Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.047243

ARTICLE

A Trust Evaluation Mechanism Based on Autoencoder Clustering Algorithm for Edge Device Access of IoT

Xiao Feng[1,2,3,*] **and Zheng Yuan**[1]

[1]Cyber Security Academy, Beijing University of Posts and Telecommunications, Beijing, 100876, China

[2]Department of Cyberspace Security, Beijing Electronic Science and Technology Institute, Beijing, 100070, China

[3]State Grid Info-Telecom Great Power Science and Technology Co., Ltd., Beijing, 102211, China

*Corresponding Author: Xiao Feng. Email: fengxiao@bupt.edu.cn

Received: 30 October 2023 Accepted: 19 December 2023 Published: 27 February 2024

ABSTRACT

First, we propose a cross-domain authentication architecture based on trust evaluation mechanism, including registration, certificate issuance, and cross-domain authentication processes. A direct trust evaluation mechanism based on the time decay factor is proposed, taking into account the influence of historical interaction records. We weight the time attenuation factor to each historical interaction record for updating and got the new historical record data. We refer to the beta distribution to enhance the flexibility and adaptability of the direct trust assessment model to better capture time trends in the historical record. Then we propose an autoencoder-based trust clustering algorithm. We perform feature extraction based on autoencoders. Kullback leibler (KL) divergence is used to calculate the reconstruction error. When constructing a convolutional autoencoder, we introduce convolutional neural networks to improve training efficiency and introduce sparse constraints into the hidden layer of the autoencoder. The sparse penalty term in the loss function measures the difference through the KL divergence. Trust clustering is performed based on the density based spatial clustering of applications with noise (DBSCAN) clustering algorithm. During the clustering process, edge nodes have a variety of trustworthy attribute characteristics. We assign different attribute weights according to the relative importance of each attribute in the clustering process, and a larger weight means that the attribute occupies a greater weight in the calculation of distance. Finally, we introduced adaptive weights to calculate comprehensive trust evaluation. Simulation experiments prove that our trust evaluation mechanism has excellent reliability and accuracy.

KEYWORDS

Cross-domain authentication; trust evaluation; autoencoder

1 Introduction

The concept of the Internet of Things (IoT) has affected all walks of life in society. However, there are many issues in terms of security and user privacy during the implementation of the concept. In the field of Internet of Things, cloud computing is an important research direction. It is crucial to design a credible trust evaluation mechanism to ensure the reliability of data [1]. The traditional cloud computing model often results in high latency because computing and data

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

storage are centralized in remote data centers, which makes real-time response requirements unmet for applications such as intelligent transportation or industrial automation [2]. Second, this model can cause network congestion problems, as many devices attempt to upload and download data simultaneously, leading to packet loss and decreased network performance [3]. In addition, with the flow of data during transmission and storage, privacy and security concerns cannot be ignored [4]. Finally, cloud computing models often require a lot of data center resources [5], but these resources are not always fully utilized, especially for lightweight computing tasks, and this highly resource-intensive model can be wasteful.

Edge computing can greatly improve the shortcomings of traditional cloud computing. Allocate computing and data processing tasks to edge devices to reduce network latency. This has been widely used in smart cities [6]. Processing data through edge devices reduces the computing pressure on the cloud. thereby reducing the risk of network collapse and improving network security [7]. But there are also many potentials and challenges in the current edge computing research field. For example, computing resources limit edge computing task processing performance. The dynamic nature of the device environment also adds to the complexity of edge computing task processing [8].

The low detection rate of trust evaluation mechanisms is one of the challenges facing the edge computing research field. There may be dynamically adaptive malicious nodes in the network. They will evade detection by pretending to be honest nodes or accumulating trust value in a short period. Therefore, a trust mechanism that can dynamically monitor node changes is crucial to ensuring the reliability of edge computing systems [9].

How to improve task allocation and processing efficiency is also a major challenge in the current research field. When computing resources are limited, factors such as network status and device performance need to be considered when processing large-scale computing tasks. Therefore, intelligent algorithms with excellent performance are needed to assist in improving computing efficiency [10].

In conclusion, edge computing is a powerful tool for addressing the needs of the IOT era, but to fully leverage its benefits, issues, and challenges such as trust assessment and task processing efficiency must be addressed. To address the above challenges, this study proposes a novel cross-domain authentication architecture, which is built based on a trust evaluation mechanism. The research contributions of this authentication architecture are as follows:

1) We introduce a credible cross-domain authentication process, encompassing registration, certificate issuance, and cross-domain authentication procedures to ensure secure interactions among devices in different domains. A direct trust evaluation mechanism based on the time decay factor is proposed, taking into account the influence of historical interaction records. We weighted the time attenuation factor to each historical interaction record for updating and got the new historical record data. We refer to the beta distribution to enhance the flexibility and adaptability of the direct trust assessment model to better capture time trends in the historical record.

2) We propose an autoencoder-based trust clustering algorithm. Feature extraction is based on autoencoders. KL divergence is used to calculate the reconstruction error. When constructing a convolutional autoencoder, we introduce convolutional neural networks to improve training efficiency and introduce sparse constraints into the hidden layer of the autoencoder. The sparse penalty term in the loss function measures the difference through the KL divergence.

3) Trust clustering is performed based on the density based spatial clustering of applications with noise (DBSCAN) clustering algorithm. During the clustering process, edge nodes have a variety of trustworthy attribute characteristics. We assign different attribute weights according to the relative

importance of each attribute in the clustering process, and a larger weight means that the attribute occupies a greater weight in the calculation of distance.

4) We combine the adaptive weight calculation method to obtain a comprehensive trust evaluation score. Simulation experiments prove that the trust evaluation mechanism proposed in this article has excellent performance in terms of reliability and accuracy.

The structure of this article is as follows. In Section 2, we present research on edge computing and trust assessment. In Section 3, we propose a cross-domain authentication framework based on trust evaluation mechanism. In Section 4, we describe a trust evaluation algorithm based on autoencoder clustering. In Section 5, we carry out simulation experiment analysis. In section 6, we summarize.

2 Related Work

Trust assessment plays a key role in the edge computing research field. Xu et al. [11] proposed an innovative trust evaluation model, which is based on risk and feedback. The research of Guo et al. [12] focused on security in an edge computing environment. Ensure that each node is trusted and that only authorized nodes can join the execution environment. The authentication method is used to verify the identity of the node. The work of Chi et al. [13] analyzed the trust domains between different devices and entities, which are composed of devices that trust each other. The establishment and management of this relationship are essential to ensure legitimacy and credibility. Din et al. [14] proposed that through the establishment of trust relationships and the use of social trust networks, the communication reliability between nodes in the cell network has been significantly improved.

Kuang et al. [15] studied how to reasonably allocate tasks to devices and nodes when storage and computing resources are limited. Jošilo et al [16] optimized task processing time through the efficient allocation of unlimited resources to unnamed devices. Dynamically obtain edge computing resources through devices to improve resource utilization and reduce overall network latency. Valerio et al. [17] aimed to solve the problem of virtual machine allocation in edge computing. The model considers energy consumption, load, and other factors, and adopts the Markov decision method to achieve the goal of minimizing delay. Ouyang et al. [18,19] proposed a UAV mission offloading trust scheme based on two key indicators, namely energy consumption and service reliability.

Multi-access edge computing (MEC) [20,21] mainly studies how to perform low-latency data retrieval. However, edge server storage resources are limited. CSEdge solves the trust problem in storage based on blockchain. It also solves the incentive problem and allocates storage space based on reputation value. Assessing credibility through consensus algorithms provides a solution for edge storage [22]. BEoT [23] combines edge computing and blockchain. There are application scenarios in many fields. Smart contracts and identity authentication are applied to improve the security of the architecture.

Vehicle edge computing utilizes vehicles (VEC) [24] to reduce application latency by using vehicles to migrate computing tasks. Prices are calculated based on the two-stage Stackelberg game model. The VEC server serves as the leader and the vehicle serves as the follower. Through deep reinforcement learning, a management strategy for non-shared computing needs is designed to maximize the profits of both parties [25]. Cvitić et al. designed a classification model based on logit boost [26]. A classification model with high accuracy is trained through multiple traffic characteristics of IoT devices. Li et al. proposed a multi-level power grid model that focused on analyzing the vulnerability of components [27].

3 Cross-Domain Authentication Architecture Based on Trust Evaluation Mechanism

In edge computing network systems, changes in edge nodes will increase network instability. Based on this problem, we propose an authentication architecture based on a trust evaluation mechanism and blockchain.

We summarized the main symbols in Table 1.

Table 1: Symbol specification

Symbol	Specification
N	The set of edge nodes
E	The set of evaluation nodes
t	The time
$VD_{n_i,n_j}(t)$	The direct trust value of n_j on-time t
$VI_{n_i,n_j}(t)$	The indirect trust value of n_j on-time t
$\mathrm{q}_{n_i,n_j}(t)$	The task performance score between n_i and n_j
$\lambda^t_{n_i,n_j}$	The ratio of successful task execution to total interactions
α	Regulatory factor
$\sum \mathrm{q}_{n_i,n_j}(t)^+$	Successful interaction of data
μ	Evaluation node weight
$\mathrm{r}^t_{i,j}$	Node interaction times
$MinPts$	The minimum number of sample points in the neighborhood
ND	Neighborhood distance
θ	Neighborhood radius
β	Edge node attribute weight
F	Feature node set
S	Edge computing task

3.1 Cross-Domain Authentication Architecture

At present, the field of edge computing is becoming increasingly mature and diversified, and a variety of computing architectures have been developed for various application scenarios. This provides a solid foundation for realizing more efficient resource collaborative utilization and application sharing. The trusted authentication architecture based on trust evaluation is shown in Fig. 1.

A good edge computing trust model should be able to cope with most malicious attack modes. The existing trust model has insufficient ability to identify and prevent, and there are shortcomings in security management. It is unable to deal with complex attack methods such as swing attacks and collusion attacks. So that malicious nodes can still hide in the network after malicious attacks.

3.2 Cross-Domain Authentication Process

To ensure the accuracy of the study and facilitate reference, we summarized the main symbols in Table 1.

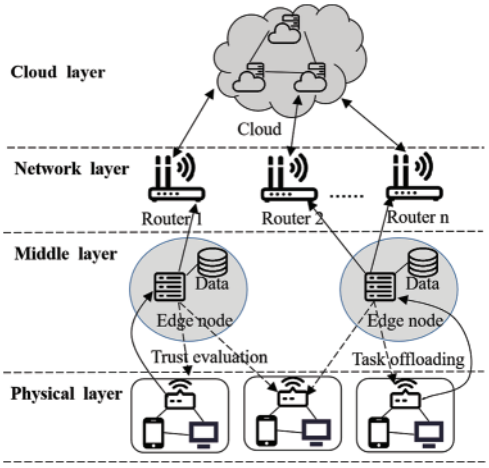

Figure 1: Cross-domain authentication architecture

3.2.1 Registration Process

First, the edge server generates identity tags LA_i, public and private keys, (PK_i, PV_i) and timestamps T_i for the node n_j. This information marks the uniqueness of the node's identity. All information is encrypted and stored on the blockchain, accessible only to edge servers. Then the edge server sends a message $Request_i$ to the blockchain node to apply for a registered node.

$$Request_i = (LA_i, PK_i, T_i, SG\,(PV_i, LA_i||T_i)) \tag{1}$$

After receiving $Request_i$ the blockchain node performs signature verification to ensure that the source is legitimate and has not been tampered with. If the digital signature passes, the registration information will be written to the blockchain. The registration is successful.

3.2.2 Certificate Issuance Process

The certificate issuance process is a key step for nodes in the same domain to apply for certificates CA from edge servers. This process serves as the basis for trust establishment and identity authentication in edge computing environments. The edge server first queries the identity information and public key $\left(LA_j, PK_j\right)$ of the certificate-issuing node through the blockchain. This information is key for subsequent verification and communication. Then send the node information I_i to the blockchain node together.

$$I_{\mathrm{i}} = \left(LA_i, T_j, SG\left(PK_i, LA_i||T_j\right)\right)||EP\left(PK_j, \mathrm{Re}\right) \tag{2}$$

If the information is correct, the blockchain server generates a number NUM_i and certificate information CA_i and sends them to the application node. The certificate is successfully issued. The certificate information is shown below:

$$CA_i = EP\left(PK_i, NUM_i||T_j\right) \tag{3}$$

3.2.3 Cross-Domain Authentication Process

Assume that the source node of the domain A needs to establish communication with the destination node of domain B. First, the domain A edge server needs to query the domain B edge

server's public key PK_B and label LA_B through the blockchain and generate identity information $Message._i$. Then n_i use PK_B encryption $Message._A$ to generate a request message $Request_A$ and send it to n_B.

$$Message._i = (LA_i, CA, T_B, SG(PV_i, CA||T_B)) \tag{4}$$

$$Request_A = EP(PK_B, Message_A) \tag{5}$$

After the edge server of the domain B receives the request message $Request_A$, the server first decrypts $Request_A$ through the private key PV_B to obtain the identity information $Message._A$. Then query the public key PK_i of the source node n_A and the public key PK_A of the server. The server of the domain B will perform verification, and if all verifications pass, subsequent services will be provided. Failure to authenticate will result in a denial of service.

3.3 Direct Trust Evaluation Mechanism

Direct trust is an important concept that can be measured. In the network, we can quantify the task execution performance of nodes to define and calculate trust. $VD_{n_i,n_j}(t)$ present node n_j trust value at a time t.

Suppose that the task during the interaction between node n_i and node n_j in time t is represented as a set of time series probability data:

$$q_{n_i,n_j}(t) = \left\{\lambda^1_{n_i,n_j}, .\lambda^2_{n_i,n_j}, \ldots, \lambda^t_{n_i,n_j}\right\} \tag{6}$$

where $\lambda^t_{n_i,n_j} \in [0,1]$ represents the ratio of the amount of successfully executed tasks and to the total number of task interactions within the time. We use this metric to describe the success rate of cooperative tasks. It expresses the probability of mission success. A record of the interaction between the two nodes during this time window will serve as the basis for evaluating direct trust.

However, it is important to note that the effectiveness of interactive recording diminishes over time. This is because early cooperative behavior may no longer reflect the current state of trust. We add weights to reduce the impact of time. This factor takes into account the temporal impact of historical interaction records, ensuring that time is taken into account when assessing direct trust. The definition is as follows:

$$\varepsilon_i = e^{-(t-t_i)} \tag{7}$$

$$\lambda^{t_i}_{n_i,n_j} = \varepsilon_i \lambda^{t_i}_{n_i,n_j} \tag{8}$$

where ε_i is the time attenuation factor of t_i time, we weighted the time attenuation factor to each historical interaction record for updating, and got the new historical record data:

$$q_{n_i,n_j}(t) = \left\{\varepsilon_i\lambda^1_{n_i,n_j}, .\varepsilon_2\lambda^2_{n_i,n_j}, .., \varepsilon_i\lambda^{t_i}_{n_i,n_j}, \ldots, \varepsilon_i\lambda^t_{n_i,n_j}\right\} \tag{9}$$

The Beta distribution has flexible capabilities, and we refer to the beta distribution to enhance the flexibility and adaptability of the direct trust assessment model to better capture time trends in the historical record.

$$VD_{n_i,n_j}(t) = \frac{\sum_{i=1}^{t} \varepsilon_i \lambda^{t_i}_{n_i,n_j}}{\sum_{i=1}^{t} \varepsilon_i}\alpha \tag{10}$$

$$\alpha = 1 - \frac{1}{\sum \mathrm{q}_{n_i,n_j}(t)^{+} + \delta} \tag{11}$$

where α is regulator of direct trust value, $\sum \mathrm{q}_{n_i,n_j}(t)^{+}$ represents the sum of success records.

4 Trust Evaluation Mechanism Based on Autoencoder Clustering

In this paper, indirect trust assessment is based on a distributed trust establishment approach. Firstly, the intermediate trust assessment module collects trust feedback information from various nodes in the entire network. The value of these feedback pieces of information is significant as they reflect the trustworthiness exhibited by the assessed nodes when interacting with other nodes. Evaluating nodes gather these feedback inputs from multiple nodes and consolidate them into an overall trust assessment. This consolidation process employs a trust aggregation algorithm. The flow chart of trust assessment is shown in Fig. 2.

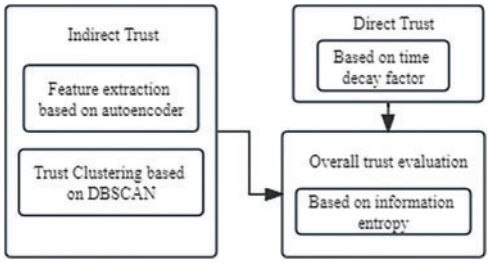

Figure 2: Trust assessment flow chart

This dynamic information is expressed in the form of $VI_{\mathrm{n}_i}(t)$, which clearly shows the growth of the trust value of the node n_i, and provides the nodes in the network with the latest information about the behavior of the node being evaluated. This dynamic trust-building process helps nodes in the network better adapt to changing trust needs and environmental conditions.

Assume that there are m evaluation nodes in the middle layer, and the set of nodes is $E = \{e_1, e_2, \ldots, e_m\}$. For evaluation node e_i, interacts with all device nodes. As the number of interactions increases, it means that the more frequently two parties interact, the greater the likelihood of trust. So we set weights for the evaluation nodes:

$$\mu_{i,j} = \frac{\mathrm{r}_{i,j}{}^{t}}{\sum_{i=1}^{m} \mathrm{r}_{i,j}^{t}} \tag{12}$$

where $\mathrm{r}_{i,j}^{t}$ is the number of communications between nodes within the time t.

The indirect trust value is expressed as:

$$VI_{\mathrm{n}_i,\mathrm{n}_j}(t) = \sum_{i=1}^{m} r_{i,j}^{t} VD_{\mathrm{n}_i,\mathrm{n}_j}(t) \tag{13}$$

Evaluating whether a node is trustworthy is crucial to the evaluation of indirect trust. If the recommendation node takes malicious actions for personal gain, such as delivering false reports to the trust evaluation node, then this will seriously interfere with the final trust evaluation of the target node.

4.1 Feature Extraction Based on Convolutional Autoencoder

Firstly, we introduce an autoencoder for feature extraction. The autoencoder first is the input layer, where the user evaluation data is encoded as a high-dimensional feature vector. This is followed by the hidden layer, which is responsible for feature extraction, and the output layer, which is responsible for decoding the feature vectors to reconstruct the data. Autoencoders measure the efficiency of feature extraction by comparing pre-encoding and post-decoding feature vectors. This is achieved by calculating the reconstruction error. In node clustering, KL divergence is usually used to calculate the reconstruction error. KL divergence is a measure used to measure the difference between two probability distributions. The lower KL divergence indicates that the similarity between the two distributions is higher, that is, the feature extraction effect is better. KL divergence is calculated as follows:

$$KL_i = -\sum\nolimits_{i=1}^{n} \{g_i \log [h_i + (1 - g_i) \log (1 - h_i)]\} \tag{14}$$

where g_i and h_i represent the input and output feature vectors, respectively.

When constructing a convolutional autoencoder, we introduce convolutional neural networks to improve training efficiency. It is an effective regularization technique to introduce sparse constraints into the hidden layer of the autoencoder. By leaving most neurons inactive, the complexity of the network is reduced and overfitting is prevented. This helps the model to better generalize to previously unseen data. By calculating the mean activation of each hidden neuron avg_j, we can learn which neurons are active and which are inactive. The sparse penalty term in the loss function measures the difference through the KL divergence. This allows the network to learn from fewer activated neurons, thereby reducing the risk of overfitting.

Suppose av_i^j is the activation value of neurons j, N is the number of nodes, and K is the number of hidden layer neurons. The mean activation of neurons j is:

$$avg_j = \frac{\sum_i^N av_i^j}{N} \tag{15}$$

The loss function is expressed as:

$$J = L + \varphi \sum\nolimits_{i=1}^{M} KL_i \tag{16}$$

where φ is the sparse penalty coefficient and KL_i measures similarity based on KL divergence.

Once the encoder is trained, the value of the loss function is relatively small, indicating that the encoder has learned an effective feature representation. Then we can use the encoder to extract the features.

4.2 Trust Clustering Based on DBSCAN

In this section, the DBSCAN algorithm is introduced based on auto-encoder feature extraction. DBSCAN organizes data by the density of sample points and performs clustering according to the density of data points. The algorithm first defines the concept of neighborhood. DBSCAN defines the neighborhood of each data point through the neighborhood radius, that is, the distance θ from the point is considered as the neighborhood.

DBSCAN defines a core point as a point that contains at least *MinPts* sample points in the neighborhood. That is, if the number of nodes in the neighborhood of the node x is greater than *Minpts*, then x is the core point. Assuming a core point x_1, if the neighborhood x_1 contains x_2, then

the x_1 density is direct x_2. DBSCAN uses the core point as the starting point to gather the points with direct density into the same cluster, thus achieving data clustering operations.

During the clustering process, edge nodes have a variety of trustworthy attribute characteristics. We assign different attribute weights according to the relative importance of each attribute in the clustering process, and a larger weight means that the attribute occupies a greater weight in the calculation of distance. Assuming a total of M attributes of edge nodes, the neighborhood distance ND of any node r and node s is calculated as follows:

$$ND = \sqrt{\sum_{i=1}^{M} \beta_i \left(\mathrm{r}_i - s_i\right)^2} \tag{17}$$

Algorithm 1: Trust clustering algorithm based on autoencoder

Input: Neighborhood radius θ; Minimum of neighbor nodes *Minpts*; Feature node set F
Output: min F (G)

```
Begin
 for each unvisited point f in the F do
     mark f as visited point
     find neighbor nodes F_b of f
     if F_b nodes ≤MinPts then
       Mark f as noise
     else then
        create cluster C and add f_b to C
        for each point f_b in the F_b do
           if f_b is not visited then
               mark f_b as visited
               find neighbor nodes F'_b of f_b
               if neighbor nodes F'_b ≥ MinPts then
                  add F'_b to F_b
               end if
           end if
           if f_b is not yet a member of any cluster then
              add f_b to cluster C
           end if
        end for
     end if
  end for
  Solve according to the constraints;
  return min F (G)
End
```

Then we describe the clustering process. First, we randomly select a node as the starting point and calculate the number of neighbor nodes. If the number of neighbor nodes is greater than *MinPts*, The node's neighbor nodes are added to the node cluster. The neighborhood expansion of the newly added points is continued to ensure that all reachable core points are added to the cluster. If the neighborhood of a core point contains other core points, they will be connected to the same cluster. All data points that are not assigned to the cluster are labeled as noise points. These points do not belong to any cluster. Repeat the above process until all data points have been accessed and classified. We end up

with one or more clusters, each containing a set of data points dense with each other, while the noise points do not belong to any cluster. The specific algorithm flow is shown in Algorithm 1.

Algorithm 2: Global trust evaluation based on entropy and adaptive weights

Input: Edge nodes set N; Evaluation nodes set E; Interactive record data $\mathrm{q}_{n_i,n_j}(t)$; Interactions $\mathrm{r}_{i,j}^{t}$;
Output: Overall trust value $VO_{n_i,n_j}(t)$

Begin
if $(t > 0)$ **then**
  for $i = 1$ to m **do**
    for $j = 1$ to m **do**
    Calculate $VD_{n_i,n_j}(t)$ and feedback to the trust evaluation node e_k;
    end for
  end for
  calculate weight $\mu_{i,j} = \dfrac{\mathrm{r}_{i,j}^{t}}{\sum_{i=1}^{m} \mathrm{r}_{i,j}^{t}}$;
  evaluation node e_k calculat $VI_{\mathrm{n}_i,\ \mathrm{n}_j}(t)$ ion and feedback to n_i
  overall trust value $VO_{n_i,n_j}(t) = \eta VD_{n_i,n_j}(t) + (1-\eta)\, VI_{n_i,n_j}(t)$
  return $VO_{n_i,n_j}(t)$
end if
End

4.3 Overall Trust Evaluation

In this section, we comprehensively consider direct and indirect trust evaluation values to calculate the final overall trust. However, determining the relative importance of these two trust parameters is a key challenge. To solve this problem, we adopt an approach based on information entropy to determine their weights more objectively. The information entropy is calculated as follows:

$$H\left(VD_{n_i,n_j}(t)\right) = -VD_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\ln\left[VD_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right] - \left[1 - VD_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right]\ln\left[1 - VD_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right] \tag{18}$$

$$H\left(VI_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right) = -VI_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\ln\left[VI_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right] - \left[1 - VI_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right]\ln\left[1 - VI_{\mathrm{n}_i,\ \mathrm{n}_j}(t)\right] \tag{19}$$

The weight is calculated as follows:

$$\eta = \frac{H\left(VD_{n_i,n_j}(t)\right)}{H\left(VD_{n_i,n_j}(t)\right) + H\left(VI_{n_i,n_j}(t)\right)} \tag{20}$$

Then the overall trust evaluation value is:

$$VO_{n_i,n_j}(t) = \eta VD_{n_i,n_j}(t) + (1-\eta)\, VI_{n_i,n_j}(t) \tag{21}$$

The overall calculation process is shown in Algorithm 2.

5 Simulation and Analysis

This section verifies the effectiveness of the proposed trust model and trust evaluation mechanism based on autoencoder clustering (TEE) through simulation experiments.

5.1 Analysis of the Trust Evaluation Model

In this section, we simulate and test the performance of the credit model. We analyzed model performance at different node sizes. According to the actual network conditions, most nodes are trusted nodes, so we set the number of malicious nodes to 30%. We use two key metrics to evaluate the effectiveness of the authentication architecture based on trust assessment. Based on the average response time (RT) and malicious detection recall rate (Recall) to assist in evaluating the performance of the model.

RT reflects the system's ability to quickly adapt to changes and is very important for performance evaluation in IoT edge computing environments. In the experiment, we use different trust computing mechanisms, including DME (Distributed reputation management), RMD (Robust multi-dimensional trust mechanism), ART (A reliable trust mechanism), and the proposed TEE mechanism. By comparing the RTS of these mechanisms, we can verify which mechanism has faster computation speed and thus better ADAPTS to changes in the network environment. As shown in Fig. 3, TEE, ART, and DME mechanisms are significantly more capable of detecting malicious nodes than RMD mechanisms. This shows that the TEE mechanism is not only excellent in terms of computational efficiency, but also very powerful in detecting malicious behavior. According to the experimental results, the proposed TEE mechanism is significantly superior to ART, RMD, and DME mechanisms in Recall. TEE performs better in Recall as the number of devices increases.

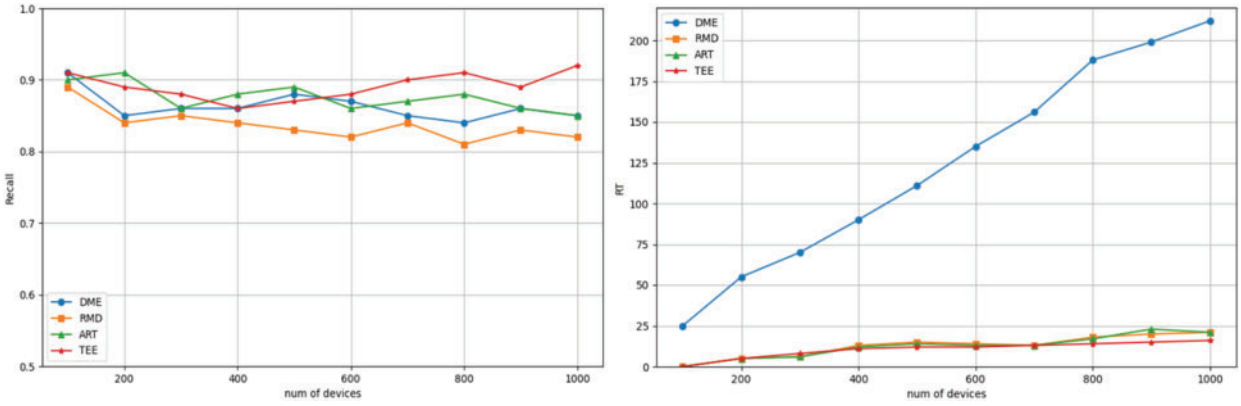

Figure 3: The value of Recall and RT varies with the number of devices

5.2 Analysis of Trusted Clustering Algorithm Based on Autoencoder

In our study, we found that the AEDC algorithm performs well in clustering tasks with different weight configurations. As illustrated in Fig. 4, under lower weights, such as 0.3 and 0.4, the AEDC algorithm exhibits superior performance compared to K-MEANS. This can be partially attributed to the adaptability of the AEDC algorithm, which enables it to better accommodate the characteristics of the data under varying weight settings. It excels in recognizing different features within the data, allowing it to capture subtle differences more effectively under lower-weight conditions, thus enhancing clustering accuracy. This adaptability is of paramount importance in numerous application domains, particularly in cases where the emphasis on different features may evolve, such as in recommendation systems.

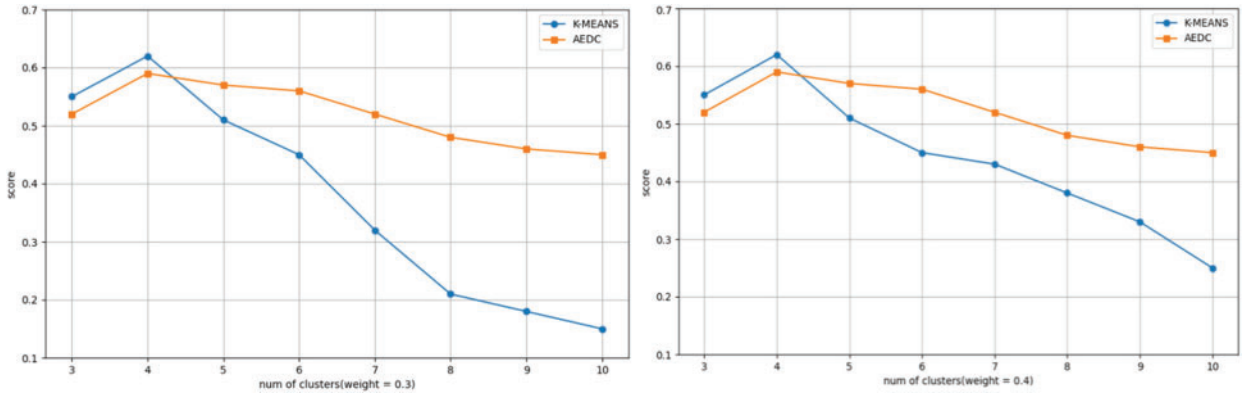

Figure 4: The score of K-MEANS and AEDC varies with the number of clusters at weights equal to 0.3 and 0.4

Furthermore, we investigated the impact of different weight settings on the performance of the AEDC algorithm and the K-MEANS algorithm. Our research demonstrates the stability of AEDC algorithm performance under higher weight values and its lower sensitivity to data features. As illustrated in Fig. 5, with the weights gradually increased from 0.3 to 0.6, we observed a convergence in the performance of both algorithms. This implies that the AEDC algorithm maintains outstanding performance in high-weight environments, with minimal interference from variations in data features. This is particularly exciting, as it suggests that the AEDC algorithm can consistently deliver exceptional performance across various application scenarios.

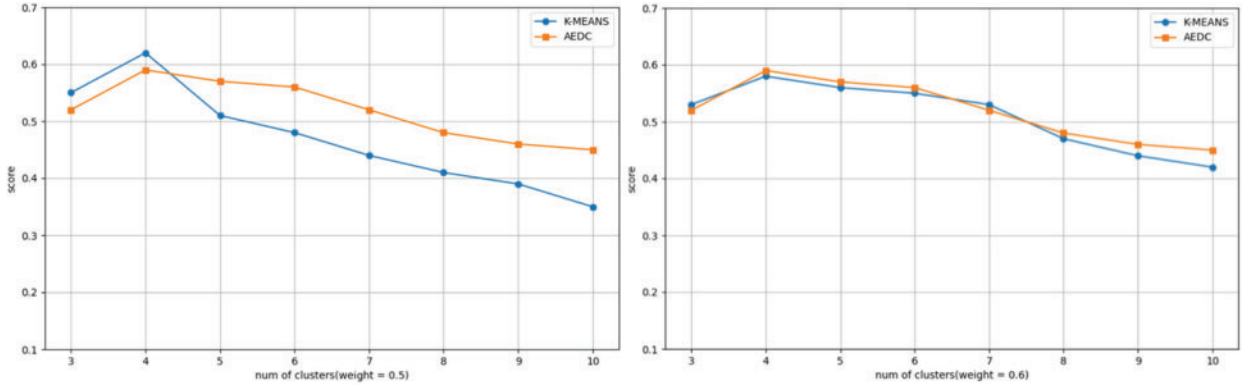

Figure 5: The score of K-MEANS and AEDC varies with the number of clusters at weights equal to 0.5 and 0.6

This phenomenon of gradual convergence in performance offers increased flexibility in algorithm selection for real-world applications. It allows us to choose suitable algorithms flexibly based on specific weight setting requirements without concerns about performance degradation. This has significant practical value in data mining tasks and cluster analysis, as the characteristics and demands of data may evolve continuously in practical applications.

6 Conclusion

In conclusion, this study introduces an innovative trust evaluation mechanism for accessing IoT edge devices, significantly enhancing system performance by integrating key elements such as cross-domain authentication, trust clustering, and adaptive weight considerations. Firstly, we successfully introduce a blockchain-based cross-domain authentication process, ensuring secure interoperability among devices in different domains, thereby enhancing system reliability. Secondly, we design a trust clustering algorithm based on autoencoder, improving trust assessment accuracy by comprehensively considering data, and effectively supporting system reliability and accuracy. Lastly, we introduce an adaptive weight calculation method, further enhancing system performance by comprehensively considering various factors affecting trust assessment. The evaluation mechanism provided in this article can be applied to actual IoT scenarios. Especially scenarios involving several elements such as the Internet of Things, blockchain, and trust assessment. Its limitation is that the blockchain limits the overall efficiency of the network and increases the network's operating costs. How to improve the processing efficiency of network tasks while ensuring high network reliability is a challenge.

Acknowledgement: Not applicable.

Funding Statement: This work is supported by the 2022 National Key Research and Development Plan "Security Protection Technology for Critical Information Infrastructure of Distribution Network" (2022YFB3105100).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Xiao Feng, Zheng Yuan; data collection: Zheng Yuan; analysis and interpretation of results: Xiao Feng, Zheng Yuan; draft manuscript preparation: Zheng Yuan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data not available due to legal restrictions. Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data is not available.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. Mohammadi, A. M. Rahmani, A. M. Darwesh and A. Sahafi, "Trust-based recommendation systems in the internet of things: A systematic literature review," *Human-Centric Computing and Information Sciences*, vol. 9, pp. 21, 2019.

[2] K. Mabodi, M. Yusefi, S. Zandiyan, L. Irankhah and R. Fotohi, "Multi-level trust-based intelligence schema for securing of Internet of Things (IoT) against security threats using cryptographic authentication," *Journal of Supercomputing*, vol. 76, pp. 7081–7106, 2020.

[3] W. S. Shi, H. Sun, J. Cao, Z. Quan and L. Wei, "Edge computing—An emerging computing model for the Internet of everything era," *Journal of Computer Research and Development*, vol. 54, no. 5, pp. 907–924, 2017.

[4] A. Altaf, H. Abbas, F. Iqbal and A. Derhab, "Trust models of internet of smart things: A survey, open issues, and future directions," *Journal of Network and Computer Applications*, vol. 137, pp. 93–111, 2019.

[5] L. Tong, Y. Li and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. of 35th Annu. IEEE Int. Conf. on Computer Communications*, San Francisco, CA, USA, pp. 1–9, 2016.

[6] P. Milan, Y. C. Hu, H. Patrice, J. Jerome, T. Chris *et al.,* "Mobile-edge computing–Introductory technical white paper," France: ETSI: Sophia Antipolis, vol. 29, pp. 854–864, 2014.
[7] W. S. Shi, X. Z. Zhang, Y. F. Wang and Q. Y. Zhang, "Edge computing: State-of-the-art and future directions," *Research and Development*, vol. 56, no. 1, pp. 69–89, 2019.
[8] D. Zhang, Y. Ma, C. H. Zheng, Y. Zhang, X. S. Hu *et al.,* "Cooperative-competitive task allocation in edge computing for delay-sensitive social sensing," in *Proc. of IEEE/ACM Symp. on Edge Computing*, Seattle, WA, USA, pp. 243–259, 2018.
[9] Z. H. Qu, B. L. Ye and G. H. Chen, "Research progress of resource optimization technology for edge computing," *Big Data*, vol. 5, no. 2, pp. 17–33, 2019.
[10] F. J. Mora-Gimeno, H. Mora-Mora, D. Marcos-Jorquera and B. Volckaert, "A secure multi-tier mobile edge computing model for data processing offloading based on degree of trust," *Sensors*, vol. 18, no. 10, pp. 3211, 2018.
[11] X. Xu, Z. Liu, 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.
[12] 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.
[13] J. Chi, E. Owusu, X. Yin, T. Yu, W. Chan *et al.,* "Privacy partition: A privacy-preserving framework for deep neural networks in edge networks," *IEEE/ACM Symp. on Edge Computing (SEC)*, vol. 2018, pp. 378–380, 2018.
[14] I. U. Din, A. Bano, K. A. Awan, A. Almogren, A. Altameem *et al.,* "LightTrust: Lightweight trust management for edge devices in the Industrial Internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 2776–2783, 2021.
[15] Z. Kuang, Z. Ma, Z. Li and X. Deng, "Cooperative computation offloading and resource allocation for delay minimization in mobile edge computing," *Journal of Systems Architecture*, vol. 118, pp. 102167, 2021.
[16] S. Jošilo and G. Dán, "Wireless and computing resource allocation for selfish computation offloading in edge computing," in *Proc. of IEEE INFOCOM Conf. on Computer Communications*, Paris, France, pp. 2467–2475, 2019.
[17] V. D. Valerio and F. L. Presti, "Optimal virtual machines allocation in mobile femto-cloud computing: An MDP approach," in *Proc. of IEEE Wireless Communications and Networking Conf.*, Istanbul, Turkey, pp. 7–11, 2014.
[18] Y. Ouyang, W. Liu, Q. Yang, X. Mao and F. Li, "Trust-based task offloading scheme in UAV-enhanced edge computing network," *Peer-to-Peer Networking and Applications*, vol. 14, no. 4, pp. 3268–3290, 2021.
[19] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, 2019.
[20] J. Yuan and X. Li, "A reliable and lightweight trust computing mechanism for IOT edge devices based on multi-source feedback information fusion," *IEEE Access*, vol. 6, pp. 23626–23638, 2018.
[21] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy *et al.,* "Mobile edge cloud system: Architectures, challenges, and approaches," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2495–2508, 2018.
[22] L. Yuan, Q. He, F. Chen, J. Zhang, L. Qi *et al.,* "CSEdge: Enabling collaborative edge storage for multi-access edge computing based on blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 8, pp. 1873–1887, 2022. https://doi.org/10.1109/TPDS.2021.3131680
[23] T. R. Gadekallu, Q. Pham, D. Nguyen, P. Reddy, N. Deepa *et al.,* "Blockchain for edge of things: Applications, opportunities, and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 964–988, 2022. https://doi.org/10.1109/JIOT.2021.3119639
[24] X. Zhu, Y. Luo, A. Liu, N. N. Xiong, M. Dong *et al.,* "A Deep reinforcement learning-based resource management game in vehicular edge computing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2422–2433, 2022. https://doi.org/10.1109/TITS.2021.3114295

[25] J. Yun, Y. Goh, W. Yoo and J. M. Chung, "5G Multi-RAT URLLC and eMBB dynamic task offloading with MEC resource allocation using distributed deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20733–20749, 2022. https://doi.org/10.1109/JIOT.2022.3177425

[26] I. Cvitić, D. Peraković, M. Periša and B. Gupta, "Ensemble machine learning approach for classification of IoT devices in smart home," *International Journal of Machine Learning and Cybernetics*, vol. 12, pp. 3179–3202, 2021. https://doi.org/10.1007/s13042-020-01241-0

[27] Y. Li, B. Wang, H. Wang, F. Ma, H. Ma *et al.,* "An effective node-to-edge interdependent network and vulnerability analysis for digital coupled power grids," *International Transactions on Electrical Energy Systems*, vol. 2022, pp. 1–13, 2022. https://doi.org/10.1155/2022/5820126