin{cases} \alpha_{t-1}, & s(t) = s(t - 1), \\ \alpha_{t-1} \cdot \left[ 1 - \frac{s(t)}{s(t) + r(t)} \right], & s(t) > s(t - 1). \end{cases}$$  \hspace{1cm} (22)

As mentioned above, it is dynamically changed according to the interaction failure rate of the two device nodes at the $t$-time. This value decreases as the number of failures increases. The behavior trust level calculation and dynamic update process is shown in Algorithm 4.

According to the direct trust calculation process, the time complexity of calculating direct trust is $O(n)$. As can be seen by Algorithm 1, the time complexity of the indirect trust calculation is determined by the number of executions of Algorithm 1. Since the maximum number of loops is reached, $n \times k$ the indirect trust calculation process complexity is $O(n \times k)$. Behavioral trust level calculations and dynamic updates are a combination of direct and indirect trust level calculations. Therefore, the time complexity of the synthetic trust calculation algorithm is $O(n \times k)$.

3.4. Comprehensive Trust Level for Edge Devices. Identity trust refers to the legitimacy of the assessed edge device node to its own identity in the edge computing environment. Therefore, the value of the identity trust $AT_{t,d_i}$ is a binary number. If certified, then $AT_{t,d_i} = 1$. Otherwise, $AT_{t,d_i} = 0$.

The comprehensive trust level is derived from the identity authentication value of the terminal device and the behavioral trust level of the terminal device. There are only two outcomes of authentication that produce authentication success and failure. In the process of computing the overall trust level of the device, identity authentication is only used as a secondary factor in calculating the trust level. Device authentication can still be interacted with if the device identity is not passed, and its trust level must be lower than that of other devices.

The composite trust level is calculated based on the weight of identity trust and behavioral trust, as shown in equation (23). The weight value satisfies $w_3 + w_4 = 1$. The weight value is determined by the calculation $\tilde{d}_i$ process.
This section describes the experimental environment and the effectiveness and accuracy of the proposed dynamic trust evaluation model for edge devices from different aspects through simulation experiments.

4. Simulation and Analysis

This section describes the experimental environment and verifies the effectiveness and accuracy of the proposed dynamic trust evaluation model for edge devices from different aspects through simulation experiments.

4.1. Experimental Environment. The operating environment is an Ubuntu 12.04-64-bit system. The computer uses a 3.10 GHz Intel(R) Core(TM)i5-3230M CPU and 8 GB of RAM. To bring the experiment closer to the edge computing environment, in the NetLogo-6.0.4-64 emulator, refer to the Random Waypoint mobile node random distribution generation model. Within the deployment area of 500 × 500, generate 1000 mobile nodes as edge device nodes in the edge network. In addition, this article numbers edge device nodes to interact between them 10 to 50 times. In addition, this article passes the interaction record between devices to MATLAB for calculation through the interface module to realize the data interaction of the entire system network. The parameters in the simulation test are set as shown in Table 1.

4.2. Test Results and Analysis. Scyther is one of the more popular automated analysis tools [18]. This tool has advantages over other tools in terms of attack output and security model for validating protocols. The analysis tool is developed in the Ubuntu operating system using the Python language and displays the protocol verification results in the form of a human-computer interaction interface and analyzes and improves the protocol more intuitively. Figure 4 shows the result of validating the SM9 authentication protocol using the Scyther tool. A random number N is added to the verification process to ensure the timeliness of the message. The addition of identity in the process of transmitting messages effectively protects against replay attacks and effectively resists the risk of impersonation attacks. In the SM9-based identification password algorithm, a malicious attacker can obtain the legitimate information of the device before the key can be calculated. Therefore, the protocol has a high level of security.

The metrics of detection rate and the false detection rate are reliable for analyzing the behavioral trust model of end devices. Define the detection rate of the model as the ratio of the number of detected rogue device nodes to the total
number of malicious device nodes. False detection includes two scenarios (normal device nodes are mistakenly detected as malicious device nodes, and malicious device nodes are mistakenly detected as normal device nodes). The false detection rate is the ratio of the number of falsely detected device nodes to the total number of device nodes detected. In this simulation test, in order to obtain a more accurate detection rate and false detection rate, five sets of tests were conducted, including 5%, 10%, 15%, 20%, 25%, and 30% of the malicious device nodes deployed in the experimental environment. To verify that the proposed algorithm can effectively detect malicious nodes, simulation experiments compared models without satisfaction functions with Bayesian-based trust evaluations in the literature.

Figures 5 and 6 show the false detection rates and detection rates of the three scenario models under different scales of malicious device nodes.

From Figure 5, it can be seen that the false detection rate increases as the proportion of malicious device nodes increases. However, the average false detection rates of the proposed algorithm, nonsatisfaction function model [19], and RFSN model [20] were 7.67%, 9.08%, and 14.28%, respectively. In contrast, the proposed algorithm has the lowest false detection rate, mainly because the satisfaction function is introduced in this paper. The performance satisfaction evaluation of the evaluated device node by the main device node avoids the influence of nonintrusion factors that cause the interaction between the two device nodes to fail and the error judgment is avoided. Therefore, the false detection rate of device nodes judged by interactive records is reduced. As can be seen from the figure, the performance of the proposed algorithm is significantly better than that of the other two models.

From Figure 6, it can be seen that the detection rate of the proposed algorithm is significantly higher than that of the other two schemes. The proposed algorithm detects 80% higher detection rates in five sets of malicious device nodes with different ratios. Since the trust evaluation model in the literature only spreads the good reputation of the node, the evaluation of trust is not accurate. In addition, the model also ignores that the trust value of malicious nodes is too high due to malicious exaggerated feedback, resulting in a decrease in the detection rate. The proposed algorithm increases the detection rate of malicious device nodes to some extent by adding a satisfaction function.

Using the data provided in the matrix combined with the proposed algorithm, the traditional entropy, and the typical gray association to calculate the weights of the recommended devices, the test results are shown in Table 2. RFSN is based on the Bayesian theory trust evaluation model, the model uses the Bayesian formula for reputation representation, update, integration, and trust evolution and provides a unified way to detect malicious behavior of nodes and improve the interaction success rate, but the model has an interactive experience that does not share nodes, and the amount of computation is large at the expense of system efficiency.

In the improved gray association method, the subjective evaluation of the subject is added as the optimal set, and the closer to the recommended device of the optimal set, the higher the weight. Due to the weight normalization, the weight difference of 0.01 when there are more recommended devices involved is also large, and the weight solution method based on the improved gray correlation analysis method is more reasonable than the traditional entropy weight solution method and the traditional gray correlation weight solution method, and the closer to the optimal device set, the greater the device weight value.

Using this method, comparing the interaction failure rate of the entropy-based evaluation model, the traditional gray correlation method, and the RFSN model, it can be seen from Figure 7 that there is no obvious gap between the three information models of the RFSN model in the case of a low proportion of malicious devices, with the increase in the proportion of malicious devices. The model interaction failure rate based on the improved gray association analysis method is significantly lower than that of the other three models, mainly because the accuracy of the weights of the recommended devices based on the improved gray association analysis method reduces the interaction failure rate between devices.

The edge device dynamic trust evaluation model in the proposed algorithm proposed in this paper shows that the overhead is an important evaluation index in the process.
Figure 5: False detection rate vs. percentage of malicious devices.

Figure 6: Detection rate vs. percentage of malicious devices.

Figure 7: Interaction failure rate vs. percentage of malicious devices.

Table 2: Comparisons of weight.

| Methods                  | $\omega_1$ | $\omega_2$ | $\omega_3$ | ... | $\omega_5$ | $\omega_6$ | $\omega_{10}$ |
|--------------------------|------------|------------|------------|-----|------------|------------|--------------|
| The proposed algorithm   | 0.106      | 0.092      | 0.096      | ... | 0.112      | 0.081      | 0.147        |
| Traditional entropy      | 0.101      | 0.105      | 0.094      | ... | 0.083      | 0.092      | 0.117        |
| Gray association         | 0.519      | 0.687      | 0.771      | ... | 0.745      | 0.689      | 0.871        |
of trust evaluation of device nodes. Therefore, by comparing the time costs of the three different scenarios of the proposed model, the RFSN model, and the hierarchical trust mechanism based on fog computing hierarchical trust mechanism (FHTM) [20], the time overhead is mainly derived from the recommended trust evaluation in the process of trust evaluation process. During the test, in order to make the time differentiating, 6 different groups of recommended devices [50, 100, 150, 200, 250, 300] were selected, due to the occasional system running time calculated by each run of the program, in order to accurately record the time overhead. Record trust data in each set of different sets of devices tested 10 times under different methods and record averages of uptime.

As can be seen from Figure 8, when the number of participating devices is small, the system consumption due to the evaluation mechanism of the proposed algorithm and FHTM models is mainly concentrated in 3 aspects: direct trust, recommended trust, and trust update, making the model proposed in this article. The trust evaluation system time overhead of the FHTM model is slightly higher than that of the RFSN model, but as the number of participating devices increases, the trust evaluation of the RFSN model is iterative throughout the network, resulting in an increase in the number of devices. The number of data lookups increases; the amount of converged data increases, resulting in a rapid increase in the time overhead based on the model, while the proposed algorithm comprehensive trust evaluation is calculated by the network layer nodes, which reduces the computing time of edge devices, improves the efficiency of the system, reduces the global convergence time of trust aggregation, and makes the time overhead grow slowly. Due to the large number of edge devices in edge computing and the increasing number of edge devices interacting with the nodes of the device being evaluated, the proposed algorithm time overhead is superior to the other two models in environments with a large number of devices.

For the dynamic trust evaluation model proposed in this paper, the trust value of the evaluated device is simulated from two aspects, because the trust model of this paper adds an incentive mechanism and a dynamic update factor in the behavior trust stage, so that when the edge device interacts, the trust value rises slowly, and if the number of interaction failures increases, the trust value of the evaluated device will decrease rapidly. In this simulation experiment, the same device performs a behavioral trust assessment on 3 different

![Figure 8: The system cost vs. the number of devices.](image1)

![Figure 9: Trust value vs. time.](image2)
Interactive devices (denoted as device 1, device 2, and device 3) over a certain period of time.

The initial values for the experiment are 0.18, 0.37, and 0.8. Device 1 is a device with a good level of trust, and the interaction recording is successful during that time period; device 2 has a high rate of interaction recording failures at a certain point in the time period; and device 3 has a high initial state trust but has been in a state of interaction failure for a long time. As can be seen from Figure 9, although the initial value of device 1 is low, the trust value has been steadily rising over the time period due to the high success rate of long-term interaction recording. Device 3 causes the trust value to drop rapidly and eventually tend to 0 due to the high failure rate of the interaction result during the interaction. The initial trust value of device 2 is 0.37, and the trust value also rises steadily due to the good interaction record of device 2 during the 0 to 160 periods, but at 160. At any time, it is detected that the device has malicious behavior, resulting in a rapid decline in the trust value of the device. It can be seen from this experiment that in the trust evaluation model of adding incentive mechanisms, the trust value rises and falls at different speeds under different behavior device nodes, and the behavior of malicious node dishonest feedback is suppressed from the trust of the control node.

5. Conclusions

In the paper, we propose the novel trust evaluation mechanism model for edge devices of the IoT. In the simulation process, the trust model and protocol security, the weight accuracy of the recommended device, the authentication bandwidth and time overhead, and the trust value of different behaviors are simulated. Simulation results verify that the trust evaluation model proposed in this paper has certain advantages for the reliability, accuracy, and efficiency of trust evaluation.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by research and system development of active distribution network dispatching and digital service technology.

References

[1] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, “A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2019.

[2] J. Xia, G. Cheng, S. Gu, and D. Guo, “Secure and trust-oriented edge storage for Internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4049–4060, 2020.

[3] S. Pinto, T. Gomes, J. Pereira, J. Cabral, and A. Tavares, “ILO-TEED: an enhanced, trusted execution environment for industrial IoT edge devices,” *IEEE Internet Computing*, vol. 21, no. 1, pp. 40–47, 2017.

[4] T. Wang, L. Qiu, A. K. Sangaih, A. Liu, M. Z. A. Bhuiyan, and Y. Ma, “Edge-computing-based trustworthy data collection model in the Internet of things,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4218–4227, 2020.

[5] Y. He, Y. Wang, C. Qiu, Q. Lin, J. Li, and Z. Ming, “Blockchain-based edge computing resource allocation in IoT: a deep reinforcement learning approach,” in *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2226–2237, 2021.

[6] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, and L. Qi, “Trust-oriented IoT service placement for smart cities in edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4084–4091, 2020.

[7] S. Guo, Y. Dai, S. Xu, X. Qiu, and F. Qi, “Trusted cloud-edge network resource management: DRL-Driven service function chain orchestration for IoT,” *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6010–6022, 2020.

[8] J. Chi, E. Owusu, X. Yin et al., “Privacy partition: a privacy-preserving framework for deep neural networks in edge networks,” *IEEE/ACM Symposium on Edge Computing (SEC)*, vol. 2018, pp. 378–380, 2018.

[9] I. U. Din, A. Bano, K. A. Awan, A. Almogren, A. Altameem, and M. Guizani, “LightTrust: lightweight trust management for edge devices in industrial Internet of things,” *IEEE Internet of Things Journal*, p. 1, 2021.

[10] M. S. Haghighi, M. Ebrahimi, S. Garg, and A. Jolfaei, “Intelligent trust-based public-key management for IoT by linking edge devices in a fog architecture,” *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12716–12723, 2021.

[11] I. Yen, F. Bastani, N. Solanki, Y. Huang, and S. Hwang, “Trustworthy computing in the dynamic IoT cloud,” in *IEEE International Conference on Information Reuse and Integration (IRI)*, vol. 2018, pp. 411–418, IEEE, Salt Lake City, UT, USA, 2018.

[12] M. Bradbury, A. Jhumka, and T. Watson, “Trust trackers for computation offloading in edge-based IoT networks,” in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications*, pp. 1–10, IEEE, Vancouver, BC, Canada, 2021.

[13] Y. Hussain, H. Zhiqiu, M. A. Akbar et al., “Context-aware trust and reputation model for fog-based IoT,” *IEEE Access*, vol. 8, pp. 31622–31632, 2020.

[14] R. T. Tiburski, C. R. Moratelli, S. F. Johann et al., “Lightweight security architecture based on embedded virtualization and trust mechanisms for IoT edge devices,” *IEEE Communications Magazine*, vol. 57, no. 2, pp. 67–73, 2019.

[15] S. Sathiyaevan, K. Achuthan, R. Doss, and L. Pan, “Protein authentication scheme – a time-bound dynamic key Gen authentication technique for IoT edge nodes in outdoor deployments,” *IEEE Access*, vol. 7, pp. 92419–92435, 2019.

[16] K. Suzuki, K. Nakajima, T. Oi, and A. Tsukamoto, “TS-Perf: general performance measurement of trusted execution environment and rich execution environment on Intel SGX, Arm TrustZone, and RISC-V Keystone,” *IEEE Access*, vol. 9, pp. 13520–13530, 2021.

[17] J. Jiang, X. Zhang, and S. Li, “A task offloading method with edge for 5G-envisioned cyber-physical-social systems security and communication,” *Networks*, vol. 2020, article 8867094, 9 pages, 2020.
P. Abeysekara, H. Dong, and A. K. Qin, “Machine learning-driven trust prediction for MEC-based IoT services,” in *IEEE International Conference on Web Services (ICWS)*, vol. 2019, pp. 188–192, Milan, Italy, 2019.

I.-H. Chuang, S.-H. Huang, W.-C. Chao, J.-S. Tsai, and Y.-H. Kuo, “TIDES: a trust-aware IoT data economic system with blockchain-enabled multi-access edge computing,” *IEEE Access*, vol. 8, pp. 85839–85855, 2020.

M. Dai, S. Xu, Z. Wang, H. Ma, and X. Qiu, “Edge trusted sharing: task-driven decentralized resources collaborate in IoT,” *IEEE Internet of Things Journal*, p. 1, 2021.