# Crowd-Learning: A Behavior Based Verification Method in Software-Defined Vehicular Networks with MEC Framework

***Abstract*—In the future open 5G internet of vehicles, identity verification is an important security issue. We find if the identity credentials of vehicles and infrastructures are stolen by adversaries (i.e., identity theft), the current cryptography based authentication methods can not cope with this problem. In this paper, we propose a behavior based verification method, named Crowd-Learning, by utilizing the idea of crowd in software-defined vehicular networks with mobile edge computing framework. It verifies vehicles and reduces the verification latency by estimating vehicles' behavior in advance. Meanwhile, it verifies infrastructures in the process of reinforcement learning based the idea of crowd intelligence. In Crowd-Learning, through incentive mechanism and distributed learning, those confidential infrastructures try to provide accurate data for future behavior estimation, and those fake infrastructures expose themselves iteratively. In experiments, we use traffic simulation tool SUMO to generate extensive vehicles' traces and evaluate the performance of Crowd-Learning verification method. The results show that Crowd-Learning verification method can ensure high verification accuracy for vehicles and infrastructures with satisfying low verification latency.**

***Index Terms*—Identity Verification, Vehicle Behavior, Crowd Intelligence, Mobile Edge Computing, Vehicular Networks.**

## I. Introduction

PUSHED by governments and car makers, internet of vehicles (IoV) has recently received increasing attentions. In IoV, information interaction occurs between vehicles and vehicles, or between vehicles and infrastructures (such as base stations, road side units). At present, some excellent technologies, like cryptography based authentication [1]–[3], blockchain based data transmission security technology [4], [5], malicious code implantation prevention [6], [7], can be applied to guarantee the security of IoV. To some extent, these technologies can ensure the identity of an entity and prevent malicious attacks from damaging or manipulating infrastructures and vehicles' electronic control units. However, there are still some unsolved problems about identity theft, especially in IoV.

Identity theft refers to attackers using various means to steal or deceive users' identity credentials and forges an entity to impersonate the original entity. Then they use the legal credentials to attack more users and commit malicious behavior such as modifying permissions, tampering with configurations, broadcasting false information. For example, in an online banking system, a criminal steals a legitimate user's username and corresponding login password. Because the username and the password used by this criminal are both legal, the common verification system can not identify this criminal. In IoV, there are following two cases regrading the identity theft.

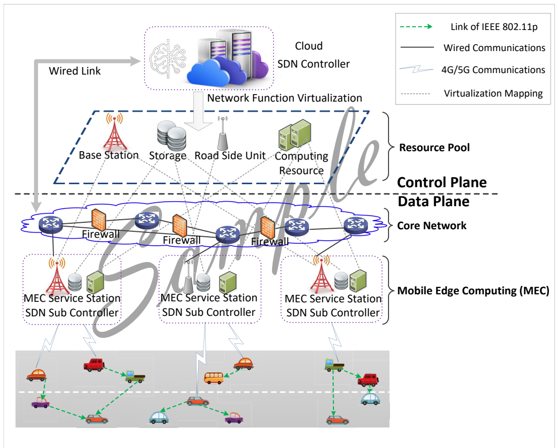


Fig. 1: An architecture of software-defined vehicular networks with MEC framework.

(1) For vehicles, there may be some illegal users who steal legal vehicles' identity credentials and obtain the driving permissions. We call vehicles the identity credentials of which are stolen by adversaries as **anomalous vehicles**.

(2) For infrastructures, many existing studies assume that infrastructures are trusted. But through stealing a confidential infrastructure's legal certificate, an adversary can invalidate and replace this original legal infrastructure. For simplicity, in the following, we use road side units (RSUs) on behalf of the infrastructures for further descriptions. We call these illegal RSUs as **fake RSUs**.

In IoV, fake RSUs and anomalous vehicles can commit various malicious behavior, such as sending false road conditions or sensor data. So the goal of our paper is to be able to identify/trace back to an anomalous vehicle or a fake RSU caused by identity theft. We need to design a new verification method to solve above problems. Our research is based on the existing cryptography. Besides, with the development of 5G technologies, vehicles will experience frequent handovers between different infrastructures during the movement. So, low latency of identity verification is needed.

Since our problem is in the domain of IoV, first we have to determine the architecture of IoV. Under future 5G technologies, the most popular network architecture of IoV proposed by many studies [8]–[10] is mobile edge computing (MEC) framework + software defined networking (SDN), which is shown in Fig. 1. This architecture is divided into two planes: control plane and data plane. In the top layer of the architecture, there is a SDN controller which can manage underlying facilities (e.g., switches) globally and control network traffics flexibly. In the middle layer, there is a core network composed of switches, which provides data forwarding and status collections via wired links. In the bottom layer, we use base stations or road side units as MEC service stations with scheduling storage and computing resources through network function virtualization. Meanwhile, each MEC service station is in charge of communicating with vehicles.

In this architecture, large calculations can be shifted to each MEC service station. Meanwhile, the SDN controller can maintain its global coordination ability. Since the identity verification in IoV requires lots of calculations to be completed with low latency, the architecture of MEC+SDN is indeed a promising architecture to meet this requirement and helps us to solve our problem.

Based on the problems mentioned above, we find that a driver has his/her own particular and stable driving behavior pattern. For example, a driver usually chooses a relative fixed route after determining a source and a destination. So the vehicle may arrive at a fixed location in a fixed period of time with a fixed direction and speed. We can define a vehicle's behavior as its timestamp, position, speed and driving direction. If a vehicle's identity is stolen, the vehicle's behavior may change because its driver changes. That is to say, due to identity theft, the vehicle may arrive at a fixed location in an unusual time, speed or direction, which deviates from the vehicle's usual habits. Therefore, we can solve the identity theft of vehicles by utilizing the stability of vehicles' behavior.

Then, since we need to satisfy the requirement of low latency when verifying vehicles, we want to obtain some behavior information about a forthcoming vehicle ahead of time from a previous RSU. Based on the information, we can start the behavior estimation for verification before the vehicle arrives. However, due to the existence of fake RSUs, we can not directly trust the information sent by only a RSU. Inspired by the idea of crowd intelligence, in the side of RSUs, we plan to call several RSUs alone the vehicle route to provide historical behavior data together for learning and estimating a vehicle's correct/trustful arrival behavior. The idea of crowd learning is similar to crowdsourcing. It describes the act of outsourcing a task or submitting a problem to a vast group of people (a crowd) in form of an open call. In this process, our method tries to obtain correct behavior results as much as possible through designing an incentive scoring policy based on reinforcement learning to motivate the crowd to provide accurate and true data. Since a fake RSU has no real historical behavior data about the passing vehicles, the fake RSU may expose itself in the process of crowd learning because of sending forged and incorrect behavior data.

Note that, except the identity theft, abnormal behavior[1] of a vehicle may also be caused by receiving false service information coming from a fake RSU or an anomalous vehicle. Fortunately, the idea of using vehicles' behavior to design verification method can not only identify vehicles with abnormal behavior caused by identity theft, but also can help us to trace back to anomalous objects (i.e., fake RSUs or anomalous vehicles) that cause vehicle behavior changes through carrying out malicious behavior. Besides, even if a vehicle passes through a fake RSU and does not show abnormal behavior, for example, this RSU does not interfere with the behavior of the vehicle, we can also utilize a distributed crowd learning to identify this hidden fake RSU. Our research is not to recognize a vehicle from tens of vehicles at the same time. Our aim is to use the stability of the vehicle's behavior pattern to solve the problem of identity theft in IoV.

So, in this paper, we propose a behavior based verification method, named Crowd-Learning, by utilizing the idea of crowd in software-defined vehicular networks with MEC framework. Crowd-Learning verification method is operated in each distributed RSU. Through central SDN guidance and distributed MEC calculation, we can solve the problem of identity theft and achieve the verifications with low latency. Especially, we design three key elements to support Crowd-Learning verification method, including incentive mechanism based on crowd learning, arrival behavior estimation based on decision tree and conflict decision based on D-S evidence theory. Meanwhile, we also give the collusion avoidance analysis and complexity analysis. Through extensive experiments, Crowd-Learning verification method shows good performance in reducing the verification latency substantially. Our verification method can still guarantee a verification accuracy of vehicles obove 90% under a relatively poor network security condition where the proportion of fake RSUs is 50% and the proportion of anomalous vehicles is 20%.

The cryptography based verification methods and our proposed behavior based verification method can be used as the first and second line of defense to guarantee the legality of vehicles' identities. They coexist in the IoV system. Meanwhile, the behavior based verification method proposed in this paper is scalable and can be extended to other human involved wireless mobile location-based systems.

The rest of this paper is organized as follows. We review related work in Section II, and present definitions and the framework of Crowd-Learning verification method in Section III. Then, we design the learning core of Crowd-Learning in detail in Section IV. And we give experiments and performance analysis in Section V. Finally, we conclude the paper in Section VI.

## II. Related Work

In recent years, cryptography based identity authentication technology has been studied in IoV. He et al. proposed an authentication technology based on conditional privacy in [1]. Lyu et al. designed a symmetric cryptography based authentication scheme called prediction-based authentication (PBA) in [11]. In order to reduce the verification delay for some emergency applications, PBA is designed to exploit the sender vehicle's ability to predict future beacons in advance. Zhang et al. and Shao et al. proposed authentication methods based on aggregation signature mechanism and group

[1] In our system, we limit one driver per vehicle by default. The user needs to register to the verification system. If a user drives a borrowed vehicle, he/she must register to the system and clarify his/her corresponding vehicle. The system will bind the historical data to the current vehicle and the borrowed vehicle will not be detected as a malicious vehicle. So our paper can extend to multi-user level behavior.

signature mechanism respectively in [12], [13]. Ying et al. provided an anonymous and lightweight authentication method based on smart card protocol in [14]. This method employs low-cost cryptographic operations to verify the legitimacy of vehicles and security of data messages. Pandi et al. proposed a dual group key management scheme that integrates fingerprint authentication techniques into a hash code creation method in vehicular ad hoc networks [15]. Liu et al. proposed a dual authentication scheme with considering vehicle reputation for V2V communications in [16]. This scheme exploits the advantage of bilinear pairing to compute encryption key without needing additional key management.

Besides, the studies that utilize behavior to do identity verification also appear in mobile or desktop clients for financial systems. Some studies take advantage of user keyboard input behavior [17] or touchscreen behavior [18]–[20] to do system login authentications. Recently, Liu et al. proposed a verification method based on transaction behavior sequences to detect financial transaction anomalies in [21], [22].

Compared with the related studies above, we can see that most traditional authentication methods rely on cryptography technology, but cannot solve identity theft. And they usually assume that all the infrastructures are trusted, but cannot identify the fake infrastructures in reality. Besides, due to the particularity of modeling objects, the existing behavior based verification methods in financial systems are not suitable to solve our problem of identity theft in IoV. Therefore, we propose a behavior based verification method, named Crowd-Learning, by utilizing the idea of crowd in IoV.

## III. Framework of Crowd-Learning Verification Method

### A. Some Definitions

#### 1) Definition of A Vehicle's Behavior

Assuming that there are $n$ vehicles and $m$ RSUs in the network, where $n$ and $m$ are constant. The RSUs cover the entire network area. Each RSU launches an identity verification actively for every entering vehicle.

We define the behavior of a vehicle as its driving properties, such as time, position (longitude and latitude), speed and driving direction. Here we use a tuple to represent a vehicle's behavior, i.e., $H_i^t = (lon, lat, v, dir)_i^t, 1 \leq i \leq n$, where $i$ denotes the label of a vehicle, $t$ denotes the current timestamp, $lon$ is the longitude, $lat$ is the latitude, $v$ denotes the speed, and $dir$ denotes the angle between the driving direction and the north direction geographically.

Each RSU accumulates lots of passing vehicles' behavior $H_i^t (1 \leq i \leq n)$ when doing verifications. Note that, the RSU only records the behavior data of the vehicles which pass the verification. If a vehicle does not pass the verification, the RSU will not record its behavior data.

#### 2) Different Roles of RSUs

In incentive crowd learning based on reinforcement learning, we define two roles for RSUs. We call the RSU which prepares and does the verifications for the forthcoming vehicles as executive RSU (ERSU), and call the RSU which sends historical behavior data to ERSUs as assistant RSU (ARSU). Since the computing core of Crowd-Learning verification method is operated in each distributed RSU, each RSU can be seen as an ERSU when doing the verification. There are usually one ERSU and several ARSUs making up of a process of incentive crowd learning in once Crowd-Learning running. From a macroscopic perspective, there are many simultaneous processes of incentive crowd learning based on reinforcement learning in the network.

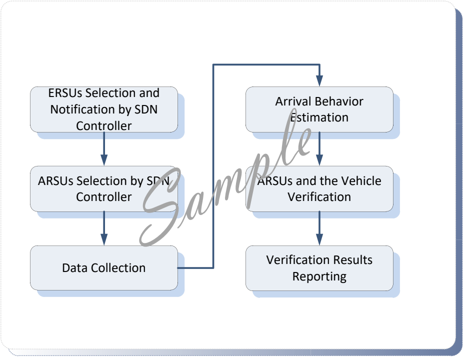

ERSUs Selection and Notification by SDN Controller
ARSUs Selection by SDN Controller
Data Collection
Arrival Behavior Estimation
ARSUs and the Vehicle Verification
Verification Results Reporting

Fig. 2: The framework of Crowd-Learning verification method.

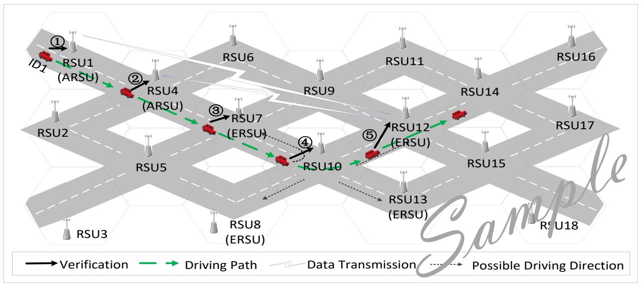


Fig. 3: An example scenario of Crowd-Learning verification method.

#### 3) Format of Verification Results Reporting

In Crowd-Learning verification method, the SDN controller possesses the road map of the entire city and the deployment of all RSUs. After running once Crowd-Learning, we ask the executive RSU (ERSU) to report the verification results of the passing vehicle and the related participating assistant RSUs (ARSUs) to the SDN controller. A format example of the recorded verification results is given below,

$$[(lon, lat, 1)_i^t; (RSU1, 1; ...; RSUj, 0; ...)^t],$$

where the former part is about verification result of vehicle $i$, and the latter part is about verification results of the participating RSUs. The meaning of notations $t$, $i$, $lon$, $lat$ is the same with the previous naming convention in $H_i^t$. Notations $RSU1, ..., RSUj, ...$ are the possible participating RSUs in the current incentive crowd learning based on reinforcement learning. $1$ indicates normal status (passing the verification) and $0$ indicates abnormal status. Therefore, the SDN controller can grasp each vehicle's driving trajectory.

### B. Procedures of Crowd-Learning Verification Method

In order to describe the framework of Crowd-Learning verification method clearly, we give following 6 steps depicted in Fig. 2. We design a notification-collection mechanism to reduce the verification latency for Crowd-Learning verification method in Step 1, Step 2 and Step 3.

**Step 1**. Supposing that a RSU has just completed verification for vehicle $i$. This RSU will report the verification result of vehicle $i$ to the SDN controller. Then, we design a notification-collection mechanism to notify some next RSUs (ERSUs) which vehicle $i$ may arrive at to prepare the future verification for vehicle $i$ in advance. Note that, SDN controller does not require to predict a vehicle's driving route. According to the road topology mastered by the SDN controller, the controller can choose those RSUs that are located at the intersections where the vehicle may arrive at as the next ERSUs.

**Step 2**. Next, the SDN controller will notify the ERSUs from which ARSUs to collect data. Considering privacy protection, the SDN controller will select some ARSUs the locations of which are inconsecutive along the route by using method[2] in [23]. The SDN controller tells the selection results to ERSUs. Then, the ERSUs will ask these ARSUs to send historical behavior data of vehicle $i$ with the condition that the historical data's timestamp should be ahead of the current time.

**Step 3**. The ERSUs receive the historical behavior data of vehicle $i$ from some related ARSUs based on above Step 1 and Step 2.

**Step 4**. The ERSUs estimate the behavior of vehicle $i$. In Crowd-Learning verification method, each ERSU utilizes reinforcement learning to collect true and accurate historical behavior data from some ARSUs designated by the SDN controller mentioned above through multiple iterations. Meanwhile, the ERSU calculates numbers of possible behavior of vehicle $i$ based on the data sent by each ARSU respectively. Since these behavior results may be conflicting with each other, we design a conflict decision method (in Section IV-C) to estimate vehicle $i$'s final trustful/correct arrival behavior results.

**Step 5**. When vehicle $i$ arrives at a certain RSU(ERSU) $j$, RSU $j$ only needs to compare the arrival behavior with the estimated correct behavior results of vehicle $i$. If they are the same, vehicle $i$ will pass the verification, and vice versa. The behavior comparison method is given in Section IV-D. Besides, in order to further ensure the security of IoV, after a vehicle passes verification of a RSU, the RSU will supervise this vehicle's behavior continuously based on real traffic conditions in its coverage area. If the RSU finds abnormal changes of this vehicle's behavior, the RSU will broadcast a risk warning to all the vehicles in its coverage area. For the remaining ERSUs, if vehicle $i$ does not arrive at within a time threshold, these ERSUs will terminate the preparation of verification for vehicle $i$. Meanwhile, since the data from each ARSU corresponds to an estimated behavior result, we can determine which results are trustful/correct by using conflict decision method in Step 4. Then, the corresponding ARSUs are considered to be confidential RSUs, and vice versa. Therefore we can verify these ARSUs at the same time.

**Step 6**. RSU $j$ reports the verification results of vehicle $i$ and the participating ARSUs to the SDN controller. If there are anomalous devices in the verification results, the SDN controller will check these reported anomalous vehicles and fake RSUs. If the reported results are false, the controller will doubt and trace back to the reporting RSU. Therefore, the RSUs dare not report abnormal results arbitrarily.

About the Step 6, here, we give further detailed explanations. If an anomalous vehicle is proven to have identity theft truly by the SDN controller, it will be removed from the network. Otherwise, if a vehicle changes its behavior because of accepting the false road condition information from a fake RSU, the SDN controller will trace back to the fake RSU which tells the false information to the vehicle. If a vehicle passes through a fake RSU and does not change its behavior, the fake RSU will be verified in the distributed Crowd-Learning verification process.

### *C. An Example Scenario of Crowd-Learning Verification Method*

We depict an example scenario in Fig. 3 to explain the framework of Crowd-Learning verification method in detail.

In Fig. 3, the driving route of vehicle ID1 is depicted by green dotted arrow lines. Vehicle ID1 has just completed identity verification at RSU1, RSU4, RSU7 and RSU10, depicted by black arrow lines with ①, ②, ③ and ④. RSU10 just reported the verification result of vehicle ID1 to the SDN controller.

Next, the SDN controller immediately notifies the nearby ERSUs. In Fig. 3, the four black dotted lines indicate four possible future driving routes of vehicle ID1, so RSU7, RSU8, RSU12 and RSU13 are the nearest ERSUs. Then, according to the notification-collection mechanism in Section III-B, the SDN controller selects the ARSUs (such as RSU1 and RSU4) based on the privacy principle.

After that, RSU7, RSU8, RSU12 and RSU13 send the verification task to RSU1 and RSU4 with requesting them to send the historical behavior data of vehicle ID1.

Taking RSU12 as an example, when RSU1 and RSU4 receive the task of RSU12, they send the behavior data of vehicle ID1 to RSU12. Then RSU12 estimates behavior of vehicle ID1. If RSU12 obtains conflicting behavior results based on historical data from RSU1 and RSU4 respectively, RSU12 will use the conflict decision method to estimate correct arrival behavior results.

When vehicle ID1 arrives at RSU12, depicted by ⑤, R-SU12 compares the arrival behavior with the estimated correct behavior results of vehicle ID1. If they are the same, vehicle ID1 will pass the verification. Besides, from RSU1 and RSU4, RSU12 can also find the fake RSU(s) the estimated behavior result of which is judged as distrustful through the conflict decision method.

Finally, RSU12 reports the verification result of vehicle ID1 and verification results of RSU1 and RSU4 to the SDN controller.

[2] Since selecting consecutive ARSUs along the route will let the ERSU grasp the precise trajectory of a vehicle and lead to privacy leakage, we should select ARSUs according to a privacy protection strategy. Here, our paper's research domain is not in the privacy protection, so any other similar privacy strategies also can be used in this part.

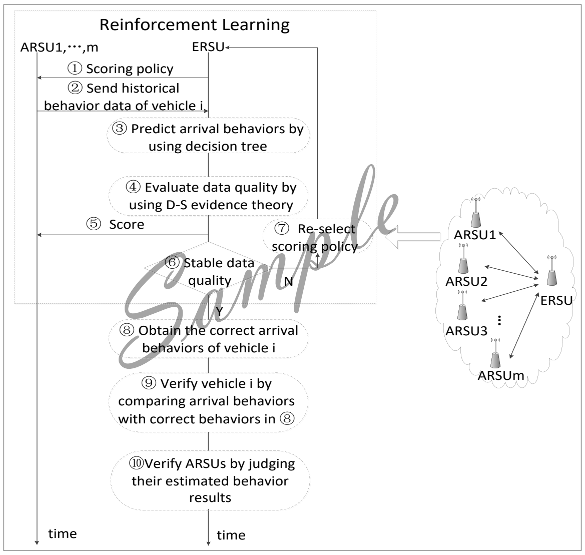

Reinforcement Learning
ARSU1,⋯,m
ERSU
① Scoring policy
② Send historical behavior data of vehicle i
③ Predict arrival behaviors by using decision tree
④ Evaluate data quality by using D-S evidence theory
⑤ Score
⑥ Stable data quality
N
Y
⑦ Re-select scoring policy
⑧ Obtain the correct arrival behaviors of vehicle i
⑨ Verify vehicle i by comparing arrival behaviors with correct behaviors in ⑧
⑩Verify ARSUs by judging their estimated behavior results
time
time
ARSU1
ARSU2
ARSU3
ARSUm
ERSU

Fig. 4: The flowchart of Crowd-Learning verification method among RSUs.

### D. Special Remarks

In most cases, the SDN controller can notify the nearest ERSUs which a vehicle may arrive at to prepare the verification in advance. But there are also few special cases, including (1) starting position and (2) no reporting. In the two cases, a vehicle may seem suddenly to appear in the coverage area of a RSU, but the RSU does not prepare the verification for this vehicle. In this subsection, we explain above special cases.

(1) **Starting Position**. Each vehicle has a consecutive trajectory in the space-time dimension when moving. The trajectory has consecutive positions and time intervals. Thus, each vehicle has a starting position in one of its consecutive trajectories. The RSU that this starting position belongs to is the starting RSU. When a vehicle locates in its starting position, the related starting RSU may find there is a vehicle that it does not prepare the authenticate for in advance.

There are two reasons that bring about this case. One is there are no preceding-hop RSUs around the starting RSU in this vehicle's trajectory. The other is the staying time of this vehicle is too long in the area of the current RSU.

As shown in Fig. 3, the starting position of vehicle ID1 belongs to RSU1. There are no preceding-hop RSUs. Therefore, when vehicle ID1 starts to move, the SDN controller can not notify RSU1 to prepare the verification for vehicle ID1 in advance.

In this paper, we design a starting verification method to solve this problem. The method verifies a vehicle according to the historical behavior data accumulated in the current RSU which the starting position belongs to. In the method, we cluster the historical behavior data to obtain the major behavior result of this vehicle. For example, it is assumed that the major behavior result of vehicle ID1 in RSU1 is 40km/h speed with north direction at 7:30. If the current behavior of vehicle ID1 is the same with this major behavior result, vehicle ID1 will pass the verification.

(2) **No Reporting**. If an ERSU is a fake RSU, it may not report the verification result of the passing vehicle to the SDN controller. So when the vehicle enters into the area of the next ERSU, the next ERSU does not prepare the authenticate for this vehicle in advance. In this case, we will verify all the previous related ERSUs in this verification to find the fake RSU.

As shown in Fig. 3, we observe a driving route of vehicle ID1 'RSU10→RSU12→RSU14'. RSU10 reports the verification result to the SDN controller after verifying vehicle ID1. Next, the SDN controller will notify next ERSUs (RSU7, RSU8, RSU12 and RSU13) to prepare the verification for vehicle ID1. Then the SDN controller waits for one of them to report the verification result. Assuming that RSU12 does not report the verification results to the SDN controller. When vehicle ID1 arrives at the coverage area of RSU14, RSU14 will find vehicle ID1 without previous verification preparation. So RSU14 will report this case to the SDN controller. The controller will verify the previous selected ERSUs (RSU7, RSU8, RSU12 and RSU13) to find which ERSU did not report the verification result. Since we solve the case of starting position in (1), RSU14 can differentiate whether this is the case of normal starting position or the case of no reporting.

## IV. The Learning Core of Crowd-Learning

In Crowd-Learning verification method, since the side of RSUs may be untrusted, we need a vehicle's historical behavior data from multiple RSUs (ARSUs) for predicting future possible arrival behavior accurately. The final correct arrival behavior results are obtained by fusing the estimated behavior results calculated based on the data sent by each ARSU respectively. In order to encourage confidential RSUs to send accurate data with reducing the rate of misjudgement, we use reinforcement learning to find an optimal scoring policy. The fake RSUs have no real historical behavior data and fabricating real data is very difficult. So, in the learning process, this incentive scoring policy makes the fake RSUs expose themselves.

Based on above idea, the learning core of Crowd-Learning verification method among RSUs are given in Fig. 4. In reinforcement learning, we design some different incentive policies. The agent (ERSU) uses these policies as its actions. And the reward is defined according to the overall quality of the data obtained by the agent.

So, in Fig. 4, first we can see that the learning agent (ERSU) selects a scoring policy randomly and sends it to related participating ARSUs. After collecting the data from the ARSUs, the agent uses decision tree to calculate the corresponding behavior results based on the data sent by each ARSU respectively. Then, the agent solves the conflict decision problem by using D-S evidence theory and evaluates the quality of the data sent by each ARSU. According to the scoring policies, each participating ARSU will obtain an incentive score.

If above learning iterations can not converge, the agent will re-select a scoring policy. This process goes on until iterations converge. Finally, we obtain the estimated trustful/correct behavior results of a vehicle through multiple iterations. Then, when this vehicle arrives, the agent (ERSU) verifies this vehicle by comparing its arrival behavior with the estimated correct behavior results. Meanwhile, the ERSU verifies the participating ARSUs by judging their estimated behavior results. So, Crowd-Learning verification method among RSUs both obtains accurate data quickly and achieves identity verifications.

In the following, first we introduce the reinforcement learning into our Crowd-Learning verification method in Section IV-A. Then, we give the core method of final behavior estimation, including behavior estimation based on decision tree in Section IV-B, conflict decision based on D-S evidence theory in Section IV-C and final verification in Section IV-D. Finally, we give a collusion avoidance analysis in Section IV-E, analyze the algorithm complexity in Section IV-F and present its implementation in Section IV-G.

TABLE I: Data Quality Level

| Level | Data Level Division Rule |
|---|---|
| 1 | The number of data is less than 10 items. |
| 2 | There are no conflicts of the data attributes among the $\xi + 1$ data subsets. |
| 3 | ① There are conflicts of the data attributes among first $\xi$ data subsets and the data attribute of $D_{j_1}$ is 'distrustful'. OR ② The number of available subsets is less than $\xi$ and the data attribute of $D_{j_1}$ is 'distrustful'. |
| 4 | ① There are conflicts of the data attributes among first $\xi$ data subsets and the data attribute of $D_{j_1}$ is 'trustful'. OR ② The number of available subsets is less than $\xi$ and the data attribute of $D_{j_1}$ is 'trustful'. |
| 5 | There are no conflicts of the data attributes among first $\xi$ data subsets but having conflicts with the $\xi + 1$ data subsets, and the data attribute of $D_{j_1}$ is 'distrustful'. |
| 6 | There are no conflicts of the data attributes among first $\xi$ data subsets but having conflicts with the $\xi + 1$ data subsets, and the data attribute of $D_{j_1}$ is 'trustful'. |

### *A. Reinforcement Learning Based Verification in Crowd*

There are many algorithms of reinforcement learning, such as Q-learning, Sarsa, TD learing. Since Q-learning has advantages in convergence and efficiency, we use Q-learning to find an optimal incentive scoring policy in Crowd-Learning verification method among RSUs. Here, an agent (ERSU) collects data iteratively by using Q-learning and finds an optimal scoring policy based on learning experiences. In each iteration, the agent needs to predict behavior result of a vehicle and determine the possible conflicting results to evaluate the data quality sent by each ARSU.

*1) Data Quality Evaluation*

It is very important to evaluate the quality of data sent by ARSUs. Too little data will lead to a biased estimation. Too much redundant data will lead to unnecessary transmission costs. Therefore, using reasonable incentive scoring policies can adjust the quality of data sent by ARSUs and motivate ARSUs to provide accurate data.

We evaluate the data quality in each iteration of Q-learning. The data quality is divided into 6 levels according to data level division rule, listed in TABLE I. We can see the division rule takes into account the data quantity and the data attribute together.

As to the data quantity, we have three types: 'sufficient', 'insufficient' and 'redundant'. Assuming that the agent (ERSU) is ready to check a certain vehicle. The data sent by ARSU $j$ to the agent in an iteration is denoted as $D_j$. Specially, we use notation $\xi$ denote a boundary to divide 'sufficient' data and 'redundant' data. The value of $\xi$ will be given in the experiment (Section V-B). We select $\xi + 1$ subsets $\{D_{j_1}, D_{j_2}, ..., D_{j_\xi}, D_{j_{\xi+1}}\}$ from $D_j$. We use 10 items of data as a sample interval. The 10 items are selected randomly from $D_j$. Then we set $|D_{j_1}| = |D_j|$, $|D_{j_2}| = |D_j| - 10, ...$, $|D_{j_\xi}| = |D_j| - 10\xi$, $|D_{j_{\xi+1}}| = |D_j| - 10(\xi + 1)$. The quantity of $\xi + 1$ data subsets is in a descending order.

As to the data attribute, we define the concept of data attribute for above each data subset. We judge the data attribute of a data subset as 'trustful' or 'distrustful' through conflict decision method based on D-S evidence theory (Section IV-C).

From TABLE I, we can see the judgement of the data attribute tends to be stable as the amount of data increases. Besides in a common sense, the data attribute decided by a large data subset shows more persuasive than that decided by a small data subset.

In Level 1, when the number of data is less than 10 items, it means the quantity is too small to let the agent carry out behavior estimation. In the experiment, we will explain and verify why we set 10 data items here. In Level 3 and Level 4, there are conflicts of the data attributes among first $\xi$ data subsets. It indicates that the data quantity sent by the current ARSU is still 'insufficient' to support the agent to obtain a stable behavior estimation result. In Level 5 and Level 6, there are no conflicts among first $\xi$ larger data subsets, but they conflict with the smaller data subset $D_{j_{\xi+1}}$. It indicates that the data attributes without conflicts appear $\xi$ times continuously. As we have stated above, the data attribute decided by a large data subset is more persuasive. It proves the data quantity is just 'sufficient' and reaches the boundary $\xi$ between 'sufficient' and 'redundant'. Finally in Level 2, there are no conflicts among the $\xi + 1$ data subsets. It indicates that the data attributes without conflicts appear more than $\xi$ times continuously. So the data quantity is 'redundant'.

Based on above data level, assuming that there are 1 ERSU and $q$ ($q < m$) ARSUs participating in a verification task. The quality level of the data sent by ARSU $j$ at $k$-th iteration is denoted by $L_j^{(k)} (1 \leq j \leq q, L_j^{(k)} = 1, 2, ..., 6)$. The proportion of each quality level data obtained by the agent (ERSU) is denoted by $N_l^{(k)}$,

$$N_l^{(k)} = \frac{\sum_{j=1}^{q} \phi(L_j^{(k)} = l)}{q}, \ (l = 1, 2, ..., 6)$$

where $\phi$ is an identify function taking the value 1 if $L_j^{(k)} = l$.

*2) Incentive Scoring Policies*

In Crowd-Learning verification method, we give each RSU an equal initial score, which is set to zero in the experiment. In our incentive scoring policies, we hope to obtain some 'trustful' data with appropriate data quantity. The incentive scoring policies for different data quality levels are quantized into 3 types, listed in TABLE II.

*Rule 1*: In order to avoid the misjudgement caused by too little data and the communication congestion caused by too

TABLE II: Incentive Scoring Policies

| Level<br>Policy | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Policy 1 | -1 | -1 | -0.8 | -0.6 | -0.4 | +1 |
| Policy 2 | -1 | -1 | -1 | +1 | -1 | +1 |
| Policy 3 | -1 | -1 | -0.5 | -0.5 | -0.3 | +1 |

much data, all three policies give the lowest score $-1$ to Level 1 and Level 2.

*Rule 2*: We have to prevent the ARSU continuously adding unnecessary redundant data for obtaining higher scores. When an ARSU sends 'sufficient' data (i.e., Level 5 and Level 6) twice in succession, ERSU will no longer collect its data.

**Policy 1.** As the data quality level becomes high, the score increases gradually.

**Policy 2.** In this policy, we only consider the data attribute. Except Level 1 and Level 2, if the data attribute is 'trustful', the score is set to the same positive score $+1$. If the data attribute is 'distrustful', the score is set to the same negative score $-1$.

**Policy 3.** If the data quantity is 'insufficient' to support a stable behavior estimation result (i.e., Level 3 and Level 4), the score is set to same negative score -0.5. If the data quantity is 'sufficient' (i.e., Level 5 and Level 6), we further consider the different scores (i.e., $-0.3$ and $+1$) based on the different data attributes.

### 3) Q-learning Applied to Crowd-Learning Verification Method

We give definitions of the action, state and reward of Q-learning as follows.

- *Action*: We have action $a^{(k)} \in A$, in which $A$ denotes a set of actions with including above 3 scoring policies.
- *State*: We use notation $s^{(k)}$ to denote a system state. Since the agent wants RSUs (ARSUs) to send high quality level data to avoid misjudgment [24], we have

$$s^{(k)} = [N_{1\leq l\leq 6}^{(k-1)}, a^{(k-1)}].$$

- *Reward*: We define a reward $r(s^{(k)}, a^{(k)})$ based on the difference of data quality received by the agent, having,

$$r(s^{(k)}, a^{(k)}) = \sum_{l=1}^{6} l \cdot N_l^{(k)} - \sum_{l=1}^{6} l \cdot N_l^{(k-1)}.$$

We describe our Crowd-Learning verification method among RSUs by using Q-learning in Algorithm 1. After initializing the algorithm, the agent selects an action via $\epsilon$-greedy strategy in Steps 2-4. $\epsilon$-greedy strategy is used to avoid staying in a local maxima. It indicates that there is a probability $\epsilon$ for the agent to select an action according to the optimal value of Q matrix. And there is a probability $1-\epsilon$ to select other actions.

After receiving the historical data of vehicle $i$ from the ARSUs, the agent (ERSU) evaluates the data quality of each ARSU separately and gives the corresponding score to each ARSU, as shown in Steps 6-7. Next, the agent obtains the proportion of each quality level data $N_{1\leq l\leq 6}^{(k)}$ in Step 8.

**Algorithm 1** Crowd-Learning Verification Method among RSUs

```
Initialize:
    $s^{(0)} = \emptyset$, $N_{1\leq l\leq 6}^{(0)} = 0$, $Q(s,a) = 0, \forall s, a$
    Select action $a^{(0)}$ randomly
    Broadcast scoring policy $a^{(0)}$ to all possible participating ARSU1...ARSU$q$
1:  for $k = 1, 2, 3, ...$ do
2:      $s^{(k)} = [N_{1\leq l\leq 6}^{(k-1)}, a^{(k-1)}]$
3:      Select action $a^{(k)}$ via the $\epsilon$-greedy algorithm
4:      Broadcast scoring policy $a^{(k)}$ to ARSU1...ARSU$q$
5:      Receiving historical behavior data of vehicle $i$ from ARSU1...ARSU$q$
6:      Apply the behavior estimation algorithm based on decision tree in Section IV-B and conflict decision algorithm based on D-S evidence theory in Section IV-C to evaluate the data quality level $L_j^{(k)} (1 \leq j \leq q)$
7:      Give the corresponding score to ARSU1...ARSU$q$ based on $a^{(k)}$
8:      Calculate $N_{1\leq l\leq 6}^{(k)}$
9:      Update $Q(s^{(k)}, a^{(k)})$ and $s^{(k+1)}$
10:     for $j = 1, 2, ..., q$ do
11:         if $ARSUj$ sends data with Level 5 or Level 6 in succession then
12:             The agent do not receive data from $ARSUj$ in Step 5 and do not give scores to $ARSUj$ further in Step 7
13:         end if
14:     end for
15:     if the value of $N_6^{(k)}$ does not change compared with $N_6^{(k-1)}$ then
16:         Break
17:     end if
18: end for
19: Obtain the estimated correct behavior results of vehicle $i$
20: Verify vehicle $i$ by comparing arrival behavior with the estimated correct behavior results
21: Verify ARSU1...ARSU$q$ by judging their estimated behavior results
```

Then, the agent updates Q value $Q(s^{(k)}, a^{(k)})$ and state $s^{(k+1)}$ in Step 9. The Q value is updated according to the Bellman equation as follows:

$$\begin{aligned} Q(s^{(k)}, a^{(k)}) \leftarrow Q(s^{(k)}, a^{(k)}) + \alpha[r(s^{(k)}, a^{(k)}) \\ + \gamma max Q(s^{(k+1)}, a^{(k+1)}) - Q(s^{(k)}, a^{(k)})], \end{aligned}$$

where $\alpha \in (0, 1]$ is a learning rate and $\gamma \in [0, 1]$ is a discount factor. $\gamma = 0$ indicates the agent only considers the current reward of the action; $\gamma = 1$ indicates the agent pays special attention to the future reward when selecting actions.

The agent controls the unnecessary redundant data for obtaining higher scores in Steps 10-14. In Crowd-Learning verification method, the key is to obtain as much $N_6^{(k)}$ as possible in multiple iterations. When the proportion of the highest quality level data $N_6^{(k)}$ no longer changes, we believe that the current data quality received by the agent is stable. It is enough to make the agent to do accurate behavior estimation. Therefore, Steps 1-14 are repeated until the data quality is stable in Steps 15-17. After that, the agent obtains estimated correct behavior results to verify vehicle $i$ and the participating ARSUs in Steps 19-21.

## B. Arrival Behavior Estimation

In Step 5 of Algorithm 1, each participating ARSU sends historical behavior data of vehicle $i$ to the agent (ERSU). The arrival behavior estimation happens between this ERSU and each ARSU. That is to say, the number of results is $q$. In this process, first, based on the behavior data sent by a certain ARSU, the ERSU needs to form ARSU-ERSU data pairs in its local storage (Section IV-B1). A sample of ARSU-ERSU data pairs is shown in Fig. 5.

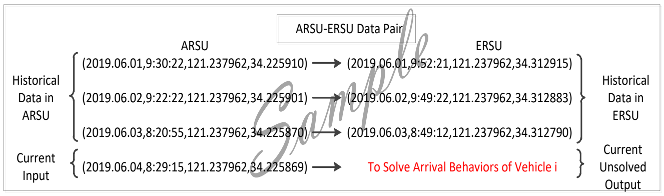


Fig. 5: A sample of ARSU-ERSU data pairs.

Then, in our problem, we need to predict the arrival behavior of vehicle $i$ based on these ARSU-ERSU data pairs. Although many machine learning methods can be used, our data presents a clear time-phased characteristic, i.e., workday, weekend, rush hours and off-peak hours. So some linear regression models are not fit for our problem. Here, we use decision tree to predict the behavior of vehicle $i$ based on the ARSU-ERSU data pairs (Section IV-B2). Of course, we can select more complex algorithm to realize the prediction, like random forests. However, our research domain is not to study the classification algorithm. So, in the paper, according to the quantity and characteristics of the data, decision tree algorithm is enough to cope with our problem.

### 1) *Formation of ARSU-ERSU Data Pairs*

According to Fig. 5, we give the coupling formation method of ARSU-ERSU data pairs in Algorithm 2. Assuming that an ARSU $j$ sends historical behavior data of vehicle $i$ to the agent (ERSU). The aim of coupling is to find the data pairs in which the data flows from ARSU $j$ to the agent with removing the data in the opposite direction. If the data themselves are untrusted, the coupling result will also be wrong. It is just good for us to find the fake RSU.

In Section IV-A1, we have given the notation of the historical data sent by ARSU $j$ to the agent as $D_j$. Here, the historical data in ERSU (agent) is denoted as $\overline{D}$. We have defined a vehicle's behavior as a tuple $H_i^t$, with including features 'time, position, speed, driving direction'. In Algorithm 2, we use notations $t_{D_j[x]}$ and $t_{\overline{D}[y]}$ to represent the value of feature time in $x$-th data item of $D_j$ and $y$-th data item of $\overline{D}$, respectively.

### 2) *Behavior Estimation Based on Decision Tree*

There are some famous decision tree algorithms, e.g., ID3, C4.5 and CART. ID3 relies on the feature that appears more frequently in the sample, but this feature may not necessarily optimal. CART recursively constructs a binary decision tree with selecting segmentation points, but selecting segmentation points is difficult to set in our problem. Thus, we use C4.5 algorithm to predict the behavior of the forthcoming vehicles.

The building of the decision tree is based on the historical data of ARSU-ERSU data pairs like Fig. 5. The data in ARSU constitute the branches of the decision tree. The data in the side of ERSU constitute the leaf nodes of the tree. Here, we do not state this classic C4.5 algorithm again, but there are several special settings that need to be explained in detail.

**Algorithm 2** Formation of ARSU-ERSU Data Pairs

```
Set two pointer $x$ and $y$ for $D_j$ and $\overline{D}$, respectively.
Sort the data in $D_j$ and $\overline{D}$ by time (including date)
while $x \le |\{D_j\}|$ and $y \le |\overline{D}|$ do
    if $t_{\overline{D}[y]} > t_{D_j[x+1]}$ then
        Remove the data item in ARSU with $t_{D_j[x]}$
        $x$++
    else if $t_{\overline{D}[y]} < t_{D_j[x]}$ then
        Remove the data item in ERSU with $t_{\overline{D}[y]}$
        $y$++
    else
        $x$++; $y$++
    end if
end while
if $x > |D_j|$ or $y > |\overline{D}|$ then
    Remove the remaining data items in $D_j$ or $\overline{D}$
end if
Finally we have $|D_j| = |\overline{D}|$ and re-sort the current data in $D_j$ and $\overline{D}$
Obtain one to one ARSU-ERSU data pairs
```

According to C4.5 algorithm, we need to first calculate the information gains of the candidate partitioning features. And then, find features the information gain of which is higher than the average level. Finally, we select the feature with the highest gain rate as the dividing node. So, first we need to determine the features that are going to be used when building the decision tree. Second, we need to discretize the continuous values of these features. Third, in order to control the scale of leaf nodes, some similar data in the side of ERSU will be clustered to some extent.

① ***Feature Selection***

Considering the difference of traffic flows between weekdays and weekends, we add a feature named 'weekend tag' with value of 0/1. 1 represents the current date is a weekend, 0 represents the current date is a weekday. Therefore, the features used in the decision tree includes: weekend tag, time, position, speed and driving direction.

② ***Feature Discretization***

The decision tree cannot directly divide the branch nodes according to the continuous values of above features. So we have to discretize the continuous values in advance.

Due to the existence of morning and evening rush hours, the daily traffic flows satisfy a normal distribution rather than a uniform distribution. Therefore, each feature may be discretized into different unequal intervals.

- Time Discretization.

We first divide the time into several coarse-grained intervals according to rush hours and off-peak hours. The intervals include 0:00-6:00, 6:00-10:00, 10:00-16:00, 16:00-19:00, 19:00-24:00. Then, based on the time value in the current input of the ARSU, we observe the time value falls in which time intervals above. Furthermore, we divide this time interval into fine-grained subintervals by using 30min as a gap.

For example, supposing that the time value in the current input of the ARSU is 08:29:15. So the time value falls in the interval 6:00-10:00. Then we divide the time interval 6:00-10:00 into eight equal fine-grained subintervals. Combining the coarse-grained and fined-grained time intervals, if a time value falls in one of the intervals, there will be a discretization branch. If we want to obtain more fine-grained prediction results, a smaller time gap which is less than 30min also can be used in our problem.

- Speed Discretization.

When a vehicle is at stage of rush hours, its speed is usually 0-30km/h. If not, its speed can reach 30-60km/h. When a vehicle is in a highway, its speed usually reaches 60-90km/h or 90-120km/h. Therefore, we divide the speed into four intervals [0km/h, 30km/h), [30km/h, 60km/h), [60km/h, 90km/h), [90km/h, 140km/h].

- Position Discretization.

There are several RSUs in IoV. The coverage area of each RSU can be seen as a hexagon cellular. We can divide each hexagon into six equilateral triangle parts. Then according to the position values in historical data of ARSU, if a position value falls in one of the equilateral triangle areas, there will be a discretization branch.

- Driving Direction Discretization.

Almost all the roads are relatively fixed. So the driving direction on a road is determined with only a small steering deviation. We discretize the angles into six intervals [0°,60°), [60°,120°), [120°,180°), [180°,240°), [240°,300°), [300°,360°). Then according to the driving direction values in historical data of ARSU, if a value falls in one of above six intervals, there will be a discretization branch.

③ ***Leaf Node Clustering***

In the ERSU (agent), there are some historical data used as leaf nodes when building a decision tree. However, the continuous values will result in many leaf nodes. It is unreal for building a decision tree. In this paper, we use K-Means clustering algorithm to group the similar behavior in the ERSU. These clusters are used as leaf nodes. In the experiment[3], the value of $K$ is set to 5.

All in all, by using above feature selection, feature discretizion and leaf node clustering, we can build a decision tree based on the ARSU-ERSU data pairs. Then, if an ARSU provides a current input to the decision tree, the agent will obtain a current output. The arrival behavior estimation is completed between each participating ARSU and the agent (ERSU).

### *C. Conflict Decision of Different Evidences*

Assuming that there are 1 ERSU and $q$ ($q < m$) ARSUs participating in a verification task. Since we collect a vehicle's historical behavior data from multiple RSUs (ARSUs), we will obtain $q$ results in which conflicts may exist via the behavior estimation (Section IV-B). However, we have to determine which one is trustful and which one is distrustful so as to verify vehicles and RSUs. D-S evidence theory is a reasoning theory proposed by Dempster and Shafer, with the ability of processing some uncertain information. Therefore, in the paper, we utilize D-S evidence theory to decide the conflicts. Different evidences may have different supporting degrees for an estimation result. So there are also conflicts between the evidences.

#### *1) Building Model for Different Evidences*

In D-S evidence theory, the uncertainty description for evidences usually includes a recognition framework, a basic probability assignment (BPA) function, a belief function and so on.

In Crowd-Learning verification method, for each ARSU-ERSU behavior result, we define a five-tuple abstract model $< \Theta, F, BPA, BEL, T >$ to identify this estimation result as trustful or untrustful.

① $\Theta$ is a recognition framework with a set of propositions, denoted by $\Theta = \{\theta_1, \theta_2, ...\}$. Each element in $\Theta$ represents a proposition. Here, we have two propositions, one is the estimation result is trustful (i.e., $\theta_1 = trustful$), the other is the estimation result is distrustful (i.e., $\theta_2 = distrustful$). So, we have $\Theta = \{trustful, distrustful\}$.

② $F = \{f_1, f_2, ...\}$ describes the evidence set used to distinguish whether the estimation result is trustful. Each element in $F$ denotes an evidence. Here, we have three evidences, i.e., $F = \{f_1, f_2, f_3\}$. $f_1$ indicates an ARSU's credibility, $f_2$ indicates the possibility of data anomalies in an ARSU and $f_3$ indicates the comprehensive evaluation of multiple estimation results. The detailed definitions will be given in Section IV-C2.

③ $BPA$ is a set of basic probability assignment functions. For $j$-th ARSU-ERSU estimation behavior result, $1 \leq j \leq q$, we have

$$\begin{aligned} BPA = \{&m_{f_1}^{j}(trustful), m_{f_1}^{j}(distrustful), \\ &..., m_{f_3}^{j}(trustful), m_{f_3}^{j}(distrustful)\}. \end{aligned}$$

Each element in $BPA$ denotes a probability assignment function of proposition $trustful$ or $distrustful$ based on the evidence $f_1$, $f_2$, or $f_3$ for $j$-th estimation behavior result[4]. The detailed calculations will be given in Section IV-C2.

④ The $BEL$ is a set of belief functions. For $j$-th ARSU-ERSU estimation behavior result, we have $BEL = \{Bel^j(trustful), Bel^j(distrustful)\}$. Each element in $BEL$ denotes a total belief function of proposition $trustful$ or $distrustful$ through fusing three different evidences for $j$-th estimation behavior result. The detailed calculations will be given in Section IV-C3.

⑤ $T$ is a threshold to infer which proposition holds. If $Bel^j(trustful) \leq T$, the $j$-th estimation result is decided to be distrustful. If $Bel^j(trustful) > T$, the $j$-th estimation result is decided to be trustful. Usually, in the experiment, we have $T = 0.5$.

#### *2) Evidence Quantization*

In this paper, for each ARSU-ERSU estimation behavior result, we select the following three evidences, which are quantified as follows.

① Evidence $f_1$ means an ARSU's credibility. It is directly related to an ARSU's incentive score. For $j$-th ARSU-ERSU estimation behavior result, assuming that the initial score of

[3] According to large numbers of tests in the experiment, we find that when $K = 5$, our method can obtain a good performance. The selection of $K$ depends on the real dataset. The main influencing factors are time and speed in the behavior data. Furthermore, speed is affected by time. We find that if the span of timestamps is large and the distribution is sparse, $K$ can be set to a relatively larger value; otherwise, $K$ can be set to a relatively smaller value.

[4] Here, for simplicity, we omit subscript $j$ in notation $BPA$ and the following notation $BEL$.

all ARSUs is $z$, the remaining score of $j$-th ARSU is $z'_j$ and the number of iterations after incentive crowd learning based on reinforcement learning is $k_j$. Usually, the initial score $z$ is set to zero. Based on evidence $f_1$, we have the following probability assignment functions.

When $z = z'_j$, we do not use this evidence.

When $z < z'_j$, the BPA can be calculated as:

$$m^j_{f_1}(trustful) = 1,$$
$$m^j_{f_1}(distrustful) = 0.$$

When $z > z'_j$, the BPA can be calculated as:

$$m^j_{f_1}(distrustful) = \frac{z - z'_j}{k_j},$$
$$m^j_{f_1}(trustful) = 1 - m^j_{f_1}(distrustful).$$

② Evidence $f_2$ means the possibility of data anomalies in an ARSU. A fake ARSU can not obtain real historical behavior data of a vehicle. If an ARSU fabricates some data, multiple similar behavior data in the ARSU will correspond to different estimation results in the side of ERSU. For $j$-th ARSU-ERSU estimation behavior result, assuming that the behavior data in an ERSU are grouped into $K$ clusters in Section IV-B2 ③. And there are maximum $K'_j$ different estimation results, which correspond to the same behavior data (after feature discretization) in the side of the $j$-th ARSU. Then, based on evidence $f_2$, we have the following probability assignment functions.

$$m^j_{f_2}(distrustful) = \frac{K'_j}{K}, (K'_j \le K),$$
$$m^j_{f_2}(trustful) = 1 - m^j_{f_2}(distrustful).$$

③ Evidence $f_3$ means the comprehensive evaluation of multiple estimation results. Here, we use the occupation ratio to obtain this comprehensive evaluation. After once incentive crowd learning based on reinforcement learning, we have $q$ estimation behavior results. Assuming that there are $q'_j$ estimation results which are the same with the $j$-th ARSU-ERSU estimation behavior result. Then, based on evidence $f_3$, we have the following probability assignment functions.

$$m^j_{f_3}(trustful) = \frac{q'_j}{q}, (q'_j < q),$$
$$m^j_{f_3}(distrustful) = 1 - m^j_{f_3}(trustful).$$

### 3) D-S Evidence Fusion

After obtaining above probability assignment functions for three types of evidences, we need use the D-S fusion rule to fuse these evidences and obtain the final belief function $Bel^j(trustful)$ for $j$-th ARSU-ERSU behavior estimation result. Since our two propositions are incompatible, the $Bel$ function can be obtained by:

$$\begin{aligned} Bel^j(trustful) &= m^j_{f_1}(trustful) \oplus m^j_{f_2}(trustful) \\ &\quad \oplus m^j_{f_3}(trustful) \\ &= (m^j_{f_1}(trustful) \oplus m^j_{f_2}(trustful)) \\ &\quad \oplus m^j_{f_3}(trustful). \end{aligned}$$

Furthermore, we have

$$m^j_{f_1}(trustful) \oplus m^j_{f_2}(trustful) =$$
$$\kappa \cdot m^j_{f_1}(trustful) \cdot m^j_{f_2}(trustful),$$

where coefficient $\kappa$ indicates the degree of conflict between evidences,

$$\begin{aligned} \kappa = (1 &- m^j_{f_1}(trustful) \cdot m^j_{f_2}(distrustful) \\ &- m^j_{f_1}(distrustful) \cdot m^j_{f_2}(trustful))^{-1}. \end{aligned}$$

Then, the fusion with the third evidence $f_3$ is the same as the above steps.

After we obtain $Bel^j(trustful)$, we can decide whether the $j$-th behavior estimation result is trustful or distrustful based on the threshold $T$.

## D. Final Verification Based on Behavior Results

For vehicle verification, it includes two parts. The first part is that the agent (ERSU) needs to calculate the differences between the vehicle's arrival behavior and each cluster obtained in the ERSU. The agent makes the vehicle's arrival behavior associate with a cluster having smallest difference. The second part is if the behavior data of this associated cluster is the same with above one of the trustful estimation results, the agent (ERSU) will verify this vehicle as a benign vehicle, and vice versa.

Then, for RSU verification, if a participating ARSU the estimation result of which is decided as trustful, the agent (ERSU) will verify this ARSU as a confidential RSU, and vice versa.

## E. Collusion Avoidance Analysis

In our problem, it is difficult to realize collusion among fake ARSUs. Assuming that there are 1 ERSU and $q$ ($q < m$) ARSUs participating in a verification task. A fake ARSU only knows the current behavior data of the arrival vehicle, it is uneasy to simulate the real historical behavior data of this vehicle. Furthermore, the fake ARSUs cannot obtain the data in the confidential ERSU. So it is almost impossible for the fake ARSUs to fabricate data corresponding to the same behavior data of the ERSU simultaneously. Evidence $f_2$ and $f_3$ will form a disadvantage to the collusion among fake ARSUs. Thus, the fake ARSUs can not make the ERSU obtain the same and false results which occupy the majority of the $q$ estimation results. In the experiment, we will display the difficulty of fabricating correct data for a fake RSU.

## F. Complexity Discussion

We analyze the time complexity of Crowd-Learning verification method. In Algorithm 1, $k$ is the number of iterations, $q$ is the number of ARSUs. Assuming that the size of the data set used in the decision tree after formation of data pairs in Algorithm 2 is $D$. We can obtain the time complexity of Algorithm 1 is $O(k \times (D^2 logD + q))$. In Crowd-Learning verification method, before a vehicle arrives at the next RSU, the next RSU completes behavior estimation for verification ahead of time. When this vehicle arrives, the RSU carries out vehicle verification by comparing the vehicle's behavior directly. Therefore, the Crowd-Learning verification method greatly saves the verification time. According to the experiment in Section V-C, the number of iteration steps and the time of learning are short.

Then as to the data quantities in Crowd-Learning verification method, first, SDN controller selects partial RSUs (not all RSUs) as ARSUs to send historical behavior data of vehicles, based on privacy principle along the route. Second, in Crowd-Learning verification method, the algorithm can control ARSUs to send a moderate amount of data (not endless massive) through incentive scoring policy. And according to the experiment in Section V-B, we will see that Crowd-Learning verification method does not require each selected ARSUs to send a large amount of data to complete the verification. In our following experiment, we quantify one item of behavior data as 39 bytes in the memory. If a vehicle leaves 6 items of behavior data on a RSU on average every day, the RSU only needs 20.08MB of memory approximately to store data of 1,000 vehicles for three months. Besides, in the Q learning iterative process, when the ARSU transmits data of 60 days to the ERSU through 8 iterations on average, an ARSU only needs to transfer 13.71KB data approximately to an ERSU for completing the authentication of a vehicle. If an RSU participates in the verification processes of 100 vehicles at the same time, it needs to transmit 1.34MB data approximately, which is tolerable for the high-speed backbone network between RSUs totally. Therefore, Crowd-Learning verification method will not occupy a lot of bandwidths during the process of data transmission.

### G. Implementation

The computing core of Crowd-Learning verification method is carried out in distributed MEC service stations in IoV, with including reinforcement learning, arrival behavior estimation and conflict decision. The SDN controller is mainly responsible for monitoring the reported verification results coming from RSUs along the driving routes. Besides, the SDN controller has to notify the possible ERSUs to prepare a verification for a forthcoming vehicle, and select some ARSUs to send data according to a vehicle's driving route and specified privacy protection principle.

## V. Experiment and Performance Analysis

In order to verify the reliability and effectiveness of our Crowd-Learning verification method, in this section, we first show the unreality of fabricating correct data for a fake RSU (Section V-A). Then, we verify the rationality of data level division rule mentioned in TABLE I (Section V-B). Finally, we evaluate the performance of Crowd-Learning verification method in terms of convergence, verification latency and verification accuracy (Section V-C).

From OpenStreetMap, we import the OSM map of Songjiang District, Shanghai into the traffic simulation tool SUMO to generate a simulation map. The selected experimental area is depicted by a red rectangle[5] shown in Fig. 6. Then we use SUMO to simulate the traffic scenario within 60 days to produce vehicles' traces. The total number of vehicles is 3400. The vehicles' traces are randomly distributed in this area. The driving speed is between 0 and 120km/h. We set two normal distributions $\mathcal{N}(28800, 720^2)$ and $\mathcal{N}(61200, 720^2)$ to realize two rush hours 7:00-9:00 and 16:00-18:00 when vehicles enter the experimental area in the simulation, where notation $\mathcal{N}$ denotes a normal distribution.

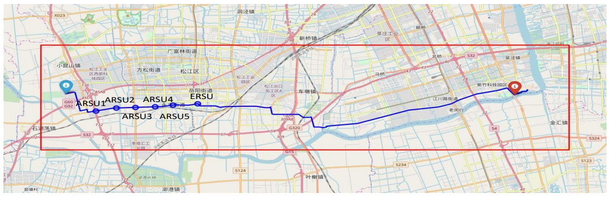


Fig. 6: The experimental area.

Based on above simulation scenario, in Fig. 6, we randomly select a driving route (nearly 31km) depicted by a blue line. The blue buoy represents the starting location and the red buoy represents the end location. Then, based on the privacy protection principle, we select 6 RSUs in the driving route marked by blue circles. We extract historical behavior data of vehicle 'ID truck107' stored in these 6 RSUs to do the following experiments. We set the first 5 RSUs as ARSUs, named ARSU1, ARSU2, ..., ARSU5 in order and set the last RSU as an ERSU, which are all labeled in Fig. 6. The communication range of a RSU is 500m.

### A. Forged Data Analysis

A fake RSU only has the current behavior data of a vehicle. When an ERSU collects data of the vehicle from this fake RSU, the fake RSU can only fabricate some data to send based on the current behavior data. As stated in Section IV-C2 ②, the similar forged behavior data (after feature discretization) in the side of this fake RSU will correspond to different estimation results in the side of ERSU.

First, we use Python to generate some forged data with different fluctuation ranges based on the current behavior data of a certain vehicle. The rule of generating forged data is shown in TABLE III. In the experiment, ARSU5 is assumed as a fake RSU and the current behavior data of 'ID truck107' is $(121.210373^\circ, 31.005605^\circ, 19.7km/h, 92.3^\circ)$ at time $07:34:46$. We generate some forged data of 'ID truck107' based on TABLE III.

Based on above generated data, the forged data analysis is given in Fig. 7. The x-axis represents the number of forged data sent by ARSU5. We use 'item' as the unit of the number of data. The y-axis represents $K_j^{'}$ which has been defined in Section IV-C2 ②. It denotes the maximum number of different estimation results in the ERSU, but in fact, these different estimation results correspond to the same behavior data in the

[5] The position values of four corners of the red rectangle are (121.112566, 31.055786), (121.505105, 31.055786), (121.112566, 30.968119), (121.505105, 30.968119).

TABLE III: Fluctuation Range Settings of Forged Data

| Behavior / Fluctuation | time($s$) | longitude($^{\circ}$) | latitude($^{\circ}$) | speed($km/h$) | direction angle($^{\circ}$) |
|---|---|---|---|---|---|
| Small | $(t \pm 300)$ | $(lon \pm 0.0001)$ | $(lat \pm 0.0001)$ | $(v \pm 5)$ | $(dir \pm 10)$ |
| Medium | $(t \pm 1200)$ | $(lon \pm 0.001)$ | $(lat \pm 0.001)$ | $(v \pm 20)$ | $(dir \pm 90)$ |
| Large | $(t \pm 2400)$ | $(lon \pm 0.01)$ | $(lat \pm 0.01)$ | $(v \pm 50)$ | $(dir \pm 180)$ |

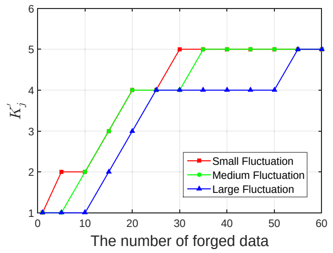


Fig. 7: Forged data analysis.

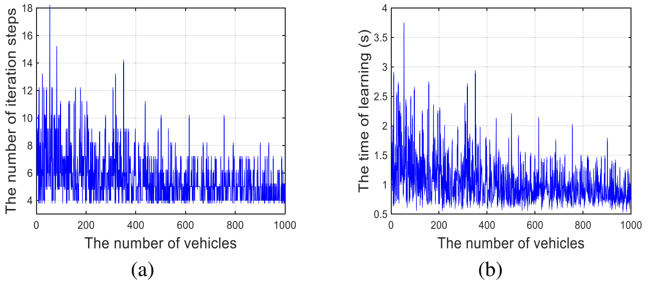


Fig. 9: Convergence test.

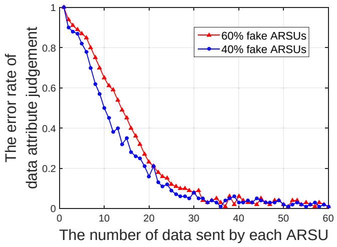


Fig. 8: Data level division rule analysis.

side of the fake ARSU. In this experiment, we select ARSU5 as a fake ARSU, so we have $j = 5$.

From Fig. 7, we can see that the value of $K_j^{'}$ rises with the increase of the number of forged data. The line with small fluctuation range reaches the maximum value 5 when the number of data is 30, which is faster than the two lines with medium fluctuation range and large fluctuation range. It shows that a fake ARSU is easy to be found when it sends forged data with small fluctuation range. As to the forged data with medium and large fluctuation ranges, we need more data to reach the maximum. This is why we use the reinforcement learning to encourage the ARSUs to send more data when doing incentive crowd learning based on reinforcement learning. No matter what kind of forged data a fake ARSU sends, we can always find it in Crowd-Learning verification method through evidence $f_2$.

### B. Data Level Division Rule Analysis

We divide the data quality according to the data level division rule in TABLE I. The data level is related to the data quantity sent by an ARSU and the data attribute determined by D-S evidence theory. In this section, we try to verify the rationality of data level division rule through observing the relationship between the number of data sent by each ARSU and the error rate of data attribute judgement. The relationship is shown in Fig. 8. The change of data attribute will finally affect the judgment of the verification results for those participating ARSUs.

We vary the number of data sent by each ARSU from 0-60 and set 40% and 60% fake ARSUs respectively. The x-axis represents the number of data sent by each ARSU. The y-axis represents the error rate of data attribute judgement. From Fig. 8, we can see that the error rate decreases as the number of data sent by each ARSU increases. But when the number of data is less than 10 items, the error rate exceeds 50%. It proves that too little data will seriously affect the data attribute judgement. If an ARSU sends such data in the process of crowd learning, it will affect the verification result of this ARSU. Therefore, we set the data quantity with less than 10 items as Level 1 in TABLE I.

As the number of data continues to increase, the error rate tends to be stable. This proves that each ARSU must send ‘sufficient’ data to guarantee the verification accuracy of an ARSU. But redundant data will cause a burden on network transmission and MEC calculation. Therefore, we divide data the quantity of which is more than ‘sufficient’ into Level 2.

Besides, the lines in Fig. 8 also reflect that if an agent encourages an ARSU to send data multiple times via reinforcement learning, different number of data sent each time will result in different judgement results for data attributes. So in TABLE I, we need to select $\xi + 1$ data subsets to determine if the data quantity sent by an ARSU is sufficient to support the agent to obtain a stable behavior estimation result. In our experiment, the value of $\xi$ is set to 3. It depends on the real dataset. The larger the $\xi$ is, the more accurate the judgement result is. But, a too large $\xi$ may cause a heavy load on computation.

### C. Performance Evaluation of Crowd-Learning Verification Method

(1) ***Convergence Analysis***

In Crowd-Learning verification method, after completing the verification of a vehicle, the experience an ERSU learned will remain on this ERSU. In the next verifications of other vehicles, the ERSU can learn from the previous experiences to find an optimal scoring policy faster. Next, in verifications of continuous vehicles, we observe the convergence and the

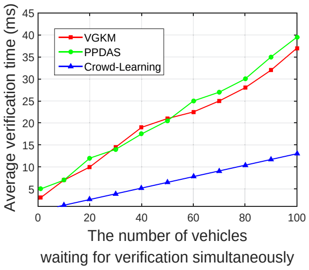


Fig. 10: Verification latency.

role of experiences of Crowd-Learning verification method in terms of iteration steps and the time of learning, which are shown in Fig. 9(a) and Fig. 9(b).

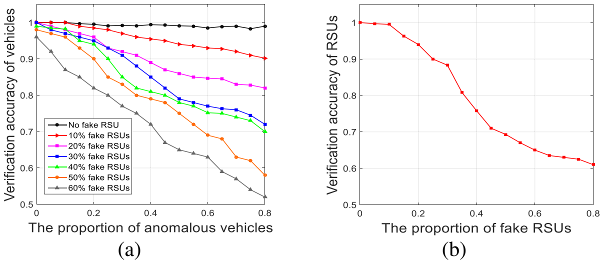
The parameters of Q-learning in Crowd-Learning verification method are set to $\alpha = 0.001$ and $\gamma = 0.8$. As to the $\epsilon$-greedy strategy in Q-learning, we set parameter $\epsilon$ raise gradually with the increase of the learning steps, having $\epsilon = 0.1 + 0.001 \times step$.

From Fig. 9(a) and Fig. 9(b), in once learning, as the number of vehicles that an ERSU has verified increases, the iteration steps and the learning time gradually decrease and tend to be stable within 6 steps and 1s approximately. It proves that the ERSU learns the experience of selecting an optimal scoring policy in continuous learning. This improves the efficiency of Crowd-Learning verification method.

(2) ***Verification Latency Analysis***

This paper is the first one that proposes a behavior based verification method by utilizing crowd to study and make a decision in IoV. There are no similar studies. Here, we choose two recent cryptography based dual authentication methods VGKM [15] and PPDAS [16] in IoV as our comparison algorithms. We use these two methods here to show that the verification latency of our method is no longer than that of cryptography based methods.

In the experiment, we have tested that the verification time of a vehicle in Crowd-Learning verification method is 0.13ms on average. If there are $n$ vehicles waiting for verification in an ERSU, the maximum verification delay does not exceed 0.13ms$\times n$.



(a)

(b)

Fig. 11: Verification accuracy.

Fig. 10 shows the verification latency. The x-axis represents the number of vehicles waiting for verification in an ERSU simultaneously. The y-axis represents the average verification time. From Fig. 10, we can see that the verification time of the three schemes rises with the increase of the number of vehicles. When the number of vehicles reaches up to 100, the verification time of Crowd-Learning verification method is 64.9% less than that of VGKM and 67.5% less than that of PPDAS. So Crowd-Learning verification method shows good performance in reducing the verification latency.

(3) ***Verification Accuracy Analysis***

In this experiment, we test the verification accuracy when there are different proportions of fake RSUs and anomalous vehicles in the network.

Fig. 11(a) shows the verification accuracy of vehicles. The x-axis represents the proportion of anomalous vehicles. The y-axis represents the verification accuracy of vehicles. We inject 0%$\sim$ 60% fake RSUs into the network respectively. We can see that with the increase of the number of fake RSUs, the verification accuracy of vehicles will decrease. It is normal because when a fake RSU performs a verification task as an ERSU, it will verify vehicles arbitrarily. But in reality, the proportion of fake RSUs in the network will not be so large. Our Crowd-Learning verification method can still guarantee a verification accuracy of over 90% approximately when the proportion of fake RSUs is 50% and the proportion of anomalous vehicles is 20%. In particular, when there is no fake RSU, our verification accuracy is above 99%. It shows that Crowd-Learning verification method is effective in terms of vehicles' verifications.

Fig. 11(b) shows the verification accuracy of RSUs. The x-axis represents the proportion of fake RSUs. The y-axis represents the verification accuracy of RSUs. The verification accuracy of RSUs drops as the increase of the proportion of fake RSUs. It is because that a fake RSU may verify other fake RSUs as confidential RSUs in order to avoid exposing itself by reporting false results to the SDN controller. But we can still keep the verification accuracy of RSUs above 94% when the proportion of fake RSUs is 20%, which indicates that Crowd-Learning verification method is effective in terms of RSUs' verifications.

All in all, through above experiments, we can see that Crowd-Learning verification method shows good performances in convergence, verification latency and verification accuracy.

## VI. Conclusion

In this paper, we point out that the problem of identity theft has not been solved in current IoV. A new verification method, called Crowd-Learning, is proposed for the verifications of vehicles and infrastructures in IoV. The SDN controller notifies some RSUs which a forthcoming vehicle may arrive at to prepare the verification for this vehicle in advance. In this way, we can reduce the verification latency effectively. Each MEC stations runs Crowd-Learning verification method in a distributed way. Through reinforcement learning, each RSU can identify the authenticities of passing vehicles and related participating RSUs in once incentive crowd learning based on reinforcement learning. The core of learning is the design of a behavior estimation method and a conflict decision method. Through combining the two methods, the learning agent can obtain a criterion for final behavior correctness decision to verify vehicles and related participating RSUs. The experimental results demonstrate that our method can guarantee high accuracy of verifications for vehicles and RSUs with low latency.

Our method is based on long-term accumulated historical behavior data which already contain a part of behavior fluctuations. In the future, we further consider behavior fluctuation problem. That is, if the behavior of a normal vehicle has never shown in its historical behavior data, we can design a behavior fluctuation range for this user, according to this users geographical proximity and personal route preference. When the behavior of a normal vehicle is in this range, we will judge it as a normal vehicle. If it exceeds the range, we will judge it as a suspicious vehicle. We believe that it is an interesting study.

## References

[1] D. He, S. Zeadally, B. Xu, and X. Huang, "An efficient identity-based conditional privacy-preserving authentication scheme for vehicular ad hoc networks," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 12, pp. 2681–2691, 2015.

[2] H. Ning, H. Liu, and L. T. Yang, "Aggregated-proof based hierarchical authentication scheme for the internet of things," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 3, pp. 657–667, 2015.

[3] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 48–53, 2012.

[4] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Network*, vol. 32, no. 3, pp. 78–83, 2018.

[5] J. Kang, R. Yu, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4660–4670, 2019.

[6] S. Hou, Y. Ye, Y. Song, and M. Abdulhayoglu, "Hindroid: An intelligent android malware detection system based on structured heterogeneous information network," in *Proc. ACM SIGKDD*, 2017.

[7] D. Wei and X. Qiu, "Status-based detection of malicious code in internet of things (iot) devices," in *Proc. IEEE CNS*, 2018.

[8] W. Kai, Y. Hao, Q. Wei, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.

[9] X. Huang, Y. Rong, J. Kang, Y. He, and Z. Yan, "Exploring mobile edge computing for 5g-enabled software defined vehicular networks," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 55–63, 2017.

[10] J. Liu, J. Wan, Z. Bi, Q. Wang, and M. Qiu, "A scalable and quick-response software defined vehicular network assisted by mobile edge computing," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 94–100, 2017.

[11] C. Lyu, D. Gu, Y. Zeng, and P. Mohapatra, "Pba: Prediction-based authentication for vehicle-to-vehicle communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 71–83, 2016.

[12] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu, "Distributed aggregate privacy-preserving authentication in vanets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 3, pp. 516–526, 2017.

[13] J. Shao, X. Lin, R. Lu, and Z. Cong, "A threshold anonymous authentication protocol for vanets," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1711–1720, 2016.

[14] B. Ying and A. Nayak, "Anonymous and lightweight authentication for secure vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 12, pp. 10 626–10 636, 2017.

[15] P. Vijayakumar, M. Azees, A. Kannan, and L. J. Deborah, "Dual authentication and key management techniques for secure data transmission in vehicular ad hoc networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015–1028, 2016.

[16] Y. Liu, Y. Wang, and G. Chang, "Efficient privacy-preserving dual authentication and key agreement scheme for secure v2v communications in an iov paradigm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2740–2749, 2017.

[17] W. Yu, C. G. Yan, Z. Ding, C. Jiang, and M. Zhou, "Modeling and verification of online shopping business processes by considering malicious behavior patterns," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 647–662, 2016.

[18] Z. Nan, K. Bai, H. Hai, and H. Wang, "You are how you touch: User verification on smartphones via tapping behaviors," in *Proc. IEEE ICNP*, 2014.

[19] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, 2013.

[20] A. D. Luca, A. Hang, F. Brudy, and C. Lindner, "Touch me once and i know it's you!: Implicit authentication based on touch screen patterns," in *Proc. ACM CHI*, 2012.

[21] L. Zheng, G. Liu, C. Yan, and C. Jiang, "Transaction fraud detection based on total order relation and behavior diversity," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 3, pp. 796–806, 2018.

[22] H. Wu and G. Liu, "A hybrid model on learning cross features for transaction fraud detection," in *Proc. IEEE ICDM*, 2019.

[23] X. Chen, X. Wu, X. Li, X. Ji, Y. He, and Y. Liu, "Privacy-aware high-quality map generation with participatory sensing," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 719–732, 2016.

[24] X. Liang, Y. Li, G. Han, H. Dai, and H. V. Poor, "A secure mobile crowdsensing game with deep reinforcement learning," *IEEE Transactions on Information Forensics & Security*, vol. 13, no. 1, pp. 35–47, 2018.