A Multi-property Method to Evaluate Trust of Edge Computing Based on Data Driven Capsule Network

Chenghan Jia, Kai Lin, Jing Deng

***Abstract*— As one of the computing paradigms in the process of traditional cloud computing turning to marginalization, edge computing is designed to meet the requirements of edge devices such as real-time response, security, privacy and so on. However, resources in edge computing are generally heterogeneous and dynamic, resulting in lack of trust among devices or a completely untrusted computing resources. Thus, novel and effective methods to evaluate the trust of edge devices in order to increasing the adoption of edge computing resources, have become an urgent issue. In this context the paper proposes a multi-property method to evaluate trust of edge computing, we establish an objective expression of trust property by considering which factors affect trust evaluation in resource request or service application process from the perspective of dynamic change, then the data-driven capsule network is utilized to analyze the correlation between trust properties and gives an objective prediction of trustworthy. The main contribution of this paper is the design of an effective trust evaluation method, which gives the objective reliability of edge devices also coordinates all levels to provide users with trustworthy application services.**

Edge computing, Trust evaluation, Data driven, Capsule network.

I. Introduction

Explosive growth of (IoT) devices makes traditional cloud computing not enough to support the storage and the process of such huge derived data. Several calculating models are proposed for this problem such as hybrid cloud computing [1], mobile cloud computing [2] and cloud-sea computing [3]. Edge computing as one of the methods has received tons of attention in providing the end-to-end services and sharing the computing task of the cloud. It often consists of a large number of terminals with limited resources, edge servers and cloud server, where the bottom layer is composed of many terminals with limited resource; edge servers with strong computing resources converged locally form the middle layer; top layer is data center servers in the cloud. In edge computing, the multi-level collaboration produces faster service response to meet the needs of real-time business, intelligent application and so on. However, due to the different capabilities of edge devices and the differences in the tasks performed, it's difficult to provide users with credible services in large scale dynamic edge computing environment. Therefore, how to implement secure and trusted edge computing becomes an urgent problem to be solved.

Chenghan Jia and Kai Lin are with the School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China (e-mail: midoricat@foxmail.com; link@dlut.edu.cn).

Deng is with the Department of Computer Science, University of North Carolina at Greensboro, Greensboro, NC 27412 USA (e-mail: jing.deng@uncg.edu).

Several research groups have reported attempts to build the trust mechanism of edge computing. Roman *et al.* [4] pointed out that trust management is an important part of security mechanism in edge computing environment from analyzing the security and privacy of user data. Soleymani *et al.* [5] proposed a fuzzy trust model based on experience and plausibility to secure the link between cloud and edge computing environment effectively. Umar *et al.* [6] built a credibility framework of the multi-access vehicular network that integrates various dimensions of trust and models the trustworthiness of the agents of other vehicles. Li *et al.* [7] proposed a localized credibility management model that protects the privacy information of users by evaluating the behavior trust of each node and its neighbors. Echeverria *et al.* [8] presented a method for building trusted identities in dishonest environment based on secure key generation and exchange, which realized effective point-to-point communication. Bennani *et al.* [9] pointed out that private cloud should be one of the models of cloud computing, and proposed a preventive/detective method to evaluate private cloud to select trusted public cloud services. Although the above methods have achieved the goal of building trust mechanisms or credibility management models, they cannot comprehensively consider the influence of resource interaction or application services on the trust evaluation from the perspective of edge device data.

In this paper, we first establish an edge computing architecture from the perspective of dynamic change based on the requirement that resources should be shared and coordinated according to scenario needs, where the trust framework of edge computing unit is constructed according to the system state or application request. Next, the trust property of edge computing are further subdivided based on the heterogeneity and dynamics, and the trust evaluation model is established correspondingly. Finally, the data-driven capsule network [10] use trust properties generated from historical data to predict trust property for the group of edge devices participating in specific computing task or resource interaction. The outcome of trust evaluation is aggregated feedback on current resource interaction or application request, and provides references for the coordination of subsequent resource.

II. Edge Computing Architecture

In this section, we first present an edge computing architecture based on application services and resource collaboration scenarios. Then, we discuss a trust evaluation method based on this architecture.

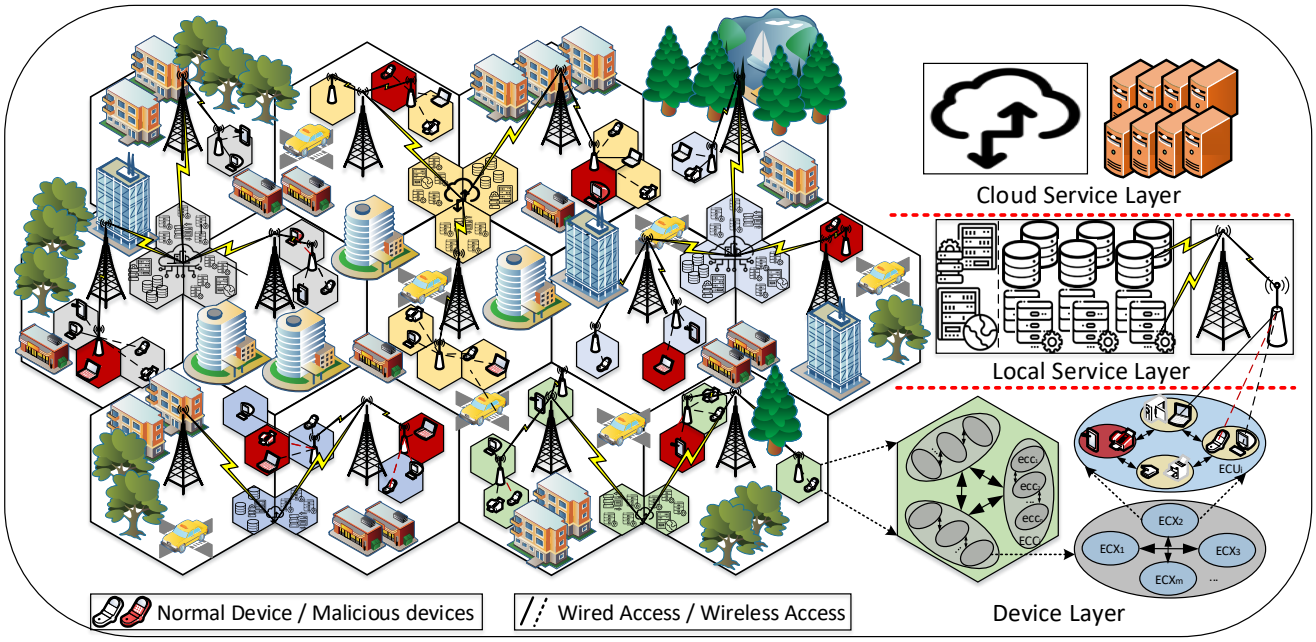


Fig. 1: Illustration of Edge Computing Architecture.

A. System Model

Figure 1 shows the architecture of edge computing based on application services or resource collaboration. Cloud server provides services for the underlying devices and distributes computing tasks, while edge devices execute local tasks when requesting services from above. This way has several advantages, such as reducing communication energy consumption, lowering service time, increasing network autonomy, and improving system scalability.

As shown in Fig. 1, the proposed architecture comprises three parts as follows: cloud service layer, local service layer and device layer.

1) Cloud Service Layer: Cloud service layer provides comprehensive application services and relies mainly on the server groups and primary database. It is considered as always reliable and available while risks involved are beyond the scope of this paper.

2) Local Service Layer: Local service layer is composed of locally converged edge servers with strong resource capability or dedicated servers. It calculates trust demand of tasks assigned to edge devices and selects appropriate computing resources. Besides, it monitors trust state of edge devices and authenticates the trustworthy of requested resources. Edge servers dedicated to trust evaluation are considered reliable.

3) Device Layer: Device layer consists of various edge devices. In the process of service request or resource interaction, participating devices communicate with cloud service layer or local service layer through multiple access ways. These devices are divided into different domains based on their common tasks or resource collaboration as shown in Fig. 1. Red points represent malicious devices participate in resource interaction. These points are distributed in normal devices, resulting in additional requirements of device trust evaluation.

The abbreviations used in Fig. 1 are given as follows:

Definition 1: Edge Computing Unit (ECU) is a resource unit executing a specific computing task or a primary application service, and we consider an edge device can be represented as an ECU. Each ECU consults only a number of other ECUs. This number is dynamically related to the size of the calculation task, which is always set to a small value that can be associated with other ecus. ECU_s is a set of ECU that meet a specific application requirement or a complete computing task at a time.

Definition 2: Edge Computing eXecutor (ECX) is a collection clustered by ECU_s that complete a whole specific computing task through trust evaluation. Each ECX satisfies a computing task or a service request.

Definition 3: Edge Computing Combo is a collection clustered by ECXs that match a computing task or a service request through trust filtering. It satisfies computing tasks or service requests in the current edge computing environment.

From the perspective of device clustering and resource interaction, this paper analyzes the data related to edge devices and evaluates the trust degree of devices from multiple aspects, then reduces risks and improves reliability of edge computing environment. More importantly, different from the single mode of trust evaluation, the proposed trust mechanism uses the device data from considering the dynamic trust change and gives the data-driven trust evaluation mechanism, which reduces the energy costs of edge devices and makes this method a lightweight scheme.

B. Trust Evaluation

In edge computing environment, resource constraint usually leads to huge differences in the number of request and

response to services. Some devices refuse to divulge identity information in privacy consideration but still seek interaction; some need high-level security mechanism because of real-time services; some only expect to get benefits from resource interaction and do not contribute any held resources, which are considered as potential dangers. These make it difficult to establish a targeted trust evaluation method. Besides, using single factor to evaluate trust is not reliable. For example, the authentication mechanism usually only has two results: pass or fail. Therefore, we need consider different trust metrics with multiple key properties to demonstrate the complexity of edge devices in performing resource interaction or application service.

The proposed trust evaluation method is based on Fig. 1. Supposing that there are several interactions between ECU$_s$(even though it might happen just once) clustered by common tasks in the past period of time, using capsule network to analyze the trust value of each ECU$_s$, the corresponding trust degree of ECX can be predicted. And so on, we can get the trust degree of the whole edge computing environment. The definition of trust property is described next. Detailed procedures for the trust evaluation process will be further discussed in Section III.

C. trust property

The trust property of ECU$_s$ is defined as follow:

$$\boldsymbol{V_t} =< V_i, V_b, V_c > \tag{1}$$

where V_i, V_b, V_c indicates identity trust, behavior trust and capability trust.

1) Identity Trust: Evaluating identity trust of device needs to consider privacy issues. For example, exposing a device's ID will leave it to the danger of identity theft, fake data generation, etc.. Public Key Infrastructure (PKI) as a reliable authentication mechanism which can easily realize one-to-one interaction in a distributed environment has been widely used in edge computing. Considering that identity trust is a part of the whole trust evaluation, this paper presents a key generation method with less computation by simplifying some contents of PKI as follows.

In the process of resource request, the credibility of identity of ECU$_s$ is defined as identity trust V_i. In order to calculate V_i, we consider identity trust of ECU$_s$ is guaranteed by an anonymous authentication method based on Elliptic Curve Cryptosystem (ECC) [11] which is used to calculate the key of each ECU$_s$. First, ECC gives a discrete elliptic curve E, then public key and private key are generated according to one point $O(x, y)$ on E, with the following two steps:

Step 1: Server Key Generation A key pair (K_p, K_s) are generated by the local edge server as follows:

$$\begin{aligned} K_p &= H(I_l, I_e) \\ K_s &= M(K_p) \end{aligned} \tag{2}$$

The public key K_p is calculated by hash function H according to the ID I_l of edge server and the ID I_e of ECU$_s$. Private key K_s is calculated by encoding function M according to K_p.

After K_p and K_s are calculated, local edge server sends identity information $D = \{K_p, E, O\}$ to ECU$_s$.

Step 2: ECU$_s$ Key Generation: Three keys K_{vi}, K_{ep} and K_{es} are calculated after the ECU$_s$ receives D as follows:

$$\begin{aligned} K_{vi} &= R_1(O(x, y)) \\ K_{ep} &= R_2(O(x, y)) \\ K_{es} &= R_1(M(K_{ep})) \end{aligned} \tag{3}$$

where virtual key K_{vi} is used to protect the real ID of ECU$_s$, K_{ep} and K_{es} are public key and private key of ECU$_s$, and $R_1()$ and $R_2()$ are pseudo-random number generator (PRNG) functions.

Then ECU$_s$ sends a message $F = \{ID_e, K_v, K_{ep}, a\}$ to the local edge server, a is a nonce number. In the later authentication stage, edge server and ECU$_s$ use the public key to contact each other and use the private key to complete the identity matching. If the ECU$_s$ passes the authentication, V_i is set to 1; otherwise V_i = 0.

D. Behavior Trust

Considering the process of service request and response, we define the behavior trust of ECU$_s$ as V_b as a weighted sum of three terms:

$$V_b = w_{bc}V_{bc} + w_{be}V_{be} + w_{br}V_{br} \tag{4}$$

where V_{bc} represents behavior constraint, V_{be} represents behavior experience, and V_{br} represents rate of behavior change. $w_{bc}+w_{be}+w_{br} = 1$ are weight coefficients and can be changed according to different specific tasks.

1) Behavior Constraint: Different ECU$_s$ observes different n of behavior constraints or interaction specifications which is determined by specific tasks:

$$V_{bc} = w_1\xi_1 + w_2\xi_2 + ... + w_c\xi_c \tag{5}$$

where ξ_i is the constraint or specification that the ECU$_s$ follows, $\sum_{i=1}^{c} w_i = 1$ are the weight coefficients.

2) Behavior Experience: Since the edge computing environment is changing constantly, it is necessary to consider the change of trust dynamically. Define the ECU$_s$ to be evaluated is recorded as ECU$_c$. For interactive behaviors of ECU$_c$ with several ECU$_s$, behavior experience V_{bh} reflects the interactive credibility of the recent past, where minimal interaction can also lead to effective trust building.

Suppose ECU$_s$ has been interacting with k other ECU$_s$, namely ECU$_1, \cdots,$ ECU$_k$. Let

$$s_j(\triangle t) = \{\tau_j^1, \tau_j^2, ..., \tau_j^s, ..., \tau_j^t\} \tag{6}$$

where $0 \leq \tau_j^s \leq 1$ is the interaction observation during time unit s between itself and ECU$_j$.

For ECU_j, assign it a threshold $\alpha_j \in (0, 1)$ according to the size of interactive resources. Define

$$H_j^+(\triangle t) = \sum_{i=1}^{t} \mathbf{1}(\tau_{\mathbf{j}}^{\mathbf{i}} \geq \alpha_{\mathbf{j}}) \tag{7}$$

where $\mathbf{1}(.)$ function returns 1 if the condition is true and 0 otherwise.

V_{be} can be defined as

$$V_{be} = \frac{1}{k}\sum_{j=1}^{k} \frac{H_j^+(\triangle t)}{t} \tag{8}$$

3) Rate of Behavior Change: Behavior changes over time, so the rate of behavior change is defined as V_{br} which reflects the actual change of behavior trust. There are

$$V_{br} = \frac{1}{k}\sum_{j=1}^{k}\left(\frac{(1-\lambda)V_{be\triangle(t-1)}}{1+\sqrt{t-H_j^+(\triangle(t-1))}} + \frac{\lambda V_{be\triangle(t)}}{1+\sqrt{H_j^+(\triangle t)}}\right) \quad (9)$$

for the last time series $\triangle(t-1)$, mainly consider $t - H_j^+(\triangle(t-1))$ of ECU_j with ECU_c. For the time series $\triangle(t)$, mainly consider $H_j^+(\triangle t)$ of ECU_j with ECU_c. λ is primary exponential smoothing factor.

ECU_s usually does not retain too much historical data because of the limited resources, so two time series $\triangle(t-1)$ and $\triangle(t)$ are selected as the basis of the rate of behavior change. We consider historical data closer to the current time as more indicative, so more weight is given to recent data $\triangle(t)$.

E. Capability Trust

Capability trust is defined as the degree of credibility to the capability property of device which includes accessibility, available bandwidth, response time and so on.

$L = \{e_1, e_2, ..., e_n\}$ is a set of ECU represents an ECU_s. $C = \{c_1, c_2, ..., c_m\}$ is the capability property set of ECU, where m represents the number of capability property related to the ECU. The value of m of any ECU under an ECU_s is identical. $C_s = \{c_{s_1}, c_{s_2}, ..., c_{sm}\}$ denotes that for any $c_{sk} \in C_s$ requires a corresponding $c_k \in C$, where c_{sk} is the standard value preset according to task requirements. $V_s =< C_s^1, C_s^2, ..., C_s^n >$ notes a vector of the ECU_s which consists of the C_s of the ECU, and $V =< C^1, C^2, ..., C^n >$ is a vector where C represents the actual C_s when ECU is working.

For V_s and V, ρ is defined as their correlation coefficient. We argue that the value of ρ denotes the capability trust of ECU_s, which can express the value of V_c. ω_v^T and $\omega_{v_s}^T$ as coefficient vectors.

Canonical correlation analysis (CCA) [12] algorithm is utilized to calculate ρ as follows:

$$\rho(V, V_s) = \frac{cov(\omega_v^T V, \omega_{v_s}^T V_s)}{\sqrt{var(\omega_v^T V) var(\omega_{v_s}^T V_s)}}$$
$$s.t. \begin{cases} \omega_v^T V = \omega_v^1 C^1 + \omega_v^2 C^2 + ... + \omega_v^n C^n \\ \omega_{v_s}^T V_s = \omega_{v_s}^1 C_s^1 + \omega_{v_s}^2 C_s^2 + ... + \omega_{v_s}^n C_s^n \end{cases} \quad (10)$$

where $\omega_v^T V$ and $\omega_{v_s}^T V_s$ are the linear representations of V and V_s, $cov(\omega_v^T V, \omega_{v_s}^T V_s)$ represents covariance between $\omega_v^T V$ and $\omega_{v_s}^T V_s$ while $\sqrt{var(\omega_v^T V) var(\omega_{v_s}^T V_s)}$ is their standard deviation multiplication.

III. TRUST EVALUATION METHOD BASED ON CAPSULE NEURAL NETWORK

It is unrealistic to evaluate trust using the data of all relevant properties of devices in heterogeneous and dynamic edge computing environment. Considering the processing of edge data, we need to consider not only the availability and timeliness of data, but also the value of existing information and establish a comprehensive edge trust mechanism. Therefore, we propose a multi Trust Property mechanism based on data-driven Capsule Neural network (TPCN), which uses the artificial intelligence algorithm with strong learning ability to achieve accurate and objective trust evaluation.

A. TPCN Method

TPCN aims to analyze $\boldsymbol{V}_t$ generated by historical data of the bottom ECU_s and predict the trustworthy of top level such as ECX. In this section, trust prediction is decomposed into three processes: capsule generation, convolution calculation, prediction results. Capsule generation generates corresponding convolution kernel for each ECU according to the specific calculation task and information related to the edge device. Convolution computation means that the trust property $\boldsymbol{V}_t$ and convolution kernel are used for multi-layer convolution computation to get the vector needed for subsequent computation. Prediction results show that the most reliable vector is selected from the output vector of the highest level according to the correlation analysis between multi-layer capsules. The process of the TPCN method is shown in Fig. 2.

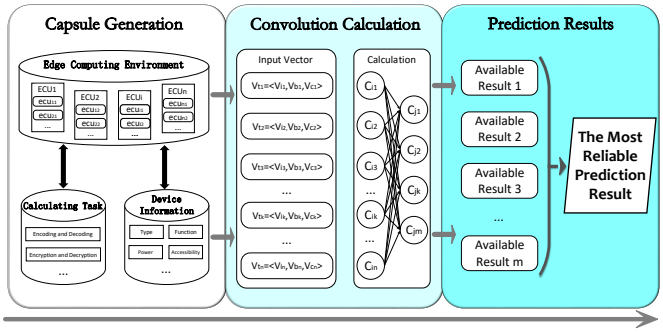


Fig. 2: The process of the TPCN method.

Because of sensitive to small change of data and capsules with various feature information, TPCN requires a shallow network structure and low data training, which is suitable for processing multimodal and small sample data in dynamic edge computing environment. In TPCN, dynamic routing algorithm [10] is mainly used to activate high-level capsules, and multiple iterations of routing will make prediction more accurate. The specific description of TPCN is as follows.

B. Description of TPCN

$\boldsymbol{V}$ is a matrix of the vector $\boldsymbol{V}_{ti}$ corresponding to ECU_i. $\boldsymbol{V}$ is input to TPCN to predict the trustworthy of ECX clustered by ECU_i, then, we give the trust relationship definitions that involved in the trust evaluation.

Definition 4: $\boldsymbol{T_i}$ is the unstandardized input vector obtained from the output vector of the lower capsule layer, and the length of $\boldsymbol{T_i}$ represents the degree of association between high-level capsules and low-level capsules. $\boldsymbol{T_i}$ is given as follow:

$$\widehat{\boldsymbol{O_{j|i}}} = w_{ij}\boldsymbol{O_i}, \qquad \boldsymbol{T_i} = \sum c_{ij}\widehat{\boldsymbol{O_{j|i}}} \quad (11)$$

where $\boldsymbol{O_i}$ is the output vector calculated from the lower input vector and the corresponding capsule. w_{ij} is a weight coefficient that indicates the position of lower capsule c_i and

upper capsule c_j related to the whole capsule network, and W_{ij} can be obtained by dynamic routing algorithm. $\widehat{\boldsymbol{O_{j|i}}}$ represents the predicted output vector from c_i to c_j. c_{ij} is defined as follows

Definition 5: c_{ij} is the coupling coefficient between c_i and c_j. In the TPCN method, a capsule usually represents a convolution kernel that is set by the amount of resources or tasks which positively correlates to the complexity of convolution kernel. In the capsule network, there are more features at the bottom and fewer features at the top, so the number of capsules in the current layer is more than the next layer ($i > j$). In addition, according to the characteristics of the capsule network, generally no more than four layers of network can meet the stop condition. The value of c_{ij} is given by dynamic routing algorithm.

Definition 6: $\boldsymbol{S_i}$ is the normalized vector of the $\boldsymbol{T_i}$ which is sent to upper capsule for convolution calculation. $\boldsymbol{S_i}$ is given as follow:

$$\boldsymbol{S}_i = \frac{||\boldsymbol{T}_i||^2}{1+||\boldsymbol{T}_i||^2} \frac{\boldsymbol{T}_i}{||\boldsymbol{T}_i||} \tag{12}$$

where $\boldsymbol{T_i}$ with longer length will be enlarge in the process of normalization, which can make more accurate predictions with less training data. $\boldsymbol{S_i}$ of the top capsule with length closer to 1 is selected as the result of trust of ECX. When two $\boldsymbol{S_i}$ have the same length, the angle size will be further compared and selected as the safety potential size.

Algorithm 1 computation steps

Require:

$\boldsymbol{V} =< \boldsymbol{V}_{t1}, \boldsymbol{V}_{t2}, ..., \boldsymbol{V}_{tn} >$: the trust property matrix
c_i: corresponding convolution kernel for each ECU$_s$
c_j: appropriate number of convolution kernels by the task size of ECX
w_{ij} and c_{ij}: initial value set to 0
r: route iterative number set according to the number and size of c_i

1: **if** route iterative number $! = r$ **then**
2: **for** ($i = 1$ to n) **do**
3: convolution calculation$(\boldsymbol{V}_{ti}) \rightarrow \boldsymbol{O}_i$;
4: **for** ($j = 1$ to m) **do**
5: update w_{ij} and c_{ij} by dynamic routing algorithm;
6: **end for**
7: **end for** $\boldsymbol{T} =< \boldsymbol{T}_1, \boldsymbol{T}_2, ..., \boldsymbol{T}_n >$
8: **for** ($j = 1$ to m) **do**
9: convolution calculation$(\boldsymbol{T}_j) \rightarrow \boldsymbol{O}_j$;
10: **end for**
11: **end if** route iterative number $= r$

Ensure:

select the vector $\boldsymbol{O}_j$ with length closest to 1;

According to TPCN, capsules of the same level represent entities of the same type, while upper capsules correspond to features abstracted from entities. For example, if ECU$_s$ are entities of the same type, upper capsules can generate trust features of sets composed of ECU$_s$. For a set of ECU_s represented by ECX, TPCN can be used to predict its credibility. Assuming that Algorithm 1 is used to predict the credibility, then there are two layers in the capsule network: the lower layer contains n capsules, the upper layer contains m capsules. Algorithm 1 is shown below, which describes in detail the multi-attribute method based on the data-driven edge computing trust evaluation of the capsule network.

IV. EXPERIMENT-BASED ANALYSIS AND EVALUATION

This section first describes device configuration and experiment setting in the simulated edge computing environment, then we show the experimental results.

A. Parameters and experimental methods

We use EdgeXFoundry simulator to implement a specific edge computing environment to test and verify the effectiveness of the proposed TPCN scheme. The performance of TPCN method is evaluated through a comparison with distributed reputation evaluation (DRE) [13] and multi-source feedback information fusion (MSFIF) [14]. DRE uses real-time data to evaluate trust among devices then calculate the common trust value by the central authority (CA); the MSFIF scheme uses multi-source feedback mechanism to calculate the global trust that weights the feedback trust factor manually or subjectively. However, DRE emphasizes the importance of distributed data to trust evaluation but does ignores the type of specific data while MSFIF puts more emphasis on the value of feedback data.

TPCN, DRE and MSFIF can detect malicious edge devices then generate a local trust value, and these three are used to calculate the trust value of the device set clustered by resource interaction or application services in the experiment. Detection rate of malicious devices (DR) and calculation efficiency that is the time needed to get the trust value of all local sets in the current edge computing environment are used to evaluate them. In trust evaluation, the time taken for the network to reach stability is one of the important metrics. Therefore, global convergence time (GCT) [15] is also used to evaluate the computation efficiency of different schemes in the simulation environment.

In TPCN, the change of trust mainly depends on the completion of resource interaction or application request. We add some malicious edge devices to the simulation environment, where these devices will occupy resources or send application requests continuously. Due to limited space, we compare DR and GCT of different number of malicious devices when the total number of devices is fixed. Other comparison methods will be supplemented in further study.

B. Result evaluation

500 edge devices and 20 edge servers are randomly distributed in the environment, and we set the trust value range of all edge devices between 0 to 1(if the trust value is lower than the set standard value, the device is marked as dishonest). DR is given by calculating the ratio of the number of marked devices to the number of pre-set malicious devices.

GCT of the network with 10% malicious devices in different total devices is shown in Fig. 3. It can be seen that when the

number of malicious devices is fixed, TPCN achieves lower GCT than MSFIF and DRE.

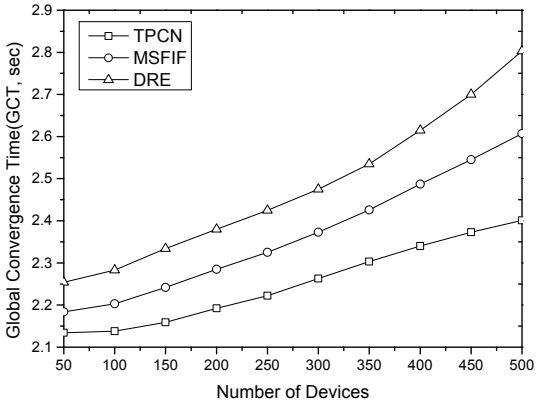


Fig. 3: GCT of TPCN, MSFIF and DRE.

Then we compared DR with 5%, 10%, 20% and 40% malicious devices in all devices and Fig. 4 shows the corresponding results. It can be seen that with the increase of malicious devices, DR of the three methods has declined, but TPCN is able to maintain a higher DR.

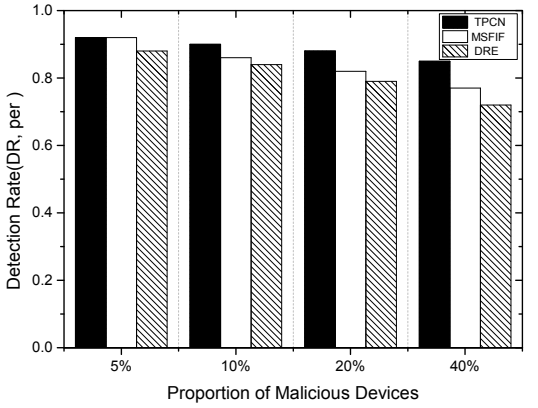


Fig. 4: DR comparison of TPCN, MSFIF and DRE.

The presented results validate that the method proposed in this paper has a higher calculation efficiency and detection rate of malicious devices than the other two, therefore providing references for the coordination of subsequent resource in a large-scale edge computing environment.

V. CONCLUSION

In this paper, trust evaluation in edge computing environment is analyzed and a TPCN method is proposed taking into account scenarios of resource interaction or application services. In TPCN, the trust property is divided into identity trust, behavior trust and capability trust based on computing tasks and interaction of edge devices. First, ECC and KPI are used to judge whether identity trust exists; we use interaction between edge devices to give definition of behavior trust and its calculation expression next; the capability trust is calculated then by the difference between capability related property of devices and its standard value. Finally, trust property is sent to a data-driven capsule network as the input vector to predict the trustworthy of the high level. The TPCN scheme is proved to have high calculation efficiency in giving local trust of edge computing environment.

The follow-up works will focus on three parts: trust property division, optimization of data-driven capsule network and comparison of more factors with other methods.

ACKNOWLEDGEMENTS

This work was supported by the Fundamental Research Funds for the Central Universities under Grant No. DUT19JC22; and the National Natural Science Foundation of China under Grant No. U1908214.

REFERENCES

[1] B. Singh, S. Dhawan, A. Arora, and A. Patail, "A view of cloud computing," *International Journal of Computers & Technology*, vol. 4, no. 2, pp. 50–58, 2013.

[2] H. Dinh, C. Lee, D. Niyato, and W. Ping, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications & Mobile Computing*, vol. 13, no. 18, pp. 1587–1611, 2013.

[3] Z. Xu, "Cloud-sea computing systems: Towards thousand-fold improvement in performance per watt for the coming zettabyte era," *Journal of Computer Science& Technology*, vol. 29, no. 2, pp. 177–181, 2014.

[4] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, no. 2, pp. 680–698, 2018.

[5] S. Soleymani, A. Abdullah, M. Zareei, M. Anisi, C. Vargas-Rosales, M. Khan, and S. Goudarzi, "A secure trust model based on fuzzy logic in vehicular ad hoc networks with fog computing," *IEEE Access*, vol. 5, pp. 15 619–15 629, 2017.

[6] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "A multifaceted approach to modeling agent trust for effective communication in the application of mobile ad hoc vehicular networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 3, pp. 407–420, 2011.

[7] Z. Li and C. T. Chigan, "On joint privacy and reputation assurance for vehicular ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2334–2344, 2014.

[8] S. Echeverria, K. Dan, K. Williams, and G. Lewis, "Establishing trusted identities in disconnected edge environments," *IEEE/ACM Symposium on Edge Computing (SEC)*, Washington, USA, 27-28, Oct. 2016.

[9] N. Bennani, K. Boukadi, and C. Ghediraguegan, "A trust management solution in the context of hybrid clouds," *2014 IEEE 23rd International WETICE Conference*, Parma, Italy, 23-25, Jun. 2014.

[10] S. Sabour, N. Frosst, and G. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc, 2017, pp. 3856–3866. [Online]. Available: http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf

[11] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology — CRYPTO '85 Proceedings*, H. C. Williams, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 417–426.

[12] D. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Computation*, vol. 16, no. 12, pp. 2639–2664, 2004.

[13] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25 408–25 420, 2017.

[14] 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. 23 626–23 638, 2018.

[15] X. Li, F. Zhou, and X. Yang, "Scalable feedback aggregating (SFA) overlay for large-scale P2P trust management," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 10, pp. 1944–1957, 2012.