Trustworthy Edge Storage Orchestration in Intelligent Transportation Systems Using Reinforcement Learning

Fuli Qiao, Jun Wu, Jianhua Li, Ali Kashif Bashir, Shahid Mumtaz, and Usman Tariq

***Abstract*—A large scale fast-growing data generated in intelligent transportation systems (ITS) has become a ponderous burden on the coordination of heterogeneous transportation networks, which makes the traditional cloud-centric storage architecture no longer satisfy new data analytics requirements. Meanwhile, the lack of storage trust between ITS devices and edge servers could lead to security risks in the data storage process. However, a unified data distributed storage architecture for ITS with intelligent management and trustworthiness is absent in the previous works. To address these challenges, this paper proposes a distributed trustworthy storage architecture with reinforcement learning in ITS, which also promotes edge services. We adopt an intelligent storage scheme to store data dynamically with reinforcement learning based on trustworthiness and popularity, which improves resource scheduling and storage space allocation. Besides, trapdoor hashing based identity authentication protocol is proposed to secure transportation network access. Due to the interaction between cooperative devices, our proposed trust evaluation mechanism is provided with extensibility in the various ITS. Simulation results demonstrate that our proposed distributed trustworthy storage architecture outperforms the compared ones in terms of trustworthiness and efficiency.**

***Index Terms*—Intelligent transportation systems, distributed storage architecture, trust evaluation, unified edge-cloud, reinforcement learning**

I. Introduction

WITH the advent of the Intelligent Transportation Systems (ITS), large-scale transportation network end-devices are interconnected to support traffic flow prediction, intelligent control technologies, and public transportation planning, etc., and thus generates massive amounts of data [1]. Data comes from diverse sources, such as floating car sensors, videos, GPS, smart cards, smart gird, and so on [2]. In this situation, data consumer and producer (prosumer) in ITS has more new requirements on the Quality of Service (QoS), especially real-time service [3]. On the other hand, the data generated by end-devices can be analyzed and processed to shape pragmatic knowledge information, which will play a more momentous role and accomplish more value in the future [4]. Due to the tremendous volume and application scenario complexity, it is challenging to store data at the transportation network edge using traditional methods [5]. Moreover, the majority of ITS devices are resource-constrained, and currently highly depend on cloud computing storage architecture, which risks a single point of failure.

In the ITS cloud-centric storage architecture, there are scalable comprehensive and systematic infrastructures, such as Apache Hadoop, which implements a distributed file system that supports processing and storage for a huge amount of data [6]. There are various researches required to promote the paradigm of ITS edge storage architecture [7]. However, existing researches on data storage at the edge mostly focus on optimizing and polishing the computation offloading solution based on the existing storage architecture [8]. The lack of a unified distributed storage architecture leads to obstacles of data sharing in different transportation networks [9], [10]. With edge computing, we envision that the role of transportation network edge nodes is shifting from a data consumer to a data producer as well as a consumer. Therefore, it is an urgent problem to keep the communication between edge nodes unhindered and not restricted by different regions and regulations in ITS.

Edge nodes in ITS can be deployed by rational third parties, which are vulnerable to security risks and attacks, including external and internal attacks [11]–[15]. This situation leads to the lack of trust between data prosumers and edge servers, which has hindered data secure storage [16]. Trust evaluation mechanism is currently regarded as a guard of distributed applications [17]. Different from the traditional authentication mechanism, it determines the trust levels of data prosumers and edge servers while provides dynamic behavior perceiving capability. As a supplemental technology, the trust evaluation mechanism makes security services more trustworthy by ensuring that data prosumers and edge servers are trustworthy during data storage and request process. Some previous trust works about trust evaluation have been applied in the wireless networks, but most of them do not focus on the ITS edge scenarios [18], [19].

Therefore, in this paper, we originally propose a distributed trustworthy edge intelligent storage scheme in ITS. In our proposed storage scheme, each edge server is modeled as a multi-functional agent, which is committed to efficiently

This work was supported in part by the National Natural Science Foundation of China under Grant 61972255.(*Corresponding author: Jun Wu.*)

Fuli Qiao, Jun Wu, Jianhua Li are with Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China. (e-mail: junwuhn@sjtu.edu.cn)

Ali Kashif Bashir is with Department of Computing and Mathematics, Manchester Metropolitan University, UK, and School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad (NUST), Islamabad, Pakistan.

Shahid Mumtaz is with Instituto de Telecomunicações (IT), Portugal.

Usman Tariq is with College of Computer Science and Engineering Prince Sattam bin Abdulaziz University, Saudi Arabia.

managing fragmented data storage. Our proposed dynamic storage mechanism ensures that highly popular data comes from trustworthy sources. The key contributions of our work are as follows:

- Firstly, a unified trustworthy edge storage architecture committed to managing data intelligently in ITS is proposed. The peculiarity of our proposed architecture is to migrate the virtues of Hadoop Distributed File System (HDFS) in Cloud Computing to Edge Computing to ensure that ITS services satisfy better QoS.
- Secondly, we propose a dynamic storage policy-making mechanism based on reinforcement learning to maximize the capabilities of edge nodes. It can recommend data with high call popularity for ITS services and update data accordingly. In order to achieve high-throughout links, we also propose a communication model for edge-cloud and edge-edge. In this case, edge nodes share information with neighbors through synchronous communication.
- Thirdly, in order to guarantee the security of data storage, we propose a federated trusted evaluation model to clarify the trustworthiness of edge servers and data prosumers in ITS, so as to judge whether data source and storage location are secure. We adopt the dual dimensions of direct and indirect trust to evaluate the trustworthiness of individuals, and indirect trust is related to recommendation mechanisms.

The rest of this paper is organized as follows. Section II shows the related works about related current works on AI in ITS, edge storage, trust mechanisms, and Hadoop/HDFS applications which we investigate. Section III provides an overview and model details of our proposed scheme. Then, our federated trust evaluation scheme is given in Section IV. The specific mathematical models and algorithms for scheduling the proposed scheme are illustrated in Section V. Simulation results are provided in Section VI. Eventually, the paper is concluded in Section VII.

II. Related Works

Most of the existing research works on AI in ITS mainly focus on the optimization of autonomous vehicles and traffic flow prediction [20]. A deep architecture for traffic flow prediction that incorporate a deep belief network at the bottom and a multitask regression layer at the top is proposed [21]. Based on their deep architecture, a group method is proposed to make multitask learning more effective to predict traffic flow density. In [22], the authors proposed a general active-learning framework for robust on-road vehicle recognition and tracking. Particle filter tracking is integrated to improve this system, and the system can be evaluated on real-world data set. Reinforcement learning has been widely used in traffic signal control, such as in multi-intersection vehicular networks, and a novel use of a multi-agent intelligent system is proposed to obtain an efficient traffic signal control strategy [23]. In order to find suitable signal timing policies, the authors proposed a deep neural network to learn the Q-function of reinforcement learning from traffic system inputs and outputs [24].

The compution offloading solutions based on known storage architectures are currently the maintstream of edge storage works. An approximation algorithm is proposed to achieve caching load balance based on an integer linear programming problem, which considers fairness metrics [25]. Besides, in order to minimize the latency of task implementation and the cost of the entire operation meanwhile maximize node utilization rate of local information and system trustworthiness, a multiplier cooperative storage algorithm based on alternating directions is proposed [26]. An optimal auction mechanism inclined to the service providers is proposed, aiming to diminish information asymmetry between users and service providers [27]. It also devises a computationally efficient method to evaluate user payments and the optimal cache space allocation.

Trust evaluation mechanism is widely applied in various distributed environments, such as mobile social networks, wireless sensor network, industrial IoT and so on. To comprehensively evaluate the trustworthiness of sensor nodes in the IoT system, an mobile edge computing based intelligent trust evaluation strategy using probabilitic graphical model is proposed which concerns on the data collection and communication behavior of sensor nodes [28]. In social networks, in order to supply secure multimedia contents retrieval schemes while preventing privacy breaches from honest-but-curious edge nodes and users, a trust evaluation mechanism is proposed to calculate the trustworthiness of edge nodes [29]. Considering existing methods neglect cross-domain communications and trust management in the IoT, a Holistic Cross-domain trust management model based on multilevel central authorities is aimed at providing multilevel security for service-centric IoT [30]. However, these works are applied in the specific scenarios, rather than a pervasive edge computing environment.

The proposal of Hadoop permits data distributed storage, while HDFS, the bottom of Hadoop, is not appropriate for use at the network edge. The nodes in the HDFS are divided into two types, the NameNode that provides metadata services and the DataNode that provides storgae blocks. HDFS has a flaw which is a single point of failure because there is only one NameNode. The performance of the Hadoop benchmark suite is investigated, which runs on the edge computing testing platform of physical and simulative infrastructure [31]. In addition, the work on MapReduce currently has a large scale which concentrates on compressing and analyzing data. In contrast to cloud computing, edge computing is short of the researches on data storage architecture.

Thus, this paper is committed to proposing a trustworthy edge storage architecture with reinforcement learning in ITS, which evaluates the trustworthiness of edge nodes for storage security and is a unified comprehensively scheme to promote ITS development.

III. Proposed Unified Trustworthy Edge Storage Architecture in ITS

Based on the functions of the edge nodes in ITS, the proposed storage architecture separates edge nodes into two categories which are data prosumers and edge servers. Each edge server consists of a Master and a few of data containers. The Master is similar to smart agents with control and

management functions, and the data container performs the storage allocation and recording tasks assigned by its Master. The Master has three functional modules, synchronization communication management, dynamic storage policy, and multi-user data writing and mapping. The most distinguishing characteristic is the dynamic storage to meet the requirements of ITS scenarios. We use distributed intelligent recommendation algorithms to determine the appropriate storage location for data by learning data classification and marking. In the communication management module, it is desirable for the Master to process requests and receipts, thereby facilitating data mobility between edge servers. To enhance the adaptability and scalability of the edge servers in ITS, we establish a communication protocol pool. Liken HDFS, data mapping tables can facilitate data lookups. However, our scheme supports arbitrary modification and deletion of data, and a file can have multiple cooperative writers. In addition, the data container immediately reports its remaining storage space to the Master. The data container stores edge segmented data and user private data, such as the ID number of vehicle drivers in some situations, and the public shared data that was previously put in the cloud. The popularity of data is a factor that affects where data is stored and is, therefore, a momentous component of data tags.

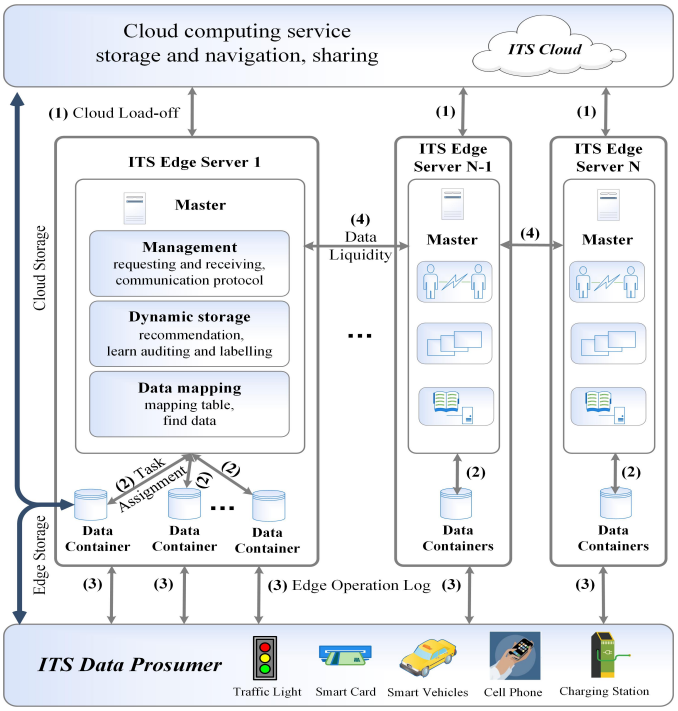

Fig. 1. Unified trustworthy edge storage intelligent architecture in ITS

The workflow of the proposed storage architecture is shown in Figure 1 which achieves a cloud-edge collaborative mechanism, based on our previous work [32]. Data prosumers can perform data pre-classification and preprocess assignments for better ITS services. The initial popularity of the data is recorded by data prosumers and initialized based on historical information. Besides, data prosumers make the data more valuable, and it's easy to analyze ITS data for some tasks, such as autonomous vehicles. For data prosumers, data uploads and downloads are basic functions. In this paper, we have enhanced the characteristics of the edge servers. Our storage architecture is more concerned with the interaction between edge servers, while edge servers can also store data sent by the cloud. The stored data is dynamic and circulated through the edge servers. For the cloud, it can audit messages and provide computing services and data storage for data sharing and message sharing.

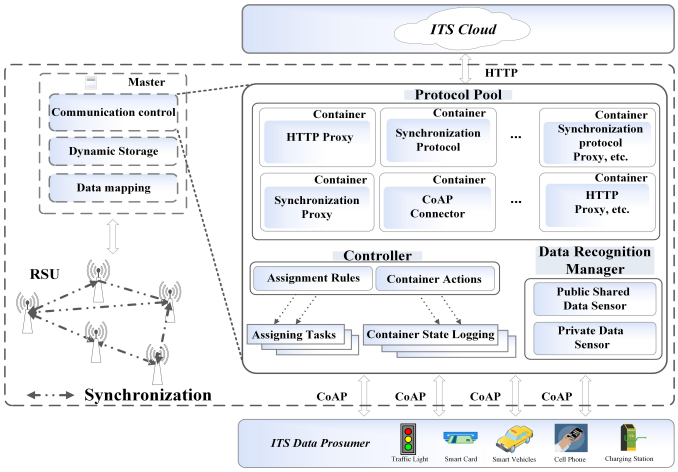

Fig. 2. Synchronization communication and data liquidity

A. Synchronization communication and data liquidity

As shown in Figure 1, Master has the following four main communication processes as follows: (1) Cloud Load-off: require the Master to communicate with the cloud to record public shared data, (2) Task Assignment: assign storage tasks by controlling the data container, (3) Edge Operation Log: record data upload behavior of the data prosumer, (4) Data Liquidity: an edge server and its counterparts exchange storage information distribution.

The communication management domain involves a protocol pool, as shown in Figure 2, which contains HTTP proxy, CoAP connector [33], and synchronization protocol proxy. It is responsible for interaction with the cloud, communication with the prosumers, and collaboration between the edge servers. Data recognition manager is arranged in this module, mainly clarifying public shared data which was previously stored in the cloud and private data which cannot be shared across the cloud. The communication controller monitors the actions of data containers and records the states of data containers. Depending on the required assignment rules, a Master has the capacity to assign tasks to the containers within its jurisdiction. In order to keep the information obtained by the edge servers in the same neighborhood consistent, a synchronization protocol is required for the communication, which cannot cause misunderstandings due to inconsistent information. On the other hand, this prominent function also facilitates data lookups.

B. Dynamic data storage policy based on trust value and Reinforcement Learning

The dynamic storage module in Master guarantees the real-time characteristics of edge computing. Firstly, the Master

identifies the data popularity on the data tag, with Q-learning [34] based on popularity to determine the data optimal storage location. It ensures that frequently used data can be stored in the edge server, and data that is not frequently used and not user privacy can be stored in the cloud. This process is committed to reducing the time for data requesting and the cost of resources consumed. Here, the Master updates the data label according to scalable-recommendation algorithms (e.g. knowledge-based recommendation and user preference recommendation). Secondly, high-trust edge servers are prioritized for storing data, while edge servers are more likely to store data submitted by high-trust prosumers. Thirdly, by checking the timestamp of the data, the Master determines whether expires and discards the expired data in time to increase the available space of the data containers. Fourthly, when the data container reaches its own storage capacity threshold, it is reported to the Master which manages it, and the Master would remove its redundant storage contents until there is additional space to assign tasks again.

C. Multi-user parallel mapping and trustworthy storage

The scheme we proposed is different from Hadoop in that the storage information of the data can be exchanged between several neighboring edge servers in the vicinity. Therefore, multiple edge servers can manipulate the same data at the same time, which means support for multi-user writes, modifies, and deletes. When the trust values of all the prosumers that manipulate the same data are the same, all the prosumers have the same priority. At this time, according to the time sequence in which the edge server receives the commands, the prosumers sequentially manipulate the data. On the other hand, when the trust values of the prosumers are different, the prosumer whose trust value is lower than the threshold has no permission to modify the data, and the prosumers would modify the data with reference to the trust value from high to low. The storage information mapping table needs to be more efficient, mainly used to record the data storage location, which also records data type, resource, and size. There are two kinds of stored data. If the stored data is file-based while the data containers in the edge server closest to it do not have enough storage space, it is cut by the Master and distributively stored in the vicinity of the edge server. Besides, the mapping table needs to record the cut block order of the cut file to facilitate the prosumer request.

IV. ITS Identity Authentication and Federated Trust Evaluation Scheme

To ensure storage security, we propose identity authentication and trust evaluation for ITS entities. Firstly, in order to authenticate the identity of both communicating parties, the identity authentication protocol based on trapdoor hashing is proposed which is divided into two phases: i) storage initialization phase; ii) storage authentication phase. Secondly, in order to establish a trustworthy interactive storage relationship, the federated trust evaluation mechanism of edge servers and data prosumers is proposed, which evaluates trust values for edge entities in geographically separated different network domains [35].

In our work, a trapdoor hash function whose collision resistance depends on the state knowledge of the user is proposed in our authentication scheme. Each trapdoor hash function consists of a pair public key and private key, represented as the hash key HK and the trapdoor key TK, respectively. Table I describes the important notations of this paper.

TABLE I
List of important notations

Term	Description
Z_q	Multiplicative group of integers modulo q
ID_{pi}	The identity of data prosumer i
G	A group of prime order q
$H(\cdot)$	Hash function
NHK_A	New trapdoor key computed by edge server A
S_B	Private key of edge server B
MA_B	Mutual authentication value of edge server B
SD_{ij}^k	Rate of stored size of edge server in k^{th} communication
$TL_i(0)$	Initial lifespan of edge server i
τ_k	k^{th} communication timestamp
Sq_{ij}^k	Storage service quality of i for P_{ji}
PE_{ij}^{τ}	Positive feedback
$K_{ii'}$	Knowledge similarity
$TP_{jj'}$	Trust value between data prosumers
$CR_{jj'}$	Credibility degree of data prosumer
AS_{ij}	Rate of data prosumer active state
Rd_{ij}	Rate of data quantity requested by data prosumer
Rc_{ij}	Rate of data prosumer communication success
CP_j	Credibility of data prosumer j
$DP_{d_{ji}}(0)$	Initial popularity of data d_{ji}
$a_s(t_s)$	Storage action
$N_{d_{ji}}$	Number of requests for d_{ji}
$s_s(t_s)$	State of edge server at slot t_s
λ_j	Request rate of data prosumer P_{ji}
d_i	Size of data requested by P_{ji}
pd_i	Size of data produced by P_{ji}
v_i	Pre-processing speed of P_{ji}
tv_j	Speed of checking the trust value of P_{ji}
vm_j	Speed of searching data for Mater y_j
t_{MP}	Delay of transmission between y_j and P_{ji}
t_{MC}	Delay of transmission between y_j and cloud
CP_j	Storage capacity of Master y_j
R_j^M	Processing latency of data tasks uploaded by P_{ji}
T_i	Processing latency of data tasks in P_{ji}

Lemma 1. [36] A trapdoor hash family consists of a pair $(\mathcal{I}, \mathcal{H})$ such that:

- $\mathcal{I}$ is a probabilistic polynomial-time key generation algorithm that on input 1^k outputs a pair (HK, TK), such that the sizes of HK, TK are polynomially related to k, where k is the size of the message.
- $\mathcal{H}$ is a family of randonmized hash functions. Every hash function in $\mathcal{H}$ is associated with a hash key HK, and is applied to a message from a space $\mathcal{M}$ and a random element from a finite space $\mathcal{R}$. The output of the hash function h_{HK} does not depend on TK.

Lemma 2. [37] A trapdoor hash family $(\mathcal{I}, \mathcal{H})$ has the following three properties:

- Efficiency: Given a hash key HK and a pair $(m, r) \in \mathcal{M} \times \mathcal{R}$, $h_{HK}(m, r)$ is computable in polynomial time.
- Collision resistance: There is no probabilistic polynomial-time algorithm $\mathcal{A}$ that on input HK outputs, with a probability that is not overlooked, two pairs $(m_1, r_1), (m_2, r_2) \in \mathcal{M} \times \mathcal{R}$ that satisfy $m_1 \neq m_2$ and $h_{HK}(m_1, r_1) = h_{HK}(m_2, r_2)$.
- Trapdoor collisions: There is a probabilistic polynomial time algorithm that given a pair $(HK, TK) \leftarrow \mathcal{I}(1^k)$, a pair $(m_1, r_1) \in \mathcal{M} \times \mathcal{R}$, and an additional message $m_2 \in \mathcal{M}$, outputs a value $r_2 \in \mathcal{R}$ such that: i) $h_{HK}(m_1, r_1) = h_{HK}(m_2, r_2)$; ii) If r_1 is uniformly distributed in $\mathcal{R}$ then the distribution of r_2 is computationally indistinguishable from uniform in $\mathcal{R}$.

A. Storage Initialization Phase

In the initialization phase, data prosumer i performs session protocol prior to key share and obtains the trapdoor key from edge server A after the successful authentication. The messages transmitted during the initialization phase are shown in Figure 3.

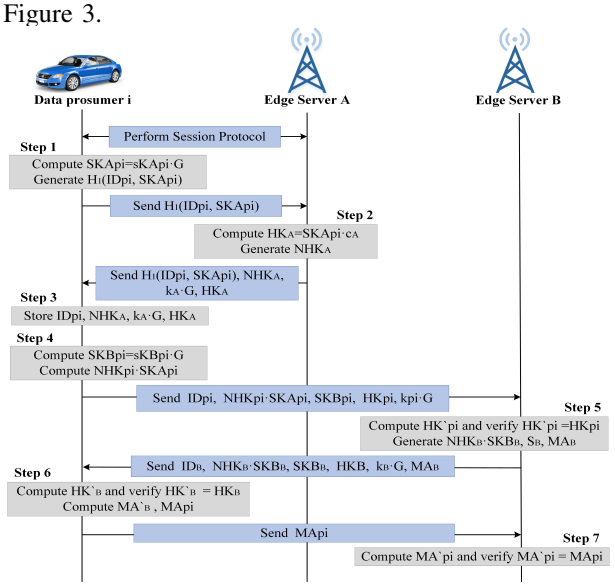

Fig. 3. Storage Initialization Phase

- **Step1**: Data prosumer i selects a random number $sKA_{pi} \in Z_q$ and computes SKA_{pi} by G which is a group of prime order q, where Z_q denotes the multiplicative group of integers modulo q. Then, data prosumer i computes message request as $H_1(ID_{pi}, SKA_{pi})$ and sends to edge server A via a secure channel. Here, ID_{pi} is the identity of data prosumer i and (sKA_{pi}, SKA_{pi}) is the long term key pair.
- **Step2**: After receiving $H_1(ID_{pi}, SKA_{pi})$, edge server A decrypts ID_{pi}. Then, it computes sKA_{pi}^{-1} and selects a random number $c_A \in Z_q$. Therefore, it can compute hash function as $HK_A = SKA_{pi} \cdot c_A$. Moreover, edge server A stochastically selects $k_A \in Z_q$ and computes the trapdoor key pair $(k_A \cdot sKA_{pi}^{-1}, k_A \cdot G)$. In addition, edge server A stores $k_{pi} \cdot G$. So edge server A computes the new trapdoor key NHK_A as

$$NHK_A = c_A - H_1(H_1(ID_{pi}, SKA_{pi}), k_A \cdot G) \cdot k_A \cdot sKA_{pi}^{-1}. \quad (1)$$

Then, it sends message response to the data prosumer i as $(H_1(ID_{pi}, SKA_{pi}), NHK_A, k_A \cdot G, HK_A)$.
- **Step3**: After receiving the response, data prosumer i verifies its identity and stores its parameters as $(ID_{pi}, NHK_A, k_{pi} \cdot G, HK_A)$.

B. Storage Authentication Phase

In the storage authentication phase, the mutual authentication process is initiated between data prosumer i and edge server B when edge server A has not enough storage space. The messages transmitted during the storage initialization phase are shown in Figure 3 and explained as follows.

- **Step4**: Data prosumer i obtains information of edge server B from edge server A. Then, data prosumer i selects a random number $sKB_{pi} \in Z_q$ and computes $sKB_{pi} \cdot G$. Also, data prosumer i computes $NHK_{pi} \cdot SKA_{pi}$ and sends $(ID_{pi}, NHK_{pi} \cdot SKA_{pi}, sKB_{pi} \cdot G, HK_{pi}, k_{pi} \cdot G)$ to edge server B.
- **Step5**: After receiving the authentication parameters from data prosumer i, edge server B selects a random number $c_B \in Z_q$ and computes HK'_{pi} of data prosumer i as [37]

$$HK'_{pi} = c_B \cdot SKA_{pi} \quad (2)$$

Further, edge server B compares HK'_{pi} with HK_{pi}. If HK'_{pi} is not equal to HK_{pi}, it will transfer a verification declined message to data prosumer i. Otherwise, it will select a random number $sK_B \in Z_q$ and compute its public key as $SKB_B = sK_B \cdot G$. Then, edge server B computes

$$NHK_B = c_B - H_1(H_1(ID_{pi}, SKA_{pi}), k_B \cdot G) \cdot k_B \cdot sKB_B^{-1} \quad (3)$$

and computes $NHK_B \cdot SKB_B$ for establishing the mutual authentication. Besides, edge server B generates its private key $S_B = H_2(sK_B \cdot (sKB_{pi} \cdot G), ID_B, ID_{pi})$ and the mutual authentication value $MA_B = H_3(k_B \cdot G, (NHK_B \cdot SKB_B) \cdot NHK_B, S_B)$. Then edge server B sends $(ID_B, NHK_B \cdot SKB_B, sK_B \cdot G, HK_B, k_B \cdot G, MA_B)$ to data prosumer i.
- **Step6**: Data prosumer i computes HK'_B of edge server B as [37]

$$HK'_B = c_B \cdot SKB_B \quad (4)$$

Further, data prosumer i compares and verifies HK'_B with HK_B. If HK'_B is not equal to HK_B, data prosumer i transfers a verification declined message to edge server B. Otherwise, data prosumer i generates it private key $S_{pi} = H_2(sKB_{pi} \cdot (sK_B \cdot G), ID_B, ID_{pi})$ and verifies edge server B by computing $MA'_B = H_3(k_B \cdot G, (NHK_B \cdot SKB_B) \cdot NHK_{pi}, S_{pi})$. If MA'_B is equal to MA_B, edge server B is authenticated by prosumer i, and data prosumer i sends $MA_{pi} = H_3(k_{pi} \cdot G, NHK_B \cdot (NHK_{pi} \cdot SKB_B), S_{pi})$ to data prosumer i. Otherwise, data prosumer i transfers the authentication failure message to edge server B.
- **Step7**: Edge server B receives mutual authentication value from data prosumer i and computes MA'_{pi}. Then edge server B verifies whether MA'_{pi} is equal to MA_{pi}.

If equal, data prosumer i is authenticated by edge server B. Otherwise, edge server B sends declined verification message.

C. Federated Trust Evaluation Scheme

Our proposed federated trust evaluation scheme consists of the trust values of edge servers and the trust values of data prosumers. We evaluate the trustworthiness of an edge server and a data prosumer by real numbers ET and PT, ranging from 0 to 1. The trust value ET is determined based on direct trust DET and indirect trust IET for balancing the effects of indirect trust and direct trust. Moreover, PT has the same derivation, and the trust relationship is as shown in the following Figure 4.

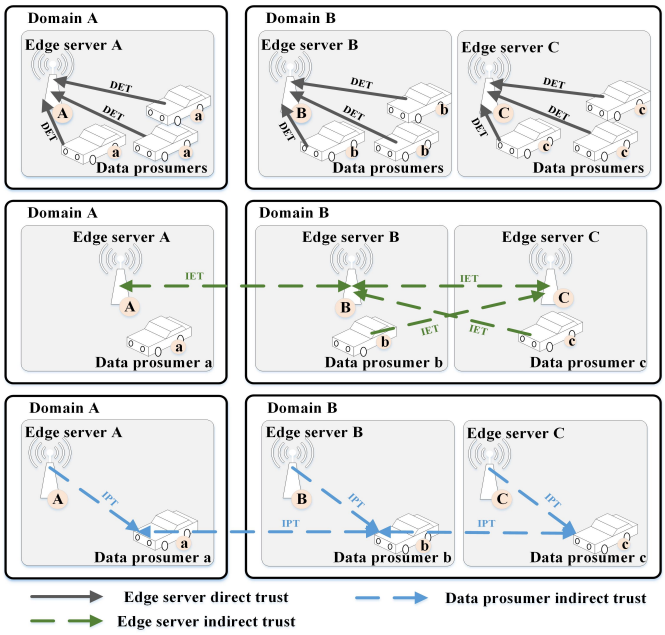

Fig. 4. ITS Federated Trust Evaluation Scheme

1) Edge Server Trust Value from Direct Trust: The direct trust of an edge server is based on the direct interactions between the edge server and the data prosumers, including the stored data scale of the edge server, the lifespan of the edge server, and the rating on the Secure Service Quality which is the feedback of data prosumers.

The first evaluation indicator is the stored data scale of the edge server E_i for data prosumer P_{ji} during a cycle. The rating of the service with the large storage size has an obvious effect on the derivation of direct trust. Let sd_{ij}^k denote the stored size provided by edge server E_i for data prosumer P_{ji} in the k^{th} communication. In order to calculate the proportion of sd_{ij}^k in a cycle τ, we use SD_{ij}^k to represent this value which is

$$SD_{ij}^k = \frac{sd_{ij}^k}{\sum_{k=1}^{T_{ij}(\tau)} sd_{ij}^k} \tag{5}$$

where $T_{ij}(\tau)$ denotes the communication times between P_{ji} and E_i in a cycle τ.

The second evaluation indicator is the rating of the lifespan of edge server E_i. We consider an exponential decay function, which is

$$TL_i(\tau_k) = TL_i(0) \cdot e^{1-f(\tau-\tau_k)} \tag{6}$$

where $TL_i(0)$ is the initial lifespan of the cycle start, $f \in [0, \infty)$, and τ_k is the k^{th} communication timestamp.

The third evaluation indicator is the rating on the Secure Service Quality which is obtained after the communication between the data prosumer and the edge server. If the edge server is secure and trusted, the value will be high. Let SQ_{ij}^k denote the trust rating of the k^{th} communication between $E_i$and P_{ji}, here P_{ji} indicates that P_{ji} is controlled by E_i, we have

$$SQ_{ij}^k = \alpha_1 \times log(1 + Sq_{ij}^k) + \alpha_2 \times |sin(Sq_{ij}^k)| \tag{7}$$

where Sq_{ij}^k is the storage service quality of E_i for P_{ji} at the k^{th} communication and $0 \leq Sq_{ij}^k \leq 1$. Here, parameter α_1 and α_2 satisfy $0 \leq \alpha_1, \alpha_2 \leq 1$ and $\alpha_1 + \alpha_2 = 1$.

Therefore, we have the positive feedback as follows

$$PE_{ij}(\tau) = \sum_{k=1}^{T_{ij}(\tau)} SD_{ij}^k \cdot SQ_{ij}^k \cdot TL_i \tag{8}$$

Likewise, the negative feedback can be derived as

$$NE_{ij}(\tau) = \sum_{k=1}^{T_{ij}(\tau)} SD_{ij}^k \cdot (1 - SQ_{ij}^k) \cdot TL_i \tag{9}$$

In order to punish malicious behaviors of the edge server, a punishment factor γ for negative feedback, which is a real number greater than 1, is used. Therefore, the direct trust for E_i can be expressed by

$$DET_i(\tau) = \frac{\sum_{j \in J} PE_{ij}(\tau)}{\sum_{j \in J}(PE_{ij}(\tau) + \gamma \cdot NE_{ij}(\tau))} \tag{10}$$

Here, we consider that the data prosumers in the same neighborhood have an equal impact on E_i.

2) Edge Server Trust Value from Indirect Trust: Indirect trust can be regarded as recommendations from other data prosumers and edge servers in the same domain. The recommendation mechanism can enhance the accuracy of trust evaluation, especially when the data prosumer is not enough to know about the edge server that manages it. In order to multi-dimensionally evaluate the trust value of the edge server, the information provided by other edge servers and data prosumers in the same domain needs to be considered.

Data prosumers with high trust values always provide positive recommendations, while recommendations from data prosumers with a low trust may be malicious. Therefore, it is necessary to recommend a reputation to assure the trustworthiness of the recommendation. Here, every two edge servers have a certain similarity, as are every two data prosumers. If two edge servers have stronger similarity, the recommendations of them are more credible. In our scheme, the indirect trust of the edge server can be divided into three parts, including data prosumer trust value, data prosumer visited gateway similarity, and knowledge similarity on edge server.

We denote the trust value of P_{ji} as PT_j. Considering the impact of all data prosumers in the same domain with the same weight, we define the data prosumer trust value PT in the domain J during a cycle as

$$PT = \frac{1}{|J|} \sum_{j \in J} PT_j \tag{11}$$

Here, $|J|$ denotes the number of data prosumers in the domain J.

The visited gateway similarity means the closeness, which is an indicator to show whether two data prosumers have the same physical contacts. The message from a data prosumer needs to go through multiple gateways to reach its destination. Each data prosumer records the ID of the gateways that the message goes through. Let G_j and $G_{j'}$ denote the sets of the visited gateways of P_{ji} and $P_{j'i}$, respectively. Therefore, the commonly visited gateways can be calculated by

$$LCP(G_j, G_{j'}) = \begin{cases} 0, & if\ G_j\ or\ G_{j'}\ is\ empty \\ 1 + LCP(Pop(G_j), Pop(G_{j'})), & if\ G_j(0) = G_{j'}(0) \\ max(LCP(Pop(G_j), G_{j'}), LCP(G_j, Pop(G_{j'}))), & o.w. \end{cases} \tag{12}$$

where $Pop(G)$ means removing the first element from G. In order to normalize $LCP(G_j, G_{j'})$, its formula can be obtained by

$$LCP_{jj'} = \frac{LCP(G_j, G_{j'})}{max(|G_j|, |G_{j'}|)} \tag{13}$$

The knowledge similarity is based on the number of the same knowledge that two edge servers sustain. The knowledge contains the sub-grid information, the gateway information, service quality, and so on. Let K_i and $K_{i'}$ denote the sets of sustained knowledge of E_i and $E_{i'}$, respectively. The knowledge similarity can be derived by

$$K_{ii'} = \frac{\overrightarrow{PK}_i \cdot \overrightarrow{PK}_{i'}}{||\overrightarrow{PK}_i|| \cdot ||\overrightarrow{PK}_{i'}||} \tag{14}$$

where $\overrightarrow{PK}_i$ and $\overrightarrow{PK}_{i'}$ are the knowledge vectors of E_i and $E_{i'}$, respectively. Therefore, the trust between P_{ji} and $P_{j'i}$ can be calculated by

$$TP_{jj'} = w_m \cdot PT + w_p \cdot LCP_{jj'} \tag{15}$$

where w_m and w_p are weighted parameters, and $w_m + w_p = 1, 0 \leq w_m, w_p \leq 1$.

The credibility degree of $P_{j'i}$ measured by P_{ji} is defined as

$$CR_{jj'} = \frac{TP_{jj'}}{\sum_{j' \in J} TP_{jj'}} \tag{16}$$

By combining the credibility degree of data prosumer and edge server knowledge similarity, the indirect trust can be obtained by

$$\begin{aligned} IET_i = &\frac{\zeta_{ie}}{|J| \cdot |J|}(\sum_{j \in J} \sum_{j' \in J} CR_{jj'} \cdot DET_{ij'}) \\ &+ \frac{1 - \zeta_{ie}}{|J| \cdot |I|}(\sum_{j \in J} \sum_{i' \in I} K_{ii'} \cdot DET_{i'j}) \end{aligned} \tag{17}$$

Here, ζ_{ie} is the weight factor, which is between 0 and 1.

3) Data Prosumer Trust Value from Direct Trust: Through multiple observations of the observed data prosumer behavior, the data prosumer can evaluate the direct trust value based on the direct experiences by exploiting the rating of data prosumer active state, the requested data quantity, and the communication success rate of data prosumer.

The rating of data prosumer active state reflects the property of fluctuation with a lapse of time. If P_{ji} has not interacted with E_i for a long period, this indicator will become small gradually. Conversely, if P_{ji} interacts with E_i frequently for a period of time, it will increase quickly. The computation formula of E_i active state floating with time going is described as follows.

$$AS_{ij} = \begin{cases} AS_{ij} \cdot (1 + e^{(INum_{ij}^{new} - INum_{ij}^{old})}), INum_{ij}^{new} > INum_{ij}^{old} \\ AS_{ij} \cdot (1 - e^{(INum_{ij}^{new} - INum_{ij}^{old})}), INum_{ij}^{new} < INum_{ij}^{old} \\ \frac{AS_{ij}}{\Delta t^{\frac{1}{k}}}, \qquad INum_{ij}^{new} = 0 \end{cases} \tag{18}$$

Here, AS_{ij} denotes the rating of P_{ji} active state for E_i and Δt means a monitoring cycle. $INum_{ij}^{old}$ and $INum_{ij}^{new}$ denote the last and new interaction number between P_{ji} and E_i respectively. k is the controlling factor and has different values in heterogeneous networks.

In our proposed storage architecture, for P_{ji}, the requested data quantity is a pivotal attribute to evaluate the direct trust. Rd_{ij} denotes the rating of data quantity that P_{ji} requests from E_i, and it can be obtained as the follows:

$$Rd_{ij} = Rd_{ij}^{old} + (Rd_{ij}^{new} - Rd_{ij}^{old}) \times \begin{cases} \beta^{(Rd_{ij}^{new} - \mu + \frac{1}{INum_{ij}^{new}})}, Rd_{ij}^{new} > \mu \\ \beta^{(\mu - Rd_{ij}^{new})}, \qquad Rd_{ij}^{new} < \mu, Rd_{ij}^{new} < Rd_{ij}^{old} \\ \beta^{(Rd_{ij}^{new} - Rd_{ij}^{old})}, \qquad Rd_{ij}^{old} < Rd_{ij}^{new} < \mu \end{cases} \tag{19}$$

Rd_{ij}^{old} and Rd_{ij}^{new} denote the original and new rating of requested data quantity respectively. Here, the value larger than μ which is the critical threshold is trustworthy, while the smaller is distrustful. In addition, β is an adjustment factor and $0 < \beta < 1$. We can see that when β is small, Rd_{ij} increases slowly, while Rd_{ij} declines quickly. Obviously, Rd_{ij} is related to $INum_{ij}$. So when $Rd_{ij}^{new} > \mu$, $INum_{ij}^{new}$ must not equal 0.

For the data prosumer communication success rate, it can be calculated as

$$Rc_{ij} = \begin{cases} \frac{w_{new} \cdot C_{normal}}{C_{total}} + w_{old} \cdot Rc_{ij}^{old}, Rc_1 < RC < Rc_2 \\ \frac{C_{normal}}{C_{total}}, \qquad RC < Rc_1 \end{cases} \tag{20}$$

Here, Rc_{ij} means the success rate of data prosumer P_{ji} communicates with edge server i. We use RC to represent $|Rc_{ij}^{old} - \frac{C_{normal}}{C_{total}}|$. And C_{normal} is the normal communication number for a fixed time, whereas C_{total} is the total communication number. Rc_1 and Rc_2 are two thresholds between the original value and new value. w_{new} and w_{old} are two weight values for the original value and new value, whose values depend on monitoring sensitivity. If w_{new} is larger than w_{old}, the value of Rc_{ij} would converge to $\frac{C_{normal}}{C_{total}}$ quickly. Apart from the above conditions, Rc_{ij} is 0 in other conditions.

We can obtain the direct trust value of P_{ji} by calculating the overall attributes as follows:

$$DPT_j = \frac{1}{|I|} \sum_{i \in I} (\alpha_{as} \cdot AS_{ij} + \alpha_{rd} \cdot Rd_{ij} + \alpha_{rc} \cdot Rc_{ij}) \tag{21}$$

The relative importance assigned to three attributes is modeled as three real numbers, $\alpha_{as}, \alpha_{rd}, \alpha_{rc}$, which satisfy $\alpha_{as} + \alpha_{rd} + \alpha_{rc} = 1$ and $0 \leq \alpha_{as}, \alpha_{rd}, \alpha_{rc} \leq 1$. All weights are determined by the network controller.

4) Data Prosumer Trust Value from Indirect Trust: We divide the indirect trust of the data prosumer into two parts, one from the impact of the edge node that supervises it, and the other from the direct value of other data prosumers in the same domain.

For a single data prosumer's trust value, we consider not only its own direct trust, but also the impact of other data prosumers in its same domain. When a data prosumer is attacked, the closer the data prosumer is to the attacked prosumer, the more likely it is to be attacked, so we can infer that the closer the equivalent of the data prosumer, the higher the credibility, vice versa. Consequently, we evaluate the credibility of P_{ji} as

$$CP_j^{new} = CP_j^{old} + \theta \cdot (1 - \sum_{j' \in J \setminus j} w_{j'} \cdot (CP_{j'}^{old})^{\frac{1}{s}} - CP_j^{old}) \quad (22)$$

Here, CP_j^{new} and CP_j^{old} are the new and old credibilities of P_{ji} respectively. θ is an impact factor that is defined as

$$\theta = \frac{e^{|1 - \sum_{j' \in J \setminus j} w_{j'} \cdot (CP_{j'}^{old})^{\frac{1}{s}} - CP_j^{old}|} - 1}{e + 1} \quad (23)$$

Besides, s denotes a strictness factor which is used to control the curve. $\sum_{j' \in J \setminus j} (CP_{j'}^{old})^{\frac{1}{s}}$ means the summary of the $1/s$ powers of the credibility of other data prosumers in the domain J except P_{ji}. We have $\sum_{j' \in J \setminus j} w_{j'} = 1$, and $w_{j'}$ reflects the deviation of the credibility evaluation.

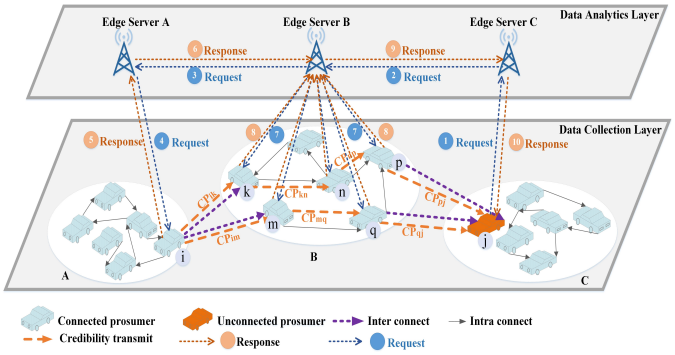

Fig. 5. The transmission of the credibility

In addition, when a data prosumer has no direct recommenders within the jurisdiction of the edge server that controls it, data prosumers from other edge servers within the same domain J can give an indirect recommendation, so as to obtain the credibility of P_{ji}. As shown in Figure 5, edge servers are at the data analytics layer and data prosumers are at the data collection layer in ITS big data analytics architecture [20]. P_{Ai} can interact with P_{Cj} through some paths. CP_{pi} is the credibility of the recommenders from the path pi. We can calculate the CP_{pi} as the following equation:

$$CP_{pi} = CP_{i1} \cdot CP_{12} \cdots CP_{mj} \quad (24)$$

CP_{xy} means the credibility of data prosumer x to data prosumer y, so the credibility of the data prosumer j can be obtained as follows:

$$CP_j^{new} = \xi_1 \cdot CP_j^{old} + \xi_2 \cdot \sum_{pi=1}^{k} wp_{pi} \cdot CP_{pi} \quad (25)$$

where $\sum_{pi=1}^{k} wp_{pi} = 1$, $\xi_1 + \xi_2 = 1$, $0 \leq \xi_1, \xi_2 \leq 1$, and wp_{pi} is related to the number of data prosumers on the path pi. Therefore, we have the indirect trust value of data prosumer P_{ji}:

$$IPT_j = \frac{\alpha_{im}}{|J| - 1} \sum_{j' \in J \setminus j} CP_{j'}^{new} \cdot DPT_{j'} + \frac{(1 - \alpha_{im})}{|I|} \sum_{i \in I} DET_i \quad (26)$$

Here, α_{im} is the indirect trust factor for the data prosumer, determined by the trustworthy of the network, and $0 \leq \alpha_{im} \leq 1$. Since a data prosumer may be controlled by more than one edge server, the direct trust value of the edge server in the domain is also taken into account when calculating the indirect trust value of the data prosumer.

V. Mathematical Modeling of Data Trust Storage and Security Invocation in the Proposed Architecture

A. Trusted dynamic storage model based on trust value and popularity

When data d_{ji} is uploaded to Master y_j of E_i, P_{ji} records the initial popularity of data as $DP_{d_{ji}}(0)$. The amount of containers in y_j to store data is K. In a time slot t_s, the data requested by each P_{ji} is a subset of the set $\hat{D} = \{1, 2, \cdots, D\}$. If the requested data is exactly in the container C_{jk}, no extra requesting cost for P_{ji} is incurred. Instead, if the requested data is not in C_{jk}, y_j will search its neighbors $NM = \{y_{jn} | n \in N\}$ at the first step, where N is the number of the neighbors of y_j. In this case, if the requested data is not recorded in NM, y_j will obtain the data from the cloud, which will result in a lot of costs, including a possible surge in electricity prices.

Suppose $a_s(t_s) \in A$ represents a storage action vector whose size is $D \times 1$ in slot t_s, where $A = \{a_s | a_s \in \{0, 1\}^D\}$. Here, $|a_s(t_s)|_{d_{ji}} = 1$ means that data d_{ji} is stored in a container of Master y_j, otherwise $|a_s(t_s)|_{d_{ji}} = 0$.

We update the popularity of the data based on requests received from P_{ji}, defined as

$$DP_{d_{ji}}(t_s) = \alpha \cdot DP_{d_{ji}}(t_s - 1) + (1 - \alpha) \cdot N_{d_{ji}} / \frac{1}{D} \sum_{q=1}^{D} N_{q_{ji}} \quad (27)$$

In the above equation, $N_{d_{ji}}$ represents the number of requests for d_{ji} at slot t_s, and $\frac{1}{D} \sum_{q=1}^{D} N_{q_{ji}}$ is the average size of P_{ji} requests at slot t_s, where y_j can record $N_{q_{ji}}$. At the end of slot t_s, considering the edge server trust value, our edge server state is expressed as

$$s_s(t_s) = [a_s^T(t_s), DP^T(t_s), ET^T(t_s)]^T \quad (28)$$

We evaluate the efficiency and the trustworthiness of the storage strategy by maintaining the most popular data with highly trustworthy edge servers using their available storage space and resources. In general, y_j compares $DP_{d_{ji}}$ with the set threshold DP. If $DP_{d_{ji}}$ is greater than DP, y_j will register data d_{ji}, and assign d_{ji} to container C_k which both has redundant storage space and is closest to P_{ji}. Otherwise, y_i uploads data d_{ji} to the cloud. In Q-learning, we estimate the conditional cost of this storage policy, expressed as $C(s_s(t_s-1), a_s(t_s)|ET(t_s), DP(t_s))$. At the end of time slot (t_s-1), the Master performs storage action $a_s(t_s)$ and calculates data $DP(t_s)$ upon updates $ET(t_s)$, so that the edge server state is transferred to $s_s(t_s)$.

The storage policy function is defined as $\pi : S \rightarrow A$ mapped the state $s_s \in S$ to the action. Storage action $a_s(t_s+1) = \pi(s_s(t_s))$ is executed under the control of the policy $\pi(\cdot)$ for the current state $s_s(t_s)$. Here, storage performance can be estimated by the state value function [38]

$$V_\pi(s_s(t_s)) = \lim_{T\to\infty} E[\sum_{\tau=t_s}^{T} \gamma^{\tau-T} C(s_s[\tau], \pi(s_s[\tau]))] \quad (29)$$

which is the total expected cost generated over the infinite time frame, with a future discount parameter $\gamma \in [0, 1)$. The discount factor γ adjusts the balance between current and future costs. The best strategy π^* to minimize cost is

$$\pi^* = \underset{\pi\in\Pi}{argmin}\, V_\pi(s_s), \quad \forall s_s \in S \quad (30)$$

where Π denotes the collection of all possible policies. To clearly understand how Q-learning works, we define a state-action value function based on the policy π, namely $Q_\pi(s_s, a_s)$ [39].

Algorithm 1 Trusted dynamic storage based on trust value and popularity

1: Initialize state $s_s(0)$ randomly, $Q_0(s_s, a_s) = 0 \quad \forall s_s, a_s$
2: Initialize $\epsilon_t \in (0, 1)$,step size λ, $\lambda_s = 1 - \lambda$
3: **for** $t_s = 1, 2, \cdots, t_{max}$ **do**
4: Take storage action $a_s(t_s)$ in a probabilistic manner
5:
$$a_s(t_s) = \begin{cases} \underset{a_s}{argmin}\, Q(s_s(t_s-1), a_s) & w.p.\ 1-\epsilon_t \\ random \quad a_s \in A & w.p. \quad \epsilon_t \end{cases}$$
6: Update $DP(t_s)$ based on prosumer requests
7: Set $s_s(t_s) = [a_s(t_s)^T, ET^T(t_s), DP^T(t_s)]^T$
8: Calculate cost $C(s(t_s-1), a_s(t_s)|ET(t_s), DP(t_s))$
9: Update $Q(s_s(t_s-1), a_s(t_s)) = \lambda_s \cdot Q(s_s(t_s-1), a_s(t_s)) + \lambda \cdot [C(s_s(t_s-1), a_s(t_s)|ET(t_s), DP(t_s)) + \gamma Q_{\underset{\alpha}{min}}(s_s(t_s), \alpha)]$
10: **end for**

In addition, ϵ_t-greedy algorithm is proposed to choose the best policy. Select the action $a_s(t_s)$ in the slot (t_s-1) to make $Q(s_s(t_s-1), a_s(t_s))$ reach the minimum with the probability of $(1-\epsilon_t)$, and we call this option as the exploit, using the current best action a_s to get the best outcome of next state. On the other hand, randomly select the action a_s from the action collection A with the probability of ϵ_t. This process is called exploration, and the selected action may become the best policy in the next state. The detailed description of this overall process is provided in Algorithm 1.

B. Data call latency model based on trusted value receiving request

Based on queuing theory, we mathematically model the total latency required to request data and execute computational tasks. The overall time can be divided into two components, one is between the edge server and the cloud, and the other is between the edge server and the data prosumer. For y_i, assume that there are S_i prosumers within its control range. Suppose that the process of from the request of P_{ji} to y_j follows the Poisson distribution, and the request rate of P_{ji} is expressed as λ_j. The size of data requested by P_{ji} is denoted by d_i, while the size of data generated by P_{ji} is expressed as pd_i. Moreover, P_{ji} requires a task of classifying and popularizing local data, with a pre-processing speed of v_i. When receiving requests from P_{ji}, y_j would firstly check the trust value of P_{ji} at a speeding tv_j. If the trust value is lower than the set threshold, y_j will reject the requests from P_{ji} until the trust value changes. Then, once y_j accepts the requests from P_{ji}, it will contact cloud or data containers to find the requested data. This searching speed for y_j is denoted by vm_j. The data and signals transmission delay between y_j and P_{ji} is t_{MP}, while the transmission latency of data and signals between y_j and cloud is t_{MC}. Here, the storage capacity of y_j can be expressed as CP_j.

$X(x_{ij})$ is defined as a $S \times E$ matrix with the element value of each column and row is a boolean value x_{ij}. Here, S is the maximum value of all S_i and E is the number of edge servers in our proposed scheme. We have

$$x_{ij} = \begin{cases} 1 & if \quad d_i \in container_k^j \\ 0 & if \quad d_i \notin container_k^j \end{cases} \quad (31)$$

In the data requesting process, y_j firstly checks the trusted value of P_i, when the trusted value is acceptable, y_j will check requested pd_{ji}. If data pd_i requested by P_i is private data, it searches preferentially in C_{jk}, and when the remaining storage space of C_{jk} is smaller than the size of pd_i, y_j needs to find pd_i in the remaining space of C_{jk} and in other data containers near C_{jk} whose storage is enough. On the other hand, if data pd_{ji} is public shared data meanwhile its popularity is lower than the set threshold, y_j needs to request pd_i from the cloud. We define a data requesting matrix $U(u_{ij})$, where

$$u_{ij} = \begin{cases} 1 & if \quad pd_i\ is\ requested \\ 0 & if \quad pd_i\ is\ not\ requested \end{cases} \quad (32)$$

According to queuing theory, for y_j, the processing latency of data task uploaded by P_{ji} can be calculated by the following formula

$$R_j^M = u_{ij} \cdot d_i \cdot \lambda_j / (vm_j + tv_j - \sum_{i=1}^{S} u_{ij} \cdot \lambda_j) \quad (33)$$

The total quantity of tasks assigned to multiple Masters in the same neighborhood to process cooperatively tasks uploaded by P_{ji} is

$$S_i^M = \sum_{j \in E} t_{MP} \cdot (u_{ij} \cdot pd_i + d_i) \tag{34}$$

Due to the limited speed of data tasks processed by each Master, it is necessary to ensure the speed of several prosumers managed by y_j does not exceed the constraint speed, therefore, the speed constraint is $vm_j > \sum_{i=1}^{S} u_{ij}\lambda_i$. Here, P_{ji} related latencies in processing data tasks themselves, including the trusted value calculation, local preprocessing, and request initiation procedures, which can be expressed as

$$T_j = (1 - \sum_{i \in E} u_{ij}) \cdot pd_i \cdot \lambda_j / (v_i - (1 - \sum_{i \in E} u_{ij})\lambda_i) \tag{35}$$

It is necessary to ensure the positive and negative of the denominator for the local data flow speed, which is $v_i - (1 - \sum_{j \in E} y_{ij})\lambda_i > 0$. For the transmission latency in the requested data transmission, in the case of y_j storing the data requested by P_{ji}, the transmission latency between P_{ji} and y_j is expressed as

$$T_j^{PM} = \sum_{i \in E} t_{PM} \cdot (u_{ij} \cdot pd_i + x_{ij} \cdot d_i) \tag{36}$$

On the other hand, y_j and other edge servers in the same neighborhood do not store the data requested by P_{ji}, then y_j needs to request data from the cloud, and the latency between y_j and the cloud is

$$T_i^{MC} = \sum_{j \in E} t_{MC} \cdot (1 - x_{ij}) \cdot d_i \tag{37}$$

In this equation, $1 - x_{ij}$ indicates that the data requested by P_{ji} is not in the edge servers, but is stored in the cloud.

A nonlinear optimization model can be used to model the efficiency of the proposed architecture to evaluate task processing delays. Our optimization model has three factual and necessary constraints: storage capacity size of the Master, the processing rate of the Master, and the pre-processing rate of the prosumer. In this scheduling model, our goal is to minimize the total time required for both data requests and transfers, which can be expressed as

$$min \quad T = \sum_{j=1}^{S} (R_j^M + T_j + T_j^{PM}) + \sum_{i=1}^{E} T_i^{MC} \tag{38}$$

Because the optimal local solution of the genetic algorithm is equal to the optimal global solution in the case where the optimization problem is convex, we can apply it to the minimum latency of the two kinds of data stored in the edge and the cloud. In the genetic algorithm, the data storage schemes of Masters is the population Pop, and population size E is the number of Masters. Pop specifically refers to the case where the edge server stores data and the allocation scheme for the data storage location is represented by a $0-1$ matrix. In general, fitness function F is represented by the target optimization function $T = \sum_{j=1}^{E} R_j^M + \sum_{i=1}^{S} (T_i + T_i^{PM} + T_i^{MC})$ in the genetic algorithm. A detailed description of this overall process is provided in Algorithm 2.

Algorithm 2 The genetic algorithm for trustworthy storage architecture

Input: Initial population Pop, population size E, etc.
Output: Overall optimal population Pop.

```
for iteration number less than a certain number do
    Calculate the fitness F of each scheme of Master
    Initialize the empty population newPop
    while not generate E children do
        Select two individuals from Pop based on F
        if random A less than cross probability then
            Perform Cross-operation on two schemes of Masters
        end if
        if random B less than mutation probability then
            Perform Mutation-operation on two schemes of Masters
        end if
        Add two new schemes of Masters to population newPop
    end while
    Replace Pop with newPop
end for
```

VI. Evaluation

In this section, we evaluate the trust value in the proposed trustworthy evaluation scheme and show the efficiency of the proposed storage architecture which is compared to the centralized storage in the cloud.

In order to study the trend of trust value changes of edge servers and data prosumers in different trusted situations, we divide the simulation scenario into four situations: full trusted situation, high trusted situation, low trusted situation, and full distrusted situation. Initially, the proportion of high trustworthy data prosumers in the above situations is 1, 0.67, 0.33, 0, respectively. SD_{ij}^k and $TL_i(\tau_k)$ of E_i individually follows a uniform distribution of $[0, 1]$, and decreases from 1. The secure service quality of the edge server is positively correlated with the trust value of the controlling data prosumers. $LCP_{jj'}$ between P_{ji} and $P_{j'i}$ is 70% uniformly distributed among $[0.7, 1]$ and 30% among $[0.1, 0.3]$. While $K_{ii'}$ between E_i and $E_{i'}$ is 65% uniformly distributed among $[0.6, 1]$ and 35% among $[0.2, 0.4]$. $TP_{jj'}$ between P_{ji} and $P_{j'i}$ is calculated by the previous equation, where the parameter w_m and w_p is 0.2 and 0.8. ζ_{ie} is 0.6 in the indirect trust of the edge server. The initial AS_{ij} of P_{ji} is 0.5.

For the high trustworthy data prosumer, AS_{ij} increases by 0.02 after each communication in the full and high trusted situation and decreases by 0.03 in the low trusted situation. Besides, Rd_{ij} increases by 0.03 after each communication in all situations. In addition, Rc_{ij} increases by 0.04 after each communication in all situations. For the low trustworthy data prosumer, AS_{ij} decreases by 0.03 after each communication in all situations. The initial Rd_{ij} is 0.5, which decreases by 0.04 after each communication in all situations. The initial Rc_{ij} is 0.5. Rd_{ij} decreases by 0.05 after each communication in all situations. Moreover, the weight parameters α_{as}, α_{rd}, and α_{rc} are 0.3, 0.3, and 0.4. Firstly, we perform the

simulation to evaluate the edge server direct trust value over the communication number. The initial direct trust value of each edge server is 0.5. The initial trust value of the high trustworthy data prosumer is 0.8, while the initial trust value of the low trustworthy data prosumer is 0.2.

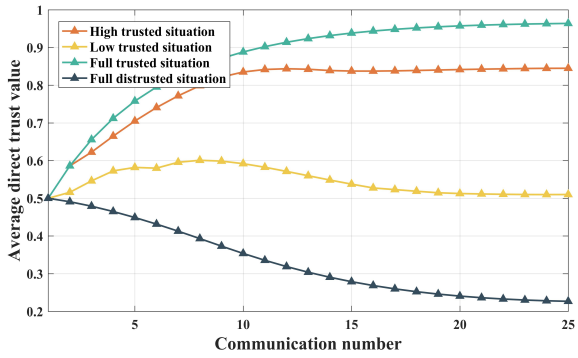

Fig. 6. Edge server direct trust value

Figure 6 shows the fluctuation of the average DET for all edge servers in four different trust environments. We can obtain that in the full trusted situation, the average DET tends to 1 as the communication number increases, due to the mounting positive feedback from the environment, that is, the environment is increasingly trusted. In the high trusted situation, the average DET inclines to 0.85 as the communication number increases, which is consistent with the trusted degree of the environment. In the low trusted situation, DET increases slightly and then stays at 0.5, because when the communication number is small, the high trustworthy data prosumer has a certain positive impact on the environment. But when the communication number increases, high trustworthy data prosumers are affected by the environment, making their PT gradually reduce. Hence, the average DET of all edge servers in the environment reaches a stable high level. In the full distrusted situation, the average DET has been decayed with the communication number.

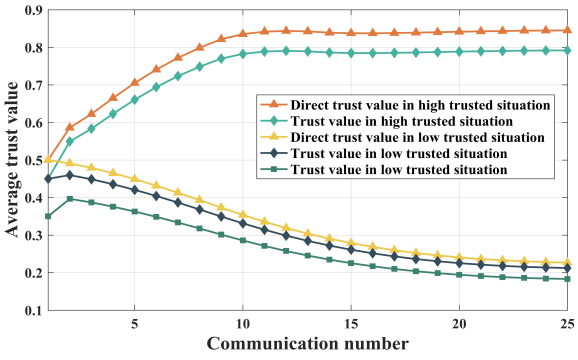

Fig. 7. Edge server trust value and direct trust value comparison

We can make four kinds of comparisons between the DET and ET of all edge servers in Figure 7. Case 1 is the comparison of DET and ET in the high trusted situation, and ET is lower than DET due to the impact of IET. Case 2 is the comparison of ET in the high trusted situation and low trusted situation, because of the influence from data prosumers, ET in low trusted situation decreases while in high trusted situation increases. Case 3 is the comparison of DET and ET in the low trusted situation. DET and ET both decrease, while ET is smaller than DET because of IET. Case 4 is the comparison of ET in the low trusted situation and the full distrusted situation. As shown, ET in the full distrusted situation is lower than that in the low trusted situation, and the difference is more obvious than Case 3.

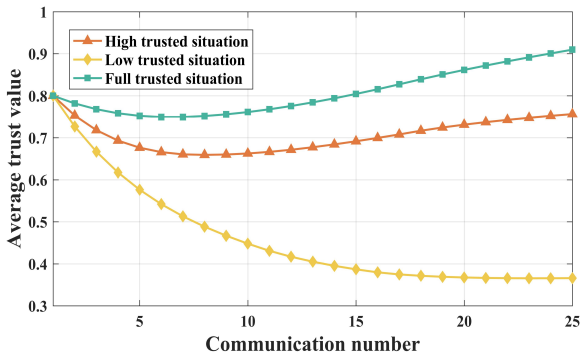

Fig. 8. High trustworthy data prosumer trust value

Figure 8 shows the fluctuation of the average PT for all high trustworthy prosumers in three different trust environments. Suppose that ET of edge servers is the same. As shown in the high trusted situation and the low trusted situation, PT of high trustworthy prosumers rises when the communication number is between 5 and 8. This is the impact of small-scale low trustworthy prosumers on IPT of high trustworthy prosumers. As the communication number increases, PT of low trustworthy prosumers becomes higher, so the overall average PT rises. In the low trusted situation, because the majority is low trustworthy prosumers, the environment where high trustworthy prosumers are located is very untrustworthy, and IPT has a great impact on high trustworthy prosumers so that the average PT continues to decrease.

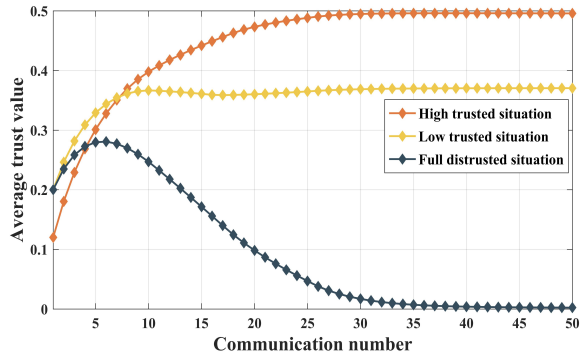

Fig. 9. Low trustworthy data prosumer trust value

Figure 9 depicts the fluctuation of the average PT for all low trustworthy prosumers in three different trust environments. The initial ET of edge servers remains the same in these contexts. In the high trusted situation and low trusted situation, PT of low trustworthy data prosumers is a slight increase but not more than 0.5, keeping at a low level of trustworthiness. As shown, the average PT in the high trusted situation is a little bit higher than that in the low trusted situation. However, in the full distrusted situation, IET from edge servers has an impact on the low trustworthy prosumers, so that the average PT increases by 0.1 at first. But all

prosumers are not trustworthy, IPT has a greater impact on the average. As the communication number increases, the trust value gradually approaches 0.

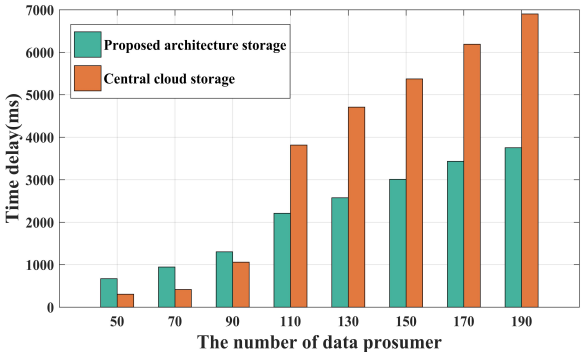

Fig. 10. Time delay of data requesting and receiving

Figure 10 shows the comparison of time delay in different storage architectures. This simulation is in a high trusted situation. The variable is S_i, the number of data prosumers, ranging from 50 to 190, with an interval of 20. The invariants for these data prosumers are the data receiving rate (1s/MB from edge servers, 2.5s/MB from the cloud), the size of requested data (from 1MB to 20MB), and the data requesting unit rate (from 1s to 10s). For Masters, the constants are the storage capacity (1000MB), the number E, and the size of C_{ik} (from 1500MB to 2000MB). As shown, when the number of data prosumers is less than 90, the time delay of our proposed architecture is higher than that of central cloud storage. This is because the time of interactive authentication and the data requesting unit rate, which are caused by the low trustworthy data prosumers' trust value not reaching the threshold, is longer. Far more than a certain amount, the delay of data requesting and receiving from the central cloud is higher than that of our proposal. So when data amount increases, our delay result is better.

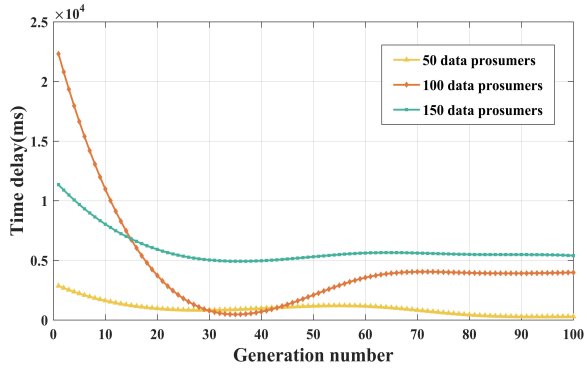

Fig. 11. The genetic algorithm convergence

Figure 11 depicts the convergence of the genetic algorithm when the number of data prosumers is changed from 50, 100, and 150, respectively, so the optimal allocation scheme is feasible. Because the genetic algorithm grows exponentially, its matrix dimension affects the delay of the search outcomes. It can be seen that when the number of prosumers is 50 and 100, the search delay greatly increases. Furthermore, because when the number of prosumers reaches a certain threshold, the difference between 100 prosumers and 150 prosumers is not very large due to the number of iterations. In addition, when the generation number is between 30 and 40, the curve of 100 data prosumers has a crossover coincidence with the curve of 150 data prosumers, because 100 is the key value of stability in the genetic algorithm and the volatility will be relatively large.

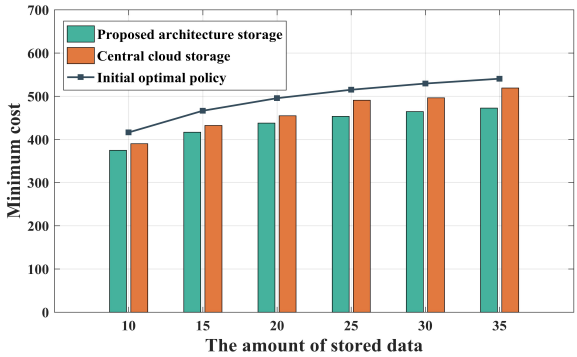

Fig. 12. The minimum cost of dynamic storage using Q-learning

To compare the proposed architecture storage and central cloud storage with the initial optimal caching strategy, in Figure 12, we simulate a small network with data volumes between 10 and 35. The capacity is 1 for each edge server with a total of 10 edge servers. The initial popularity is obtained from the Markov chain which is derived from the Zipf distribution parameters $\eta_1 = 0.8$ and $\eta_2 = 2.1$. Figure 12 notes the lowest cost for different amounts of data, comparing our proposed architecture storage with central cloud storage because both of them use Q-learning. We can see that the dynamic storage cost gained through Q-learning is always lower than the initial optimal policy. In the storage process, the proposed architecture is lower consumption than centralized cloud storage.

VII. Conclusion

In this paper, we worked on complex ITS scenarios and a large amount of data generated by edge nodes, which posed challenges for edge-cloud collaboration. Utilizing the advantages of Hadoop and trust evaluation, a unified trustworthy storage intelligent architecture in ITS was proposed to improve the performance of ITS services and better bolster the interaction of edge computing systems. The proposed federated trust evaluation ensured that communications in the proposed storage architecture are comparatively secure. Moreover, the simulations showed that the task processing delay and the minimum cost of the proposed architecture are better than the traditional method. In the future work, we will consider big data integrated analytics within storage under the edge of the distributed network.

References

[1] L. Qi, "Research on Intelligent Transportation System Technologies and Applications," *2008 Workshop on Power Electronics and Intelligent Transportation System, Guangzhou*, pp. 529-531, 2008.

[2] K. Wang, J. Wu, X. Zheng, A. Jolfaei, J. Li, and D. Yu, "Leveraging Energy Function Virtualization with Game Theory for Fault-Tolerant Smart Grid," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2020.2971584.

[3] J. Wu, M. Dong, K. Ota, J. Li, W. Yang, and M. Wang, "Fog Computing enabled Cognitive Network Function Virtualization for Information-Centric Future Internet," *IEEE Communications Magazine*, vol. 57, no. 7, pp. 48-54, 2019.

[4] G. Li, G. Xu, A. K. Sangaiah, J. Wu, and J. Li, "EdgeLaaS: Edge Learning as a Service for Knowledge-Centric Connected Healthcare," *IEEE Network*, in Press.

[5] A. Khan, A. Muhammad, Y. Kim, S. Park, and B. Tak, "EDGESTORE: A Single Namespace and Resource-Aware Federation File System for Edge Servers," *IEEE International Conference on Edge Computing*, pp. 101-108, 2018.

[6] A. Elgazar, M. Aazam, and K. Harras, "EdgeStore: Leveraging Edge Devices for Mobile Storage Offloading," *IEEE International Conference on Cloud Computing Technology and Science*, pp. 56-61, 2018.

[7] D. Huang, D. Han, J. Wang, J. Yin, X. Chen, X. Zhang, J. Zhou, and M. Ye, "Achieving Load Balance for Parallel Data Access on Distributed File Systems," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 388-402, 2018.

[8] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322-2358, 2017.

[9] Y. Tai, L. Wei, M. Xiao, H. Zhou, Q. Li, J. Shi, and S. Nahavandi, "A High-Immersive Medical Training Platform Using Direct Intraoperative Data," *IEEE Access*, vol. 6, pp. 69438-69452, 2018.

[10] J. Li, J. Wu, G. Xu, J. Li, X. Zheng, and A. Jolfaei, "Integrating NFV and ICN for Advanced Driver Assistance Systems," *IEEE Internet of Things Journal*, doi: 10.1109/JIOT.2019.2953988.

[11] X. Lin, J. Li, J. Wu, H. Liang, and W. Yang, "Making Knowledge Tradable in Edge-AI Enabled IoT: A Consortium Blockchain-based Efficient and Incentive Approach," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2019.2917307, 2019.

[12] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, "Big Data Analysis-Based Secure Cluster Management for Optimized Control Plane in Software-Defined Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 27-38, 2018.

[13] S. He, B. Cheng, H. Wang, X. Xiao, Y. Cao, and J. Chen, "Data security storage model for fog computing in large-scale IoT application," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 39-44, 2018.

[14] S. Ghane, A. Jolfaei, L. Kulik, K. Ramamohanarao, and D. Puthal, "Preserving Privacy in the Internet of Connected Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, doi: 10.1109/TITS.2020.2964410.

[15] J. Wu, M. Dong, K. Ota, J. Li, W. Yang, "Application-Aware Consensus Management for Software-defined Intelligent Blockchain in IoT," *IEEE Network*, vol. 34, no. 1, pp. 69-75, 2020.

[16] D. Kim, E. Ko, J. Son, Y. Kim, and J. Seo, "A Lightweight and Transparent Compensation Mechanism for Fog-Cloud Storage Framework," *IEEE Fourth International Conference on Big Data Computing Service and Applications*, pp. 254-259, 2018.

[17] J. Yuan and X. Li, "A multi-source feedback based trust calculation mechanism for edge computing," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 819-824, 2018.

[18] Y. Wang, J. Wen, W. Zhou, and F. Luo, "A Novel Dynamic Cloud Service Trust Evaluation Model in Cloud Computing," *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pp. 10-15, 2018.

[19] J. Zhang, "A Survey on Trust Management for VANETs," *2011 IEEE International Conference on Advanced Information Networking and Applications*, pp. 105-112, 2011.

[20] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big Data Analytics in Intelligent Transportation Systems: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383-398, 2019.

[21] W. Huang, G. Song, H. Hong, and K. Xie, "Deep Architecture for Traffic Flow Prediction: Deep Belief Networks With Multitask Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191-2201, 2014.

[22] S. Sivaraman and M. M. Trivedi, "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 267-276, 2010.

[23] I. Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128-135, 2010.

[24] L. Li, Y. Lv, and F. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 3, pp. 247-254, 2016.

[25] Y. Huang, X. Song, F. Ye, Y. Yang, and X. Li, "Fair Caching Algorithms for Peer Data Sharing in Pervasive Edge Computing Environments," *IEEE 37th International Conference on Distributed Computing Systems*, pp. 605-614, 2017.

[26] G. Wu, J. Chen, W Bao, X. Zhu, W. Xiao, J. Wang, and L. Liu, "MECCAS: Collaborative Storage Algorithm Based on Alternating Direction Method of Multipliers on Mobile Edge Cloud," *IEEE International Conference on Edge Computing*, pp. 40-46, 2017.

[27] X. Cao, J. Zhang, and H. V. Poor, "An Optimal Auction Mechanism for Mobile Edge Caching," *IEEE 38th International Conference on Distributed Computing Systems*, pp. 388-399, 2018.

[28] T. Wang, H. Luo, W. Jia, A. Liu, and M. Xie, "MTES: An intelligent Trust Evaluation Scheme in Sensor-Cloud enabled Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, doi: 10.1109/TII.2019.2930286, 2019.

[29] P. Zhou, K. Wang, J. Xu, and D. Wu, "Differentially-Private and Trustworthy Online Social Multimedia Big Data Retrieval in Edge Computing," *IEEE Transactions on Multimedia*, vol. 21, no. 3, pp. 539-554, 2019.

[30] K. A. Awan, I. U. Din, M. Zareei, M. Talha, M. Guizani, and S. U. Jadoon, "HoliTrust-A Holistic Cross-Domain Trust Management Mechanism for Service-Centric Internet of Things," *IEEE Access*, vol. 7, pp. 52191-52201, 2019.

[31] M. Femminella, M. Pergolesi, and G. Reali, "Performance Evaluation of Edge Cloud Computing System for Big Data Applications," *IEEE International Conference on Cloud Networking*, pp. 170-175, 2016.

[32] F. Qiao, M. Dong, K. Ota, S. Liao, J. Wu, and J. Li, "Making Big Data Intelligent Storable at the Edge: Storage Resource Intelligent Orchestration," *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6, 2019.

[33] D. Garcia-Carrillo and R. Marin-Lopez, "Multihop Bootstrapping With EAP Through CoAP Intermediaries for IoT," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4003-4017, 2018.

[34] C. Sun, "Fundamental Q-learning Algorithm in Finding Optimal Policy," *International Conference on Smart Grid and Electrical Automation*, pp. 243-246, 2017.

[35] H. Liang, J. Wu, S. Mumtaz, J. Li, X. Lin, and M. Wen, "MBID: Micro-Blockchain based Geographical Dynamic Intrusion Detection for V2X," *IEEE Communications Magazine*, vol. 57, no. 10, pp. 77-83, 2019.

[36] S. Adi and T. Yael, "Improved Online/Offline Signature Schemes," *Advances in Cryptology — CRYPTO 2001*; Kilian Joe; Publisher: *Springer Berlin Heidelberg*, Germany, pp. 355-367, 2001.

[37] H. Krawczyk and T. Rabin, "Chameleon Signatures," *Symposium on Network and Distributed Systems Security*, pp. 143-154, 2000.

[38] C. Zhang and Z. Zheng, "Task migration for mobile edge computing using deep reinforcement learning," *Future Generation Computer Systems*, vol. 96, pp. 111-118, 2019.

[39] C. Zhang, Z. Liu, B. Gu, K. Yamori, and Y. Tanaka, "A deep reinforcement learning based approach for cost- and energy-aware multi-flow mobile data offloading," *IEICE Transactions on Communications*, vol. E101.B, no. 7, pp. 1625-1634, 2018.

Fuli Qiao received the B.S. degree from the School of Mathematics, Southeast University, Nanjing, China, in 2018. Now, she is currently pursuing the Master degree with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. Her research interests are focusing on edge computing storage, Internet of Things, and so on. She is a Student Member of IEEE.

Jun Wu received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011. He was a Post-Doctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He was a Researcher with the Global Information and Telecommunication Institute, Waseda University, Japan, from 2011 to 2013. He is currently an associate professor of School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He is also the vice director of National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiao Tong University, China. He is the chair of IEEE P21451-1-5 Standard Working Group. He has hosted and participated in a lot of research projects including National Natural Science Foundation of China (NFSC), National 863 Plan and 973 Plan of China, Japan Society of the Promotion of Science Projects (JSPS), etc. His research interests include the advanced computing, communications and security techniques of software-defined networks (SDN), information-centric networks (ICN) smart grids, Internet of Things (IoT), 5G, etc., where he has published more than 120 refereed papers. He has been the Track Chair of VTC 2019 and the TPC Member of more than ten international conferences including ICC, GLOBECOM, WINCON, etc. He has been a Guest Editor of the IEEE Sensors Journal, Sensors, ICT Express. He is an Associate Editor of the IEEE Access.

Jianhua Li is a professor/Ph.D. supervisor and the dean of School of Cyber Security, Shanghai Jiao Tong University, Shanghai, China. He is also the director of National Engineering Laboratory for Information Content Analysis Technology, the director of Engineering Research Center for Network Information Security Management and Service of Chinese Ministry of Education, and the director of Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, China. He is the vice president of Association of Cyber Security Association of China. He got his BS, MS and Ph.D. degrees from Shanghai Jiao Tong University, in 1986, 1991 and 1998, respectively. He was the chief expert in the information security committee experts of National High Technology Research and Development Program of China (863 Program) of China. He was the leader of more than 30 state/province projects of China, and published more than 300 papers. He published 6 books and has about 20 patents. He made 3 standards and has 5 software copyrights. He got the Second Prize of National Technology Progress Award of China in 2005. His research interests include information security, signal process, computer network communication, etc.

Ali Kashif Bashir is a Senior Lecturer at the Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom. He is also with School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad (NUST), Islamabad, Pakistan. He is a senior member of IEEE and Distinguished Speaker of ACM. His past assignments include Associate Professor of Information and Communication Technologies, Faculty of Science and Technology, University of the Faroe Islands, Denmark; Osaka University, Japan (71 in QS Ranking 2020); Nara National College of Technology, Japan; the National Fusion Research Institute, South Korea; Southern Power Company Ltd., South Korea, and the Seoul Metropolitan Government, South Korea. He is also advising several startups in the field of STEM based education, robotics, and smart homes. Currently, he is supervising RNS Solutions, a multinational organization as Chief Research Officer. He received his Ph.D. in computer science and engineering from Korea University (83 in QS Ranking 2020) South Korea. MS from Ajou University, South Korea and BS from UMT, Pakistan. He is author of over 80 peer-reviewed articles. He is supervising/co-supervising several graduate (MS and PhD) students. His research interests include internet of things, wireless networks, distributed systems, network/cyber security, cloud/network function virtualization, etc. He is serving as the Editor-in-chief of the IEEE FUTURE DIRECTIONS NEWSLETTER.

Shahid Mumtaz has more than 10 years of wireless industry/academic experience and is currently working as Principal Research Scientist and Technical Manager at Instituto de Telecomunicações (IT) Portugal. He has received his Master and PhD degrees in Electrical & Electronic Engineering from Blekinge Institute of Technology, Sweden, and University of Aveiro, Portugal in 2006 and 2011, respectively. From 2005 to 2006 for Ericsson and Huawei at Research Labs in Sweden as Researcher. Since January 2015, he has been with the Department of Electrical and Electronic Engineering at the University of Aveiro, Portugal where he is Adjunct Associate Professor and Director of the Wireless Networking and Communications Group. He uses mathematical and system level tools to model and analyze emerging wireless communication architectures, leading to innovative and/or theoretically optimal new communication techniques. He is working closely with leading R&D groups in the industry to transition these ideas to practice. His work is currently supported by Samsung, Huawei, and Ericsson, as well as FCT. He has also been providing consultancy to Intel, Huawei, and Portugal Telecom. In January 2017, IEEE has started a new standard on P1932.1: Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Networks and was elected as Vice chair for this standard. Furthermore, in January 2015, he was also elected Vice-Chair of IEEE Research Project on Vision of Green Standardization due to his pioneer work in green communication as well. He is also actively involved in 3GPP standardization on LTE release 12 onwards, along with major manufacturers (Huawei and Intel).Dr Mumtaz has published 3 books and more than 150 publications in very high-rank IEEE transactions, journals, book chapters, international conferences. He is serving as Scientific Expert and Evaluator for Research Funding Agencies, such as European Commission, and COST.

Usman Tariq is a skilled research engineer with doctorate in Information and Communication Technology in Computer Science from Ajou University, S. Korea. Strong background in ad hoc networks and network communications. Experienced in managing and developing projects from conception to completion. Have worked in international, large scale and long term projects with multinational organizations. Currently, he is attached with Prince Sattam bin Abdul-Aziz University as an assistant professor in College of Computer Engineering and Science. Usman's research interests span networking and security fields. His current research is focused on several network security problems: botnets, denial-of-service attacks, and IP spoofing. Additionally, he is interested in methodologies for conducting security.