IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 July 2023, accepted 28 July 2023, date of publication 3 August 2023, date of current version 9 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3301563

RESEARCH ARTICLE

Trust-Based Reliability Enhancements Provisioning With Resilience Under Information Asymmetry in IoV System

YANFEI LU, GUIYU ZHANG, (Student Member, IEEE), XIAOXUAN WANG, (Member, IEEE), AND XUEHAN LI
School of Electronics and Information Engineering, Beijing Jiaotong University, Beijing 100091, China

Corresponding author: Xiaoxuan Wang (xiaoxuanw@bjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 62102019 and Grant 61931001.

ABSTRACT The advent of the Internet of Vehicles (IoV) has sparked strong scholarly interest in the determination of dependable offloading destinations for tasks. The lack of global information, however, prevents the existing research from being applied in the context of information asymmetry. This paper proposes an original framework for intermediary vehicle-assisted task offloading (IVATO) in scenarios of information asymmetry. Through IVATO, we introduce an intermediary vehicle election mechanism fully based on trust and information mastery. Specifically, a new method is developed to evaluate trust based on resilience. In addition, we conceive a degree of information mastery to measure the amount of information in the vehicle. Based on the intermediary vehicle selected, an objective function for designing the offloading strategy is formulated to maximize both reliability and effectiveness. Proximal Policy Optimization (PPO) based deep reinforcement learning algorithm is adopted to tackle the optimization problem. Simulation results show that the proposed trust evaluation method is more rational than the existing methodologies in the long term. The proposed offloading mechanism shows a 49% increase in utility and a 45% increase in reliability over other schemes.

INDEX TERMS IoV, information asymmetry, intermediary vehicle, resilience, PPO.

I. INTRODUCTION

The Internet of Vehicles (IoV) as a new paradigm has received considerable attention in recent years [1]. IoV is envisaged as an integrated, large-scale network model in an attempt to offer reliable and intelligent services to vehicles [2]. The reliability of IoV, however, may be affected adversely when vehicles incur significant processing delays for local resource limitations as they complete independently computationally intensive tasks generated by those services. To address this issue, the notion of Vehicular Edge Computing (VEC) has been introduced in [3]. In VEC, vehicles may offload the tasks to roadside-units (RSU) or peers with unused resources to expedite the processing [4], [5], [6]. The characteristics of the offloading destination decisively determine whether the tasks can be processed in a timely and reliable manner. Hence, it is vital to determine the efficient and reliable offloading destinations for the task requesting vehicles.

The existing studies on offloading can be broken into two broad categories: the offloading tasks to RSU for execution, and the offloading to other vehicles for execution. Dai et al. [7] construct a multi-user multi-server model and present an improved algorithm to make a VEC server election. A self-learning algorithm is developed in [8] to decide whether the tasks are to be executed by the servers. This approach, however, might cause excessive queue delays when various vehicles offload the tasks to the RSU concurrently. Thus, it is an appealing strategy for the requesting vehicles to offload the tasks to nearby vehicles. Zeng et al. [9] assume that vehicles are able to form a stable resource pool to aid the RSU in executing tasks. Sun et al. [10] study the difficulties of tasks offloading

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Sookhak.

through the vehicular cloud. All these studies hypothesize that RSUs have comprehensive knowledge of every vehicle within their coverage so as to effectively determine the optimal global task scheduling known as information symmetry scenario.

However, as the VEC advances rapidly, the above assumption confronts the following four limitations: (1) Significant communication overhead will be consumed by RSU to collect large amounts of vehicle information [11]. (2) The storage resources of RSUs will be burdened by massive amounts of information [12]. (3) Queuing delays will arise when multiple vehicles request RSU to design an offloading strategy simultaneously [13]. (4) The rising development of RSUs will result in huge economic costs [14]. These four limitations give rise to information asymmetry, a scenario where RSU cannot perfectly primary all the vehicle information within the coverage [15]. How to conduct task offloading in this scenario has sparked strong interest among scholars.

Zhou et al. [16] leverage the multi-armed bandit (MAB) to optimize offloading by only using local information. A contract dependent on the type of vehicle is designed in [17] to recruit vehicles, which effectively incentives idle vehicles to share resources under information asymmetry scenario. While Huang et al. [18] motivate the parked vehicles to achieve efficient workload allocation. Wu et al. [19] take full advantage of the real-time data rate information of the access link to design a reliable task offloading strategy, mitigating the uncertainty of data transfer latency from the edge server to the cloud. Dhelim et al. [20] propose a trust management system to resist large-scale trust attacks. Nevertheless, to primary a vast amount of statistical data in advance is a must for RSU in all these studies, which does not radically alleviate the four above-mentioned limitations. Furthermore, all the research focuses on latency reduction and energy consumption, and the reliability of offloading has drawn scant attention.

Basically, the reliability of task offloading is understood as the efficient completion of tasks within a specified time without being attacked. The reliability of the offloading has immediate impacts on the security of IoV [21]. It is, therefore, necessary to consider reliability in optimizing the offloading process. Liu and Zhang [22] jointly optimize the time delay and dependability of task offloading. Nevertheless, they only evaluate the effect of transmission failure on dependability, while disregarding the influence of processing failure. To tackle this problem, Hou et al. [23] conceive a reliability guarantee mechanism jointly taking the reliability of the communication link and processing node into account. They principally suggest a retransmission method to increase the dependability of work offloading, which, however this has a substantial influence on the timeliness of results. The transmission power and offloading strategy for offloading at Terahertz frequencies are designed in [24]. In their study, reliability is referred to as the probability that the end-to-end delay remains below a certain threshold. In other words, they overlook the influence of the computational node on the reliability of the result, i.e., the node can quickly process a malicious result back.

We fill these gaps by building an original IoV communication architecture under the information asymmetry scenario, an improvement on our previous work [25]. The enhancement is that we specifically use government vehicles with fixed driving routes as intermediary vehicles and divide the RSU coverage into multiple areas based on the driving routes, which makes the model more realistic. In addition, we design an efficient and reliable offloading strategy based on the improved framework, and propose an algorithm with fast convergence characteristics to solve the target problem. To be specific, several intermediary vehicles are introduced as auxiliary roles to aid the RSU in selecting the optimal service vehicles. The intermediary vehicles refer to government vehicles operating on fixed driving routes, which can obtain other vehicles' computing capabilities through interaction. Moreover, electing multiple intermediary vehicles to manage the vehicles of the network can also effectively alleviate the burden on RSU. Furthermore, since each intermediary vehicle is responsible for a portion of requests from all vehicles, the queuing delay problem can be mitigated. In this architecture, a mechanism for electing intermediary vehicles is proposed, considering trust and information mastery degree. Based on the elected intermediary vehicles, a reliable offloading scheme is developed that comprehensively considers the effectiveness and reliability of offloading. By virtue of characteristics of IoV, the deep reinforcement learning (DRL) algorithm Proximal Policy Optimization (PPO) with fast convergence speed is adopted to address the objective optimization problem [26]. Our contributions are summarized as follows:

- We present an improved intermediary vehicle-assisted task offloading (IVATO) scheme. In the proposed model, the intermediary vehicles are exploited to aid RSU in selecting the optimal service vehicles and thus the signaling overhead and storage resources of the RSU can be mitigated.
- We provide an intermediary vehicle election mechanism while considering trust as well as information mastery degree. Specifically, an original method for evaluating trust is proposed inspired by resilience [27], which accurately captures the process of trust's long-term development.
- We design a reliable and effective strategy for task offloading. The reliability takes into account the impact of offloading delays and computational nodes on the results. Considering the characteristics of this framework, the offloading process is divided into two types which are discussed separately. The criteria for classification are whether the requesting vehicle can be directly connected to the service vehicle.
- To solve the optimization problem, we use the PPO algorithm to obtain convergence results quickly and stably. The simulation results verify that our scheme can improve performance dramatically and outperforms the existing ones in utility and reliability.

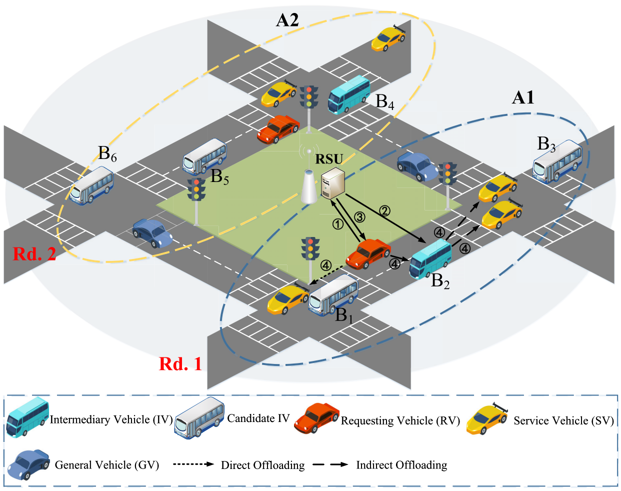

FIGURE 1. The system model of IVATO framework.

The rest of the paper is organized as follows. The system model is proposed in Section II. Section III presents the intermediary vehicle election mechanism based on resilience. The reliable offloading strategy is discussed in Section IV. Section V proposes a task offloading algorithm leveraging PPO. The extensive simulation is shown in Section VI. In Section VII, some conclusions are drawn.

II. SYSTEM MODEL

Information asymmetry refers to the situation where RSU cannot possess all the vehicle information within the coverage due to extreme communication overhead and the burden of storage resources, which makes it difficult to carry out task offloading. In this section, we introduce an original system model under information asymmetry, as shown in FIGURE 1.

A. SYSTEM MODEL DESCRIPTION

Some government vehicles run on fixed routes (e.g., buses) in the urban road network. Based on the social attributes we propose a task offloading mechanism in IVATO [28]. In FIGURE 1, the coverage of RSU is divided into two areas (i.e., $A1, A2$) according to the vehicles' fixed routes. Specifically, some vehicles regularly run on $Rd.1$ (e.g., B_1, B_2, and B_3). $A1$ is the area swept by the buses operating on $Rd.1$ with itself as the center of the circle and the communication range as the radius in the process of driving. $A2$ is formed in a similar way. Assume that $A1$ and $A2$ do not overlap and can fully cover RSU's coverage area. To simplify the problem, we do not consider the movement of vehicles across areas.

Based on the above zoning, we further realize the regional management of vehicles. Take $A1$ as an example. RSU elects a leader from the vehicles with fixed routes periodically (i.e., B_1, B_2, and B_3). The leader, otherwise known as intermediary vehicle, acts as an intermediary between RSU and vehicles in the area. The fixed routes allow the intermediary vehicles to

TABLE 1. Major notations.

Notation	Explanation
D_m	Data size of computation task m
C_m	The number of CPU cycles needed to process the task m
$\overline{P_j}$	Average packet loss rate of the vehicle
E_j	Energy consumption rate of the vehicle
S_j	Packet satisfaction of the vehicle
L_j	Ratio of the vehicles selected as intermediary vehicles
m_i	The time forgetting factor
N_s, N_f	The cumulative effect of behavior records of the vehicle
k^j	The slope of the vehicle's trust
$Trust^j$	The trust value of the vehicle
IMD^j	The information mastery degree of the vehicle
PFV^j	The performance value of the vehicle
R_{ij}, R_{i-IV}	The communication rate between vehicles and between vehicles and the IV
D^m, E^m	The transmission delay and energy consumption of task m.
D^m_{exe}, E^m_{exe}	The execution delay and energy consumption of task m.
V	The time value generated by offloading task m
p^m_j	The unit resource price of s_j for task m
$Cost^m$	The offloading cost of task m
λ_j	The failure rate of s_j
R^m_j	The probability of task m being successfully completed by s_j
U^m_{ij}	The utility of task m to the requesting vehicle

frequently interact with other vehicles in the area to primary their information. In addition, the computing resources of government vehicles are relatively large. Therefore, the intermediary vehicles can design offloading strategies motivated by remuneration. Furthermore, we believe that vehicles use C-V2X(Cellular Vehicle-to-Everything) for communication [3]. We argue that the vehicles in the same area all travel in the same direction and are not subject to external attacks when communicating. Thus, it is assumed the vehicles can establish a stable communication link with each other, and the intermediary and service vehicles can provide stable services to the requesting vehicles. Correspondingly, IVATO effectively alleviates RSU resource constraints by utilizing the off-the-shelf network entity. This is because the intermediary vehicles are responsible for assisting the RSU in designing offloading strategies for vehicles in the area. The RSU only needs to manage the information of vehicles with fixed driving routes and elect the optimal intermediary vehicles.

In FIGURE 1, IVATO mainly consists of the following network entities: requesting vehicles, intermediary vehicles, candidate intermediary vehicles, service vehicles, and the RSU. The details of each entity's duty are described as below:

- *Requesting Vehicle (RV)*: Vehicles that generate tasks for offloading are called requesting vehicles for the lack

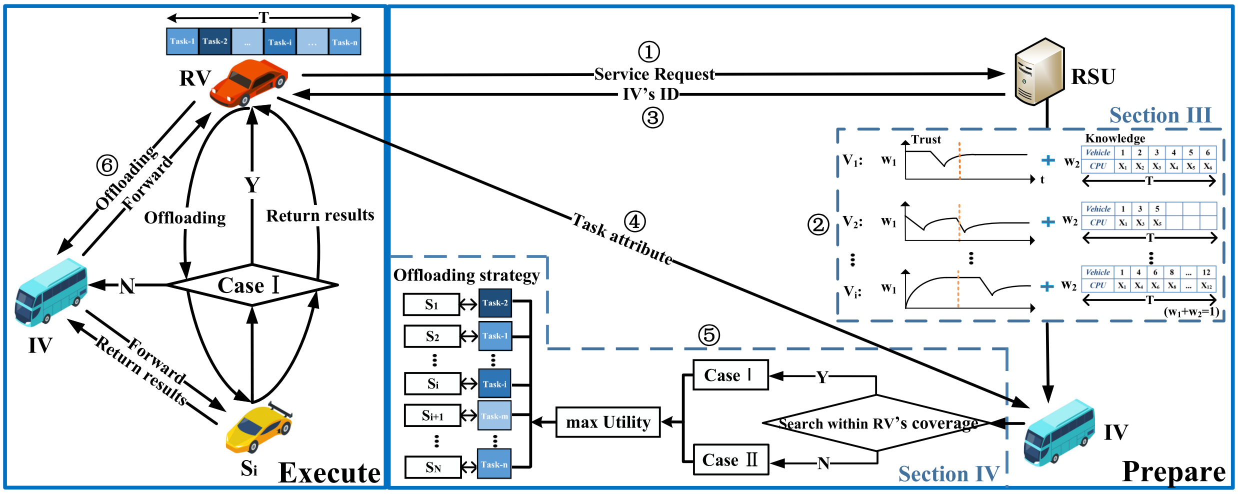

FIGURE 2. **The workflow of IVATO framework.**

of resources. The set of the RV is U^{RV}. Once the RV $r_i \in U^{RV}$ generates a series of tasks within a period time, it sends the task attributes to the IV. The IV designs the offloading strategy, and the RV obtains the information of the SV for each task. The RV offloads each task to the corresponding SV for processing. Further, the RV receives the computation results from SVs after completing the process.

- *Intermediary Vehicle (IV) and candidate IV*: Candidate IVs are vehicles with fixed driving routes. As aforementioned, they can design offloading strategy for the RV. The set of candidate IV can be denoted by $U^{IV} = \{v_1^{IV}, \ldots, v_J^{IV}\}$. To ensure the reliability and the effectiveness of offloading, the IV is both the reliable and informed vehicle elected by RSU from U^{IV}.
- *Service Vehicle (SV)*: The set of the SV is U^{SV}, service vehicle $s_j \in U^{SV}$ is the optimal vehicle selected by IV considering computation capacity and trust. In IVATO, each task needs to be processed by an SV, and an SV processes one task at a time at most. In order to maintain a stable communication connection, we assume that the SV and the RV travel in the same direction.
- *RSU*: RSU documents the behavior information of candidate IVs. As a result, it can assess the reliability and measure the amount of knowledge each potential IV has regarding other vehicles. Hence, it is RSU's duty to elect IVs and occasionally broadcast information on their identities. Moreover, the RSU also needs to conduct tasks that cannot be processed by any SV.

B. SYSTEM FRAMEWORK DESCRIPTION

The system framework is shown in FIGURE 2. The task offloading process of IVATO can be divided into two stages: the preparation stage and the execution stage. The preparation stage mainly includes the election of the IV and the design of offloading strategy. In the execution stage, the RV offloads each task to the corresponding SV according to the offloading strategy. Then it obtains the processing results from SVs. We elaborate on each step as follows:

1) Assume that the task set generated by the RV during period T is $M = \{m_1, \ldots, m_i, \ldots, m_n\}$. The key attributes of the task m_i can be described by a triplet $\{D_i, C_i, \tau_i\}$, where D_i represents the task size, C_i is the required computation resource, and τ_i represents the delay constraint. RV delivers the request to the RSU upon it generates a series of computationally intensive tasks.

2) RSU elects an IV to aid in selecting the SV for each mission. To select the best IV, the RSU takes into account the trust and information mastery degree. The IV elected by these two indicators is both trustful and knowledgeable.

3) RSU broadcasts the identity information of elected IV to the vehicles in the area periodically. Unlike unicast adopting peer-to-peer communication, broadcasting needs to be sent only once for all vehicles to obtain the identity information of the intermediary vehicle, which not only saves bandwidth resources but also reduces the traffic load in the network. In addition, the RSU periodically broadcasts the identity information of the intermediary vehicles, which ensures the freshness of the information and helps the vehicles to keep track of the identity of the high-performance intermediary vehicles in the network.

4) The RV establishes a communication link with the IV. In order to reduce the cost for transmitting information and communication, the RV sends the task attributes of n tasks to the elected IV.

5) The elected IV possesses bulks of vehicle information within its coverage leveraging social attributes. The set of these vehicles is U^{SV}. IV selects an SV from U^{SV} for each task regarding computing capability and trust. Case I refers to that the SV is able to communicate with RV directly. Otherwise, it is regarded as case II. The offloading strategy aims to maximize the utility of the RV.

6) In the execution stage, the tasks can be offloaded directly if the SV is within the RV's coverage (i.e., case I). Otherwise, the task needs to be offloaded by forwarding the IV. The computation results are returned in the same way.

In section III, we elect an optimum IV with an original mechanism. Moreover, Section IV presents the designing of the offloading strategy. For convenience, the main notations used in this paper are summarized in Table 1.

III. INTERMEDIARY VEHICLE ELECTION MECHANISM BASED ON RESILIENCE

The election of intermediary vehicles comprehensively considers trust and the information mastery degree. In this section, we first discuss several measures that have an impact on the computation of trust. Furthermore, resilience is the basis for the long-term development of trust. Finally, we propose the information mastery degree and construct a function to elect the optimal intermediary vehicles.

A. TRUST EVIDENCE COLLECTION

Trust evidence is the component factor for calculating trust, which can be classified into link-based, node-based, data-based, and experienced-based trust evidence. All kinds of trust evidence are recorded and collected for further trust calculation.

1) LINK-BASED TRUST METRIC

The average packet loss rate is an important measure of link dependability. A lower average packet loss rate often suggests a more reliable node.

Several communication links may be established between a vehicle and numerous additional vehicles in a specific statistical period. We assume that each pair of vehicles is limited to having one communication link created in order to simplify the issue. It is assumed that v_j^{IV} has established L communication links with others. Then the average packet loss rate of v_j^{IV} in Δt is calculated as follows

$$\overline{P_j(\Delta t)} = \frac{1}{L}\sum_{l=1}^{L}\frac{n_{l0}}{n_{l0}+n_{l1}}. \tag{1}$$

where n_{l0} and n_{l1} are the numbers of successful and unsuccessful packet transmission of lth link, respectively.

2) NODE-BASED TRUST METRIC

By analyzing the energy consumption from the perspective of the vehicle node's behavior, we use the energy consumption rate to assess trust. Vehicles may consume an excessive quantity of energy when performing malicious attacks within a particular time interval.

Specifically, it is deemed that a vehicle lacks sufficient computational power if its remaining energy is below a predetermined level. The vehicle is therefore thought to have a poor trust since the tasks given to him in this case could not be completed in time, which will bring bad influence to the requesting vehicle. Additionally, this method compares a vehicle's energy consumption rate with that of others to avoid the heterogeneity of the scenarios. In particular, it is suggested that a vehicle is non-malicious if its energy consumption rate is comparable to that of other vehicles in the network. Consequently, the energy consumption rate of v_j^{IV} in Δt can be written as

$$E_j(\Delta t) = \begin{cases} 0, & \text{j} \in U^{RV}, E_{res} < E_{min} \\ exp(-\dfrac{|E_{rate}-\mu_e|}{\theta_e}), & \text{j} \in U^{RV}, E_{res} \geq E_{min}, \end{cases} \tag{2}$$

where E_{res} is the current residual energy, E_{min} is the energy threshold, E_{rate} is the energy consumption rate within Δt, μ_e is the average energy consumption rate, and θ_e is the variance. We can carefully design these four parameters to accurately distinguish between the two situations of low residual energy brought on by malicious attacks and by requesting several tasks.

3) DATA-BASED TRUST METRIC

Another metric for evaluating data quality is packet satisfaction. Not only should the data packets be effectively sent, the confidentiality of the forwarding is also vital. According to [29], the transmitted packets should not be tampered with or eavesdropped. As a result, we propose packet satisfaction to measure how satisfied the receiver is with the packets supplied by the sender. Specifically, a packet that satisfies the receiver is the one that meets the requirements for integrity, consistency, and trustworthiness, all of which can be verified using techniques like digital signatures or hash functions [30].

Packet satisfaction of v_j^{IV} in Δt is denoted by

$$S_j(\Delta t) = \frac{N_s}{N_t}, \tag{3}$$

where N_s and N_t are the number of packets the receiver is satisfied with and the number of received in Δt, respectively.

4) EXPERIENCE-BASED TRUST METRIC

The punishment and trust-rebuilding process are also influenced by the vehicle's prior experiences. Specifically, an IV's repeated malicious behavior will have adverse impacts on the network. To avoid such problems, it is necessary to increase the penalties and lower the recovery rate for these vehicles.

We introduce the ratio of the number of v_j^{IV} selected as IV N_L to the number of activities in the network N_P by t_i as

$$L_j(t_i) = \frac{N_L}{N_P}. \tag{4}$$

Higher values of $L_j(t_i)$ suggest that a vehicle is frequently chosen as IV and is more dependable. According to what was previously said, $L_j(t_i)$ should be regarded as a penalty component to cause the trust to decline more rapidly and recover slowly.

In this paper, we examine the behavior of the vehicle in terms of time slots Δt. Collectively, $\overline{P_j(\Delta t)}$, $E_j(\Delta t)$, and $S_j(\Delta t)$ are used to assess a vehicle's performance in a certain time interval. We employ $L_j(t_i)$ as a punishment component effecting the trust change.

B. TRUST CALCULATION AND EVOLUTION USING RESILIENCE

A large body of existing work adopts fuzzy algorithms or subjective logic to derive the change in trust over a period while ignoring the long-term impacts of the vehicle's experience and the malicious behavior [31], [32], [33], [34]. To tackle this issue, the presented method blends the idea of resilience with the quantitative evaluation [35] to characterize the development of trust through time.

The increase or decrease in trust is subject to the vehicle's recent actions, either positive or negative. The trust increases in the current time slot if the vehicle performs positively in the prior time slot. Conversely, the vehicle's poor performance during the previous period caused a decline in trust. In order to highlight the issue, we use a two-tuple $\left\{I_f^j(t_i), I_s^j(t_i)\right\}$ to indicate whether v_j^{IV} perform well or poorly in time slot t_i. In detail, if the vehicle v_j^{IV} behaves positively, then at the end of t_{i-1}, the two-tuple can be denoted by $\left\{I_f^j(t_{i-1}) = 0, I_s^j(t_{i-1}) = 1\right\}$. Otherwise, it can be expressed as $\left\{I_f^j(t_{i-1}) = 1, I_s^j(t_{i-1}) = 0\right\}$. Whether v_j^{IV} behaves well has links to $\overline{P_j(\Delta t)}$, $E_j(\Delta t)$, and $S_j(\Delta t)$. Three binary variables are matched by these three indicators. Below are the precise formulas

$$PLR^j(t_i) = \begin{cases} 1, & 0 < \overline{P_j(\Delta t)} < \eta_0, \\ 0, & \text{otherwise}, \end{cases} \tag{5}$$

$$EC^j(t_i) = \begin{cases} 1, & 0 < E_j(\Delta t) < \eta_1, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

$$PS^j(t_i) = \begin{cases} 1, & 0 < S_j(\Delta t) < \eta_2, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where η_0, η_1, and η_2 are the thresholds of $\overline{P_j(\Delta t)}$, $E_j(\Delta t)$, and $S_j(\Delta t)$, respectively.

Only when all three of the aforementioned measures are within the permitted limits at the same time do vehicles qualify as behaving positively. Accordingly, the behavior record of v_j^{IV} in time slot t_i can be stated as follows

$$I_s^j(t_i) = PLR^j(t_i) \cdot EC^j(t_i) \cdot PS^j(t_i), \tag{8}$$

$$I_f^j(t_i) = 1 - I_s^j(t_i). \tag{9}$$

Furthermore, building trust is a long-term process of accumulation. Given the timeliness of interaction data, the recent occurrences have a higher influence on the trust rating, and vice versa. Thus, in order to quantify trust, we specify the time forgetting factor. If an interaction record happens at t_i and the current time slot is t, the impact factor of this interaction record may be stated as follows:

$$m_i = \rho^{t_i - t}, \tag{10}$$

where ρ is the predefined parameter. The cumulative effect of behavior records of v_j^{IV} is formulated as

$$N_s^j(t) = \sum_{t_i=0}^{t} \rho^{t_i - t} I_s^j(t_i), \tag{11}$$

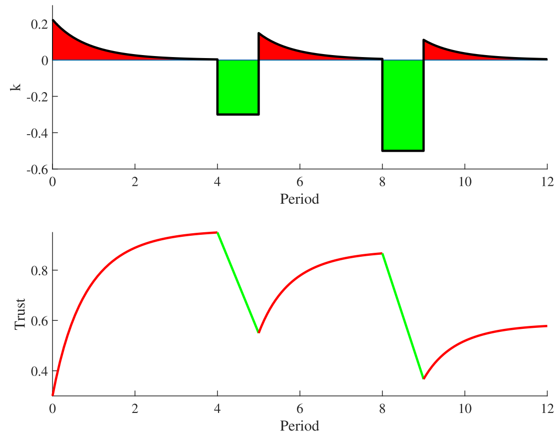

FIGURE 3. Resilience-based trust value evolution process.

$$N_f^j(t) = \sum_{t_i=0}^{t} \rho^{t_i - t} I_f^j(t_i). \tag{12}$$

As stated above, the trust value will increase at t_i if $\left\{I_f^j(t_{i-1}) = 0, I_s^j(t_{i-1}) = 1\right\}$ is gained at t_{i-1}. As shown in the red region in FIGURE 3, the slope k declines as the number of good conduct rises and eventually approaches 0, which is similar to [31], [32], and [33]. The trust recovery rate may thus be represented as follows

$$k^j(t_i) = (-1)^{I_f^j(t_{i-1})} \cdot \frac{\alpha}{1 + \sum_{t=0}^{t_i} I_f^j(t)} \cdot \left[exp\left(\frac{N_f^j(t)}{N_s^j(t) + N_f^j(t)} \right) - 1 \right], \tag{13}$$

where α is the adjustment factor.

Conversely, if $\left\{I_f^j(t_{i-1}) = 1, I_s^j(t_{i-1}) = 0\right\}$, the trust will decline. To improve vehicles' reliability, we will increase the penalties for malicious behavior as it accumulates. Therefore, as indicated by the green region in FIGURE 3, the decrease rate of trust increases as the number of malicious behavior accumulates. Integrating with penalty factor $L_j(t_i)$, the slope k of the decline in trust can be expressed as

$$k^j(t_i) = (-1)^{I_f^j(t_{i-1})} \cdot \beta \cdot [(L^j(t_{i-1}) + 1) \cdot N_f^j(t)], \tag{14}$$

where β is the predefined factor.

To sum up, the slope of the trust value of v_j^{IV} at t_i is written as

$$k^j(t_i) = (-1)^{I_f^j(t_{i-1})} \Big\{ I_f^j(t_{i-1}) \cdot \beta \cdot [L^j(t_{i-1}) + 1] \cdot N_f^j(t_{i-1}) + I_s^j(t_{i-1}) \Big[\frac{\alpha}{1 + \sum_{t=0}^{t_i} I_f^j(t)} \cdot exp\Big(\frac{N_f^j(t_{i-1}) + 1}{N_f^j(t_{i-1}) + N_s^j(t_{i-1}) + 2} - 1 \Big) \Big] \Big\}, \tag{15}$$

and the trust of v_j^{IV} at t is given by

$$Trust^j(t) = \int_0^t k^j(t_i)\, dt_i. \tag{16}$$

As aforementioned, as the positive behavior continues, the rate of trust recovery declines. Therefore, compared to the follow-up, the recovery rate produced by the positive behavior that occurs right after the negative behavior is the highest. We refer to this rate as the resilient recovery rate, which may be seen, for instance, in the slopes on the 0^{th}, 5^{th}, and 9^{th} periods in FIGURE 3. The resilient recovery rate declines as the negative behavior increases in order to prevent the malicious behavior and strengthen the network's dependability. We use the resilient recovery rate to assess if the trust evolution process has reached its end. This is because each vehicle may be assigned a different initial trust when entering the network. Terminating the evolution process by evaluating the trust value is unfair. Particulary, the trust evolution process ceases, and v_j^{IV} is placed on the blacklist when the resilient recovery rate of v_j^{IV} falls below the threshold η_{rate}.

C. THE CONSTRUCTION OF ELECTION FUNCTION FOR INTERMEDIARY VEHICLES

1) INTERMEDIARY VEHICLE ELECTION INDICATORS

Given the duty of IV, the elected IV should be dependable and informed. Thus, we elect the IV comprehensively considering the trust and the information mastery degree. We introduce these two indicators as follows:

- Trust: The possibility that a vehicle will be chosen as an IV increases as its trust rises. Vehicles on the blacklist, on the other hand, will no longer participate in the election of IV. The trust value of a vehicle at a specific instant t can be obtained in accordance with the preceding section.
- Information Mastery Degree: The IV is responsible for aiding the RSU in determining the optimal service vehicle. For this purpose, the IV should get as much knowledge as they can about others. Similar to actual social networks, a vehicle's knowledge of another person depends on whether they have ever interacted. Therefore, as shown in FIGURE 2, we establish a historical interaction record table that is updated at the end of each time slot. Newly interacting cars will be added to the table, while previously interacting vehicles will overwrite entries when they re-interact.
  As the moment t, the number of vehicles for which v_j^{IV} possesses information can be denoted by

$$C^j(t) = \sum_{i=1}^{K} I_{i,j}, \tag{17}$$

  where K is the number of items in the set U^{SV}, $I_{i,j}$ is a binary variable. If v_j^{IV} has interacted with v_i^{SV}, then $I_{i,j} = 1$. Otherwise, $I_{i,j} = 0$.
  The ratio of the number of interacted vehicles to the average interaction of various other vehicles for v_j^{IV} represents the information mastery degree and can be calculated as follows

$$IMD^j(t) = \frac{C^j(t)}{\overline{C^j(t)}}, \tag{18}$$

  where $\overline{C^j(t)} = \frac{1}{J} \sum_{v_j^{IV} \in U^{IV}} C^j(t)$. J is the total number of elements in the set U^{IV}.

2) INTERMEDIARY VEHICLE ELECTION FUNCTION

Combining the above two indicators, the performance value of v_j^{IV} is denoted by

$$PFV^j(t) = w_1 \cdot Trust^j(t) + w_2 \cdot IMD^j(t), \tag{19}$$

where w_1 and w_2 are the factor weight and $w_1 + w_2 = 1$. The vehicle with the highest PFV value should be picked as IV, which can be written as

$$\max_{v_j^{IV} \in U^{IV}} \{w_1 \cdot Trust^j(t) + w_2 \cdot IMD^j(t)\}. \tag{20}$$

IV. RELIABLE OFFLOADING STRATEGY BASED ON ELECTED IV

As described above, the elected IV need to assist the RSU in designing the offloading strategy. In this section, we propose a scheme of a reliable offloading strategy. In detail, there are two procedures to execute the task processing (i.e., communication and computation), which will be presented in the following section.

A. COMMUNICATION MODEL

The communication model can be divided into two cases according to whether the RV is directly connected to the SV. Let binary variable u_{ij} denote whether requesting vehicle r_i can establish a direct connection with service vehicle s_j, which is measured by

$$u_{ij} = \begin{cases} 1, & r_i \textit{ can communicate with } s_j \textit{ directly} \\ 0, & \textit{otherwise}. \end{cases} \tag{21}$$

Then we propose corresponding communication schemes for the two cases. To simplify the problem, we assume the system to be quasi-static. In the following, we explain the two cases in detail.

1) CASE I

While $u_{ij} = 1$, task generated by r_i can be offloaded directly to s_j. The communication rate between r_i and s_j is given by

$$R_{ij} = B_{ij} \cdot \log_2(1 + \frac{p_{i-j} \cdot h_{ij}}{\vartheta^2}), \tag{22}$$

where B_{ij} represents the available bandwidth of the communication link between r_i and s_j. p_{i-j} and h_{ij} are the transmission power of r_i and the channel gain, respectively. In addition, ϑ^2 is the noise power.

The transmission delay of task m can be calculated by

$$D_{i,j}^m = \frac{D_m}{R_{i,j}}. \tag{23}$$

The energy cost of the transmission process is expressed as

$$E_{i,j}^m = p_{i-j} \cdot D_{i,j}^m. \tag{24}$$

2) CASE II

While $u_{ij} = 0$, tasks cannot be offloaded directly to s_j. In this case, since the IV can be connected with both r_i and s_j, the IV assists in forwarding tasks. Specifically, r_i offloads the tasks to IV first, then the IV forwards the tasks to s_j. In turn, the process of returning results is reverse.

The achievable data rate from r_i to the IV and from the IV to s_j are denoted by

$$R_{i-IV} = B_{i-IV} \cdot \log_2(1 + \frac{p_{i-IV} \cdot h_{i-IV}}{\vartheta^2}), \tag{25}$$

$$R_{IV-j} = B_{IV-j} \cdot \log_2(1 + \frac{p_{IV-j} \cdot h_{IV-j}}{\vartheta^2}), \tag{26}$$

where B_{i-IV} represents the available bandwidth of the communication link between r_i and the IV. p_{i-IV} and h_{i-IV} are the transmission power of r_i to the IV and the channel gain, respectively. B_{IV-j} represents the available bandwidth of the communication link between the IV and the s_j. p_{IV-j} and h_{IV-j} are the transmission power of IV to the s_j and the channel gain, respectively. In addition, ϑ^2 is the noise power.

Therefore, ignoring the queuing delays at the IV, the total transmission delay of task m can be denoted by

$$D^m_{i-IV-j} = D^m_{i-IV} + D^m_{IV-j} = \frac{D_m}{R_{i-IV}} + \frac{D_m}{R_{IV-j}}, \tag{27}$$

The energy consumption of transmission E^m_{i-IV-j} is expressed as

$$E^m_{i-IV-j} = p_{i-IV} \cdot \frac{D_m}{R_{i-IV}} + p_{IV-\mathrm{j}} \cdot \frac{D_m}{R_{IV-j}}. \tag{28}$$

In conclusion, the energy consumption of the transmission can be expressed collectively as

$$E_{trans} = u_{ij} \cdot E^m_{i,j} + \left(1 - u_{ij}\right) \cdot E^m_{i-IV-j}. \tag{29}$$

B. COMPUTATION MODEL

We use a binary variable x^m_j to represent the offloading strategy. In detail, while $x^m_j = 1$, the task m is offloaded to the service vehicle s_j. While $x^m_j = 0$, the task m will not be offloaded to the s_j.

If $x^m_j = 1$, the execution time required to process the task m is calculated as

$$D^m_{exe} = \frac{C_m}{f_j}, \tag{30}$$

where f_j denotes the processing capability (i.e., the amount of CPU frequency in cycles/s) at s_j assigned for computing.

Consequently, the energy consumption of s_j for executing the task m is expressed as

$$E^m_{exe-j} = \kappa_j \cdot C_m \cdot (f_j)^2, \tag{31}$$

where κ_j is a coefficient related to power in s_j.

Therefore, the total latency for processing task m is denoted by

$$T^m_{total-j} = u_{ij} \cdot D^m_{i,j} + \left(1 - u_{ij}\right) \cdot D^m_{i-IV-j} + x^m_j \cdot D^m_{exe-j}. \tag{32}$$

Furthermore, the total energy consumption for processing task m is expressed as

$$E^m_{total-j} = u_{ij} \cdot E^m_{i,j} + \left(1 - u_{ij}\right) \cdot E^m_{i-IV-j} + x^m_j \cdot E^m_{exe-j}. \tag{33}$$

C. PROBLEM FORMULATION

Based on the given communication and computation model, we can obtain the delay and energy consumption caused by offloading, which are the component of the optimization problem. In general, the tasks are anticipated to be processed timely, efficiently, and reliably, i.e., with small time delay, energy consumption, and a high probability of successful completion. Therefore we define the time value function, resource pricing method, and reliability function to jointly design the offloading strategy.

- *Time Value Function*: To ensure a timely offloading process, we introduce a novel concept named time value function by comparing offloading delay with local completion time. Specifically, by offloading the tasks, the processing time will be less than if they are executed locally. Besides, the less processing time, the better the user experience. Therefore, the time value function quantifies the value of time saved by offloading. As aforementioned, we model the time value function as a decreasing function of the time, which is given by

$$V\left(T^m_j, T^m_{local}\right) = \begin{cases} \mathrm{e}^{-\gamma \cdot \left(T^m_j - T^m_{local}\right)} - 1, & \text{if } T^m_j \leq T_{local}, \\ 0, & \text{otherwise}, \end{cases} \tag{34}$$

  where $T^m_{local} = \frac{C_m}{f_{local}}$ represents the local completion time, f_{local} denotes the local processing capability, γ is the adjustment factor. In such a case, the time value exponentially decreases over time.
- *Resource Pricing Method*: Due to the vehicles' selfish nature, both the RV and the SV expect to minimize their cost and maximize their income, so the RV needs to pay the SV for the task offloading service. The SV determines its unit resource price based on energy consumption and computing delay. Specifically, if the energy consumption of service vehicle s_j is high, the price of computing resources is large. Moreover, if the offloading computing delay is low, the price of computing resources is also large. Therefore, the unit resource price for s_j to complete task m is denoted by

$$\mathrm{p}^m_j = a \cdot \frac{\left(T_{total-j}\right)_{\max}}{T^m_{total-j}} + b \cdot \frac{E^m_{total-j}}{\left(E_{total-j}\right)_{\max}}, \tag{35}$$

  where a and b are the factor weight. As a result, the overhead that the RV r_i needs to pay to the SV s_j for task m can be written as

$$Cost^m_{ij} = C_m \cdot p^m_j. \tag{36}$$

- *Reliability Function*: The reliability of task m depends on the reliability of the service vehicle. This paper employs a widely accepted reliability evaluation method that calculates the failure probability statistically. The failure refers to the failure of the SV to complete the task successfully. To put it another way, each failure case can be transformed into a Poisson distribution driven by a failure rate that can be determined by a specific value. We use $\lambda_j(T_0)$ to denote the failure rate of service vehicle s_j at T_0, which can be expressed as

$$\lambda_j(T_0) = e^{-\eta \cdot Trust^j(T_0)}, \tag{37}$$

where η is the coefficient. $Trust^j(T_0)$ is the trust value of s_j at T_0, which can be obtained according to Sec. III-B. Consequently, based on the knowledge of Poisson process, the probability of task m being successfully completed (i.e., reliability function of task m) can be written as

$$R_j^m = x_j^m \cdot e^{-\lambda_j(T_0) \cdot D_{exe}^m}. \tag{38}$$

Based on Eq. (36)-(38), the utility of requesting vehicle r_i for offloading task m can be denoted by

$$U_{ij}^m = \sum_{j=1}^{J} x_j^m \cdot R_j^m \cdot \left[V(T_j^m, T_{local}^m) - E_{trans} - Cost_{ij}^m \right], \tag{39}$$

and the formalized problem is described as

$$\max_{x_j^m} \sum_{m=1}^{M} x_j^m \cdot U_{ij}^m \tag{40}$$

$$\text{s.t. } x_j^m \in \{0, 1\} \tag{40a}$$

$$\sum_{j=1}^{J} x_j^m = 1 \tag{40b}$$

$$\sum_{m=1}^{M} x_j^m \leq 1 \tag{40c}$$

$$p_{i-j}, p_{i-IV} < P_{\max} \tag{40d}$$

$$T_j^m < \tau_m. \tag{40e}$$

where P_{max} is the maximum transmission power of r_i. Constraint (40a) indicates that offloading strategy x_j^m is a binary variable. Constraint (40b) implies that each task needs to be offloaded to a service vehicle for processing. Constraint (40c) indicates that each service vehicle can process one task at most. Constraint (40d) represents the r_i transmission power limitation. Constraint (40e) represents the limitation of task completion time.

The objective problem is an integer nonlinear programming problem. Because the number of tasks and the set of service vehicles are fixed, the number of solutions is finite, so there exists an optimal solution. Due to model solidification, traditional optimization algorithms lack the ability to learn actively, limiting environmental adaptability and scalability. Therefore, we use a deep reinforcement learning algorithm to tackle it in Section V.

V. PPO BASED DRL ALGORITHM FOR TASK OFFLOADING

In this section, we first introduce the basic knowledge of Proximal Policy Optimization (PPO) algorithm, and then demonstrate the state space, the action space, and the reward function of the formulated Markov Decision Process (MDP). Finally our proposed algorithm is presented in detail.

In IVATO, we assume that a RV generates a series of tasks to offload in a short amount of time. The agent in IV decides the service vehicle for each task in each slot. Thus, the task offloading problem is regarded as an optimal sequential decision-making problem. Since the previous action affects the distribution of computing resources, which in turn affects the subsequent actions, we cannot make decisions only based on the current observed state. In this regard, the problem can be modeled as an MDP. After casting the problem as an MDP, we adopt a model-free deep reinforcement learning algorithm named PPO to solve this problem. The advantages of PPO compared to other DRL algorithms lie in: (1) Reducing model complexity; (2) Avoiding convergence to local optimum; (3) Accelerating convergence speed.

PPO alternates between maximizing a "surrogate" objective function via stochastic gradient ascent and sampling data by interacting with the environment. As opposed to the conventional policy gradient approaches that only perform one gradient update per data sample, the PPO suggests a novel objective function that can accomplish tiny batch updates in several training steps. This resolves the issue with the classic Policy Gradient algorithm's difficulty in identifying the step size. Furthermore, PPO adds the off-policy learning policy to address the issue that on-policy (training and sampling under the same policy) updates typically lead to a local optimum. Thus, PPO has better sample complexity and is therefore considerably easier to implement.

The procedure of PPO is shown in FIGURE 4. Generally, the implementation of the PPO algorithm involves three networks, including one critic network, and two actor networks (Actor_old and Actor_new). Actor_old network is responsible for interacting with the environment and collecting data, while Actor_new samples the data collected by Actor_old and updates the policy. Thus, the network that interacts with the environment is not the same as the network that updates the policy. In this case, multiple uses of the sampled data can be achieved, which is the core of off-policy. Because Actor_old interacts with the environment instead of Actor_new, the gap between the two networks should not be too large, so the PPO algorithm introduces the important ratio to correct the objective function [26], which can be denoted by

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}. \tag{41}$$

When updating the policy network (i.e., Actor Network), considering the above probability ratio, the widely used form is given in [36]

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right]. \tag{42}$$

The clip function $clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)$ constrains the value of r_t, which removes the incentive for moving r_t outside the interval $[1-\varepsilon, 1+\varepsilon]$, with ε being a hyperparameter to control the clip range. The final target is constrained as a lower bound to the unclipped objective by taking the

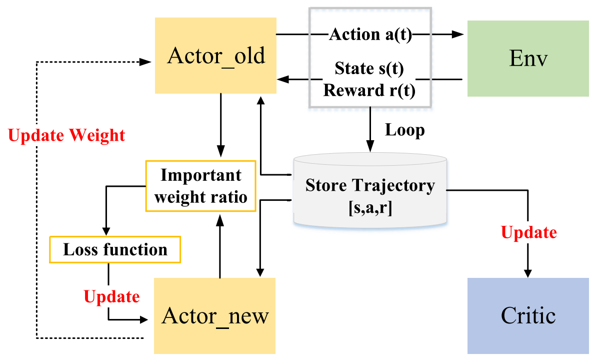

FIGURE 4. Structure of PPO based task offloading algorithm.

minimum of the clipped and unclipped objectives. $\hat{A}_t$ can be obtained by a given length T trajectory segment in [37]

$$\hat{A}_t = -V(s_t) + r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t+1} r_{T-1} + \gamma^{T-t} V(s_T), \tag{43}$$

where $r(t)$ is the reward function, $V(S_T)$ is the state-value function, γ is the discount factor. We update the Actor_new parameters by optimizing the loss function and at the end of the cycle, the Actor-old network is updated with the actor-new network weights.

If a neural network is employed to design the parameters sharing between the policy and value function, a loss function should be constructed that combines the policy surrogate and a value function error term [36], [37]. This aim may further be strengthened by providing an entropy bonus to guarantee adequate exploration, as recommended in earlier work. Combining these terms, we have the following critic network objective

$$L_t^{CLIP+VF+S}(\theta) = \hat{E}_t \left[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t) \right], \tag{44}$$

where c_1, c_2 are coefficients, and S denotes an entropy bonus, and L_t^{VF} is a squared-error loss $\left(V_\theta(s_t) - V_t^{t\arg}\right)^2$.

A. STATE SPACE

In each slot, the agent in IV observes the vehicular environment and gathers the following parameters:

- $H_{i-j}(t)$: The channel gain between the r_i and the s_j at time t.
- $H_{i-IV}(t)$: The channel gain between the r_i and the IV at time t.
- H_{IV-j}: The channel gain between the IV and the s_j at time t.
- $T_j(t)$: The trust of the service vehicle s_j at time t.
- $F_j(t)$: The remaining computing resource of s_j at time t. For simplicity, we assume that the remaining computing resources of the service vehicle remain unchanged during the task execution. In addition, the service vehicle uses all the remaining resources to process the offloading task.
- $D_m(t)$, $C_m(t)$, $\tau_m(t)$: The data size, the computation size, and the time to live for the task m at time t, respectively.

We use $\boldsymbol{S}$ to denote the state space, and the state vector at time t is:

$$\begin{aligned} \mathbf{s_t} = [&H_{i-1}(t), H_{i-2}(t), \ldots, H_{i-J'}(t), H_{i-IV}(t), \\ &H_{IV-1}(t), H_{IV-2}(t), \ldots, H_{IV-J'}(t), \\ &T_1(t), T_2(t), \ldots, T_J(t), \\ &F_1(t), F_2(t), \ldots, F_J(t), D_m(t), C_m(t), \tau_m(t)]. \end{aligned} \tag{45}$$

B. ACTION SPACE

As described in the previous section, the IV needs to find the SV for each task to maximize the RV's global utility. Therefore, the action of the IV is the service vehicle selection.

In an episode, the IV needs to take n steps, (i.e., select a service vehicle for each task m respectively). In detail, the action of IV at the mth step can be expressed as

$$\mathbf{a_m} = [x_1(m), x_2(m), \ldots, x_J(m)], \tag{46}$$

where $x_j(m)$ is a binary variable, $x_j(m) = 1$ manifests that the task m is processed by s_j.

C. REWARD FUNCTION

In each step, the agent take action $\mathbf{a_m}$ by observing state $\mathbf{s_t}$, and then gets an immediate reward, which can be denoted by $R(\mathbf{s_t}, \mathbf{a_m}) = U_{ij}^m$, where, U_{ij}^m indicates the utility of the task in Eq. (39).

In order to make each service vehicle perform one task at most, we introduce a penalty when designing the reward function. The reward function can be written as

$$R(t) = \begin{cases} U_{ij}^m, & \textit{if } s_j \textit{ is assigned only one task} \\ -\Gamma, & \textit{otherwise}, \end{cases} \tag{47}$$

where Γ is a large positive real number.

D. ALGORITHM DESIGN BASED ON PPO

The proposed approach is summarized in **Algorithm1**. We elaborate on the primary steps of the algorithm as follows:

1) First, we randomly initialize the network learning rate l, the clip ratio ε and the batch size $\mathcal{B}$. In addition, we set the experience replay buffer B as $\emptyset$. The parameter of actor-value network is also initialized. For the sake of performance evaluation, we initialize the channel gain, trust value, remaining computing resources, etc.

2) In each episode, the initial state is firstly obtained by the agent. Once the agent receives the offloading request from the RV, state s_t can be estimated by gathering state information from the driving environment. Additionally, the agent carries out the action in accordance with the state s_t and policy network $\pi_\theta(a_t|s_t)$.

3) Given action a_t, the agent sends service assignment information to the RV (i.e., which SV process the task m). Then the assigned SV sends the allocation information back to the agent, and the agent calculates the reward r_t. At last, the RV transmits the task m to the SV.

4) The agent observes the vehicular environment and forecasts the next state s_{t+1}. Loop executes an episode. The agent stores the trajectory $\tau = \{s_t, a_t, r_t\}_{t=1}^M$ into buffer.

Algorithm 1 Joint Task Offloading Scheduling

Input: The state space shape s

Output: The optimal offloading decision.

Initialize: Randomly initialize the actor network l, the clip ratio ε and the batch size $\mathcal{B}$. Set the experience replay buffer B as $\emptyset$
for episode $=1,2,\ldots,L$ **do**
  for a step t $=1,2,\ldots,n$ **do**
    The agent executes action according to $\pi_{\theta_{\text{old}}}(a_t|s_t)$
    Get the reward r_t, and the next state s_{t+1}
    Update state $s(t)=s(t+1)$
    step+1
  end for
  Get a trajectory: $\tau=\{s_t,a_t,r_t\}_{t=1}^{n}$ and store it into B
  Compute advantage function $\left\{\hat{A}_t\right\}_{t=1}^{n}$ using (43)
  for $k=1,2,\ldots,n$ **do**
    Select B group of data $\left\{s_t,a_t,r_t,\hat{A}_t\right\}$
    Calculate the gradient of formula (44) and then update the critic network by Adam
    Update new actor network parameter by (42)
    $\theta_{old}\leftarrow\theta$
    $k+1$
  end for
  episode+1
end for

5) The agent computes the advantage function $\left\{\hat{A}_t\right\}_{t=1}^{M}$ according to Eq. (43).

6) According to the trajectory information and the advantage function in the buffer, we calculate the gradient of the objective function of the actor network and critic network to update the network parameters.

7) The above network interacting with the environment is the Actor_old. The Actor_new network updates the strategy using the data collected by Actor_old. After certain steps, the cycle ends and the Actor_old network is updated with Actor_new network parameter.

8) Finally, when the loss function is smaller than a certain threshold, the algorithm converges and the optimal task unloading strategy is obtained.

VI. SIMULATION RESULTS

In this section, we verify the advantages of IV election and offloading strategy mechanisms via simulations. We evaluate the performance of the developed IV election mechanism before demonstrating the effectiveness and reliability of offloading strategy.

Experiments are conducted on a GPU-based server with 3 NVIDIA Tesla T4 GPUs, where the CPU is Intel(R) Xeon(R) Silver 4210R CPU, 2.40GHz basic frequency. The software platform used to conduct the simulations are Matlab 2021 and Python 3.7.6. For PPO, the open-sorce implementation provided by OpenAI is used. To apply PPO to the target problem, we write the environment profile according to Section V. In addition, we implement the proposed algorithm using Tensorflow-GPU 2.3.0. All parameters used are summarized in TABLE 2, which reference the literature [8], [23], [38], and [39]. The hyper-parameters for training deep neural networks are added to TABLE 3.

TABLE 2. Simulation parameters.

Parameter	Value
Time forgetting factor(ρ)	4
Number of candidate IV (J)	20
Communication probability (p)	0.5
Number of vehicles (K)	2∼14
Transmission power of RV (P_{RV})	1w
Transmission power of IV (P_{IV})	1.2w
Bandwidth of channel (B)	0.5 MHz
Number of tasks (n)	2∼8
Data size of task (D)	2∼8 Mbit

TABLE 3. Training Hyperparameters.

Parameter	Value
Learning rate(lr)	0.0006
Discount factor (γ)	0.95
Clipping range (ϵ)	0.3
Value difference coefficient ($c1$)	1.0
Entropy coefficient ($c2$)	0.01
Batch size	64

A. RATIONALITY AND ROBUSTNESS OF THE IV ELECTION MECHANISM

As described in Section III, the IV election mechanism mainly includes the trust evaluation method and the election function construction. Therefore, in this subsection, we discuss the reasonability of the trust evaluation method and the robustness of the IV election mechanism.

1) RATIONALITY OF TRUST EVALUATION METHOD

We use a vehicle's behavior over 100 time slots to depict the development of trust so as to show the reasonability of the designed trust evaluation approach. The vehicle performs poorly in the 45-th and 46-th time slots out of 100. After the trust rises to a particular level, it engages in illegal conduct in the 74-th time slot again. Beta and weak trust in [40] are leveraged as comparison benchmarks depending on the identical behavior data.

All three trust assessment methods start with an identical initial trust value, as seen in FIGURE 5. In the beta trust, the trust decline rate following illegal conduct is substantially lower than it is for the other two methods, which indicates that beta trust can not harshly punish illegal conduct. This is because beta trust only considers the effects of the most recent malicious behavior, neglecting previous malicious occurrences. Despite the fact that the weak trust method heavily penalizes bad behavior, the trust loss rate following every malicious instance is the same because of the fixed smoothing factor. The resilient trust, however, raises the

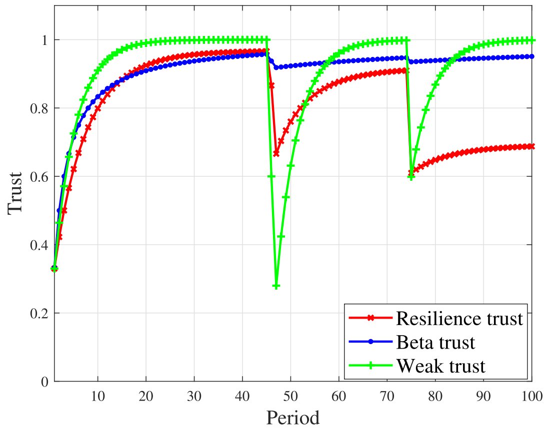

FIGURE 5. Comparison of trust evaluation method.

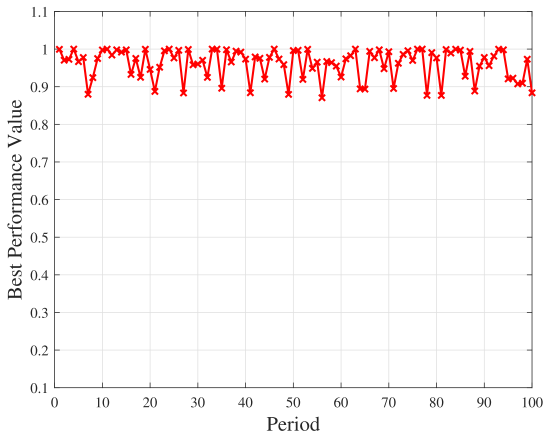

FIGURE 6. Robustness of IV election mechanism.

severity of the punishment for any misconduct. Furthermore, trust is enabled to be fully recovered in both beta trust and weak trust. The fact that resilient trust can only regain 93.8% of the initial trust after the second bad conduct and 71.2% after the third suggests that multiple malicious acts have strict penalties for trust value recovery. A low trust recovery level affects the probability that the vehicle will be elected as an intermediary vehicle in the future, but does not affect its participation in general activities in the network. This demonstrates that the proposed method is the most legitimate and evaluating the trust of each vehicle with the proposed method can significantly improve the system's dependability.

2) ROBUSTNESS OF INTERMEDIARY VEHICLE ELECTION MECHANISM

The robustness of the intermediary vehicle election mechanism is also discussed over 100 periods. It is assumed that there are J potential intermediary vehicles as well as K vehicles within their coverage. With a given probability p, vehicles communicate with one another. To illustrate the robustness of this mechanism, the PFV value associated with the winning vehicle at each interval is shown as a curve.

It is desirable to determine an IV with a comparatively stable performance value while taking its duty of it into consideration. The performance value of the elected IV fluctuates steadily between 0.86 and 1, as seen in FIGURE 6. This is because the trust and the information mastery degree are considered. The high performance value shows that the proposed mechanism can elect an IV that is trustworthy and informed. On top of that, the variance of the performance value stands at 0.0018, demonstrating the ability of the proposed mechanism to consistently elect the best IV for the network. To put it another way, the proposed scheme is robust.

B. EFFECTIVENESS AND RELIABILITY OF OFFLOADING STRATEGY

To simplify the simulation process, we assume that the tasks generated by the requesting vehicle are the same during T. To compare the performance of the proposed offloading strategy in terms of utility and reliability, we compare the proposed approach with three baseline strategies, that is, the local computing method [8], the RSU offloading with PPO under information asymmetry algorithm and the IV randomly offloading algorithm [39], which is described as follows.

- Local Computing (LC): Requesting Vehicle executes all tasks locally. We adopt this method to indicate how much utility offloading brings to the RV relative to local computation. This method serves as a base utility reference.
- RSU Offloding with PPO under Information Asymmetry (IAPPO): Under information asymmetry scenario, RSU uses PPO algorithm to design offloading strategy based on the part of the network information it has mastered. Comparing this strategy with the one proposed, it can be verified whether the proposed intermediary vehicle-assisted offloading framework makes sense in an information asymmetric scenario.
- Intermediary Vehicle Randomly Offloading (IVRO): IV randomly assigns the tasks to the SVs. To compare with other mechanisms more fairly, the best result of 50 random results is taken as the final result of IVRO. We use this strategy for comparison experiments to verify that there is value in developing an offloading strategy using the PPO algorithm as opposed to random offloading.

1) IMPROVEMENT OF CONVERGENCE PERFORMANCE

FIGURE 7 shows the variations in the utility produced by episodes under various strategies. The simulation assumes that the RV generates 5 tasks during T, and the task size is 6Mbit (i.e., $n = 5$ and $D = 6$ Mbit). FIGURE 7 indicates that the utility of IAPPO can reach 50 in the nearly 200*th* episode, while the utility of IAPPO can reach 70 in the nearly 400*th* episode. In other words, IVPPO tends to converge to a higher utility but at the cost of more episodes. This is because the IV is the policy maker in IVPPO, while the RSU is the policy maker in IAPPO. As aforementioned, the RSU cannot perfectly primary all the information in the network, but the IV performs better because of fixed routes and social attributes. Accordingly, the IV has more information about others, and the exploration space of IVPPO

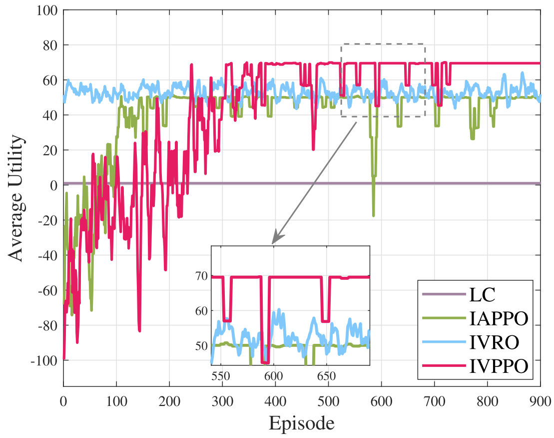

FIGURE 7. **Average Utility achieved during episodes ($w_1 = 0.6, w_2 = 0.4$).**

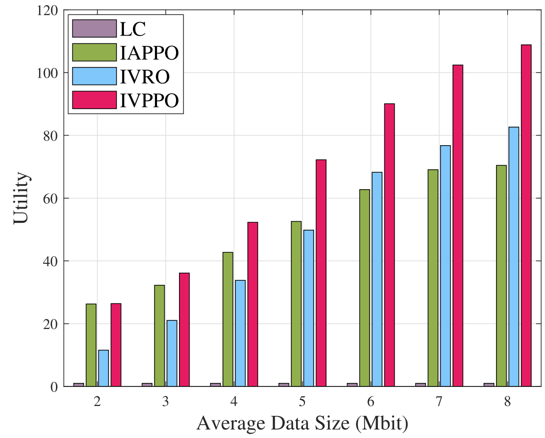

FIGURE 8. **Utility under different average data size ($n = 6$).**

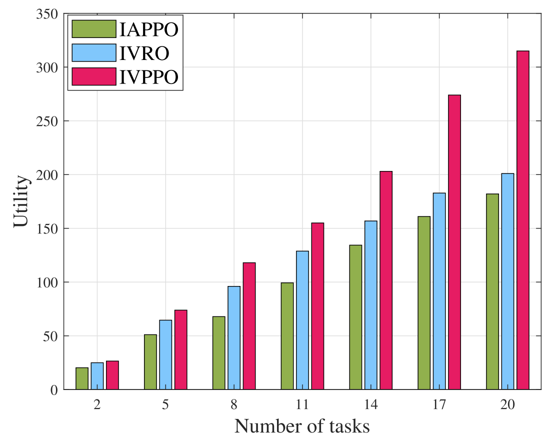

FIGURE 9. **Utility under different number of tasks ($D = 6$ Mbit).**

is bigger. Therefore, the IVPPO takes a longer convergence time and is more likely to obtain higher utility.

In addition, according to Eq. (34-40), the offloading utility of other mechanisms is obtained by comparing them with LC. For comparison with other strategies, the utility in LC is set to 1 and remains unchanged during training episodes. Besides, it can be seen that IVRO outperforms IAPPO in terms of utility. This is because IV is more knowledgeable as mentioned earlier, and the best result of 50 random results is taken as the final result of IVRO. Since the neural network can be used repeatedly, the problem of convergence time can be tolerated, and a higher convergence value is more valuable for the system. In all, the IVPPO proposed in this paper has the best convergence performance.

2) IMPROVEMENT OF OFFLOADING EFFECTIVENESS

FIGURE 8 shows how the average data size impacts the utility of RV when we set $n = 5$. It indicates that the utility of all strategies increases with the expanding average data size. Moreover, the performance gap between the IVPPO and the other two schemes widens with the rise average data size. This is because when the data size of the task becomes larger, the delay caused by offloading through this algorithm is significantly reduced compared with other schemes, which will bring great time value. The greater the data amount is, the more delay is saved, and the more obvious the difference between this algorithm and others. In addition, when the data size is small, the task does not have high performance requirements for the SVs, and the PPO algorithm is more able to design offloading strategies than random offloading. Therefore, the performance of IAPPO is better than IVRO when the data size is small. However, as the data size increases, IVRO outperforms IAPPO because IV has more information about high-performance SVs.

FIGURE 9 shows the utility of the RV versus the different number of tasks when we set $D = 6Mbit$. The results indicate that when the number of tasks increases, the utility of the RV increases. The main reason is that the more tasks the RV generates in a period time, the greater the time saved by offloading and the greater the utility it brings to RV. Besides, IVPPO achieves the best performance, followed by IVRO and IAPPO, and when the number of tasks is large, this advantage is more obvious. This is because, compared with IAPPO, the proposed approach possesses more information about other vehicles to make a better decision for each task. Compared with IVRO, IVPPO uses PPO algorithm to continuously explore the state space and obtain an approximately optimal offloading strategy.

3) IMPROVEMENT OF OFFLOADING RELIABILITY

In FIGURE 10, we examine the offloading reliability of different schemes with varied average data volumes. In LC, since the RV processes the tasks, the reliability of task processing is 1 at most. According to Eq. (38), the reliability of the task is affected by the trust of the SV and the processing time. In IVPPO, IV has more information about other vehicles, some of which may have higher trust. In addition, the processing delay of tasks offloaded by IV is very low, so the reliability of IVPPO is the highest compared with IVRO and IAPPO. And the gap will increase with the rise of average data size. The explanation is that the processing latency will increase as the average task size goes up. The probability of failure will also increase in the process of task completion.

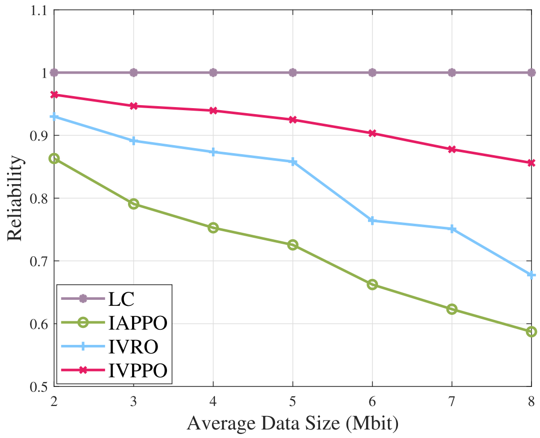

FIGURE 10. Reliability under different average data size.

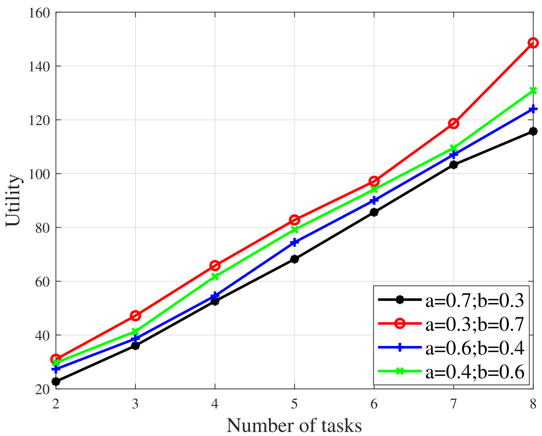

FIGURE 11. Comparison of trust evaluation method.

4) IMPROVEMENT OF OFFLOADING STABILITY IN DIFFERENT APPLICATION SCENARIOS

FIGURE 11 demonstrates the influence of the number of tasks on utility under various weight coefficients when we set $D = 6$Mbit. Since the utility of the RV is the sum of each task it generates, the utility increases with the number of tasks rises. According to Eq. (35), a is the delay coefficient, while b is the energy consumption coefficient. The high value of a and the low value of b mean that the application is more sensitive to delay, and vice versa. FIGURE 11 indicates that IVPPO can achieve relatively high utility for both delay sensitive applications and energy consumption sensitive applications. In other words, the IVPPO can provide stable services for various applications. The main reason is that the elected IV possesses the information of various types of vehicles in the network, it can design corresponding offloading strategies for personalized tasks.

VII. CONCLUSION

In this paper, we proposed an original intermediary vehicle-assisted task offloading strategy in the context of information asymmetry. First, we, inspired by the concept of resilience, designed an intermediary vehicle election mechanism. To be specific, we developed a long-term trust evaluation method. Moreover, we introduced information mastery degree to aid in the election of IV. Then, based on the elected IV, we formulated an objective function to design the offloading strategy, aiming at jointly maximizing the reliability and effectiveness of the offloading. Furthermore, the PPO algorithm was adopted to solve the optimization problem. Finally, numerical results have shown that the developed trust evaluation method can outperform the existing methods in terms of rationality. Besides, compared with other algorithms, the proposed offloading strategy improved convergence performance, effectiveness, and reliability, and it could be stably applied in various actual scenarios.

Our future work will consider the incentives to IV, the mobility of the vehicles and supplementing the simulation with real IoV experimental data. In addition, we will also increase the number of intermediary vehicles and consider the impact of malicious intermediary vehicles on network security.

REFERENCES

[1] C. R. Storck and F. Duarte-Figueiredo, "A survey of 5G technology evolution, standards, and infrastructure associated with vehicle-to-everything communications by Internet of Vehicles," *IEEE Access*, vol. 8, pp. 117593–117614, 2020.
[2] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of Internet of Vehicles," *China Commun.*, vol. 11, no. 10, pp. 1–15, Oct. 2014.
[3] L. Liu, C. Chen, Q. Pei, S. Maharjan, and Y. Zhang, "Vehicular edge computing and networking: A survey," *Mobile Netw. Appl.*, vol. 26, no. 3, pp. 1145–1168, Jun. 2021.
[4] N. Cha, C. Wu, T. Yoshinaga, Y. Ji, and K. A. Yau, "Virtual edge: Exploring computation offloading in collaborative vehicular edge computing," *IEEE Access*, vol. 9, pp. 37739–37751, 2021.
[5] M. S. Bute, P. Fan, L. Zhang, and F. Abbas, "An efficient distributed task offloading scheme for vehicular edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 13149–13161, Dec. 2021.
[6] S. Raza, M. A. Mirza, S. Ahmad, M. Asif, M. B. Rasheed, and Y. Ghadi, "A vehicle to vehicle relay-based task offloading scheme in vehicular communication networks," *PeerJ Comput. Sci.*, vol. 7, p. e486, Apr. 2021.
[7] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4377–4387, Jun. 2019.
[8] Q. Luo, C. Li, T. H. Luan, W. Shi, and W. Wu, "Self-learning based computation offloading for Internet of Vehicles: Model and algorithm," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 5913–5925, Sep. 2021.
[9] F. Zeng, Q. Chen, L. Meng, and J. Wu, "Volunteer assisted collaborative offloading and resource allocation in vehicular edge computing," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3247–3257, Jun. 2021.
[10] F. Sun, F. Hou, N. Cheng, M. Wang, H. Zhou, L. Gui, and X. Shen, "Cooperative task scheduling for computation offloading in vehicular cloud," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11049–11061, Nov. 2018.
[11] D. P. Proos and N. Carlsson, "Performance comparison of messaging protocols and serialization formats for digital twins in IoV," in *Proc. IFIP Netw. Conf. (Networking)*, Jun. 2020, pp. 10–18.
[12] S. Kumar and S. Misra, "Backhaul-aware storage allocation and pricing mechanism for RSU-based caching networks," *IEEE Trans. Wireless Commun.*, vol. 21, no. 3, pp. 1725–1734, Mar. 2022.
[13] M. Khabbaz, J. Antoun, S. Sharafeddine, and C. Assi, "Modeling and delay analysis of intermittent V2U communication in secluded areas," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3228–3240, May 2020.
[14] W. Chen, Y. Zhu, Z. Wu, and Z. He, "Vehicle—RSU interaction behavior modeling and RSU deployment evaluation for bus priority," in *Proc. IEEE 25th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2022, pp. 3188–3193.
[15] C. Yang, J. Li, and A. Anpalagan, "Hierarchical decision-making with information asymmetry for spectrum sharing systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 4359–4364, Sep. 2015.
[16] Z. Zhou, H. Liao, B. Gu, S. Mumtaz, and J. Rodriguez, "Resource sharing and task offloading in IoT fog computing: A contract-learning approach," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 3, pp. 227–240, Jun. 2020.

[17] Z. Zhou, H. Liao, X. Zhao, B. Ai, and M. Guizani, "Reliable task offloading for vehicular fog computing under information asymmetry and information uncertainty," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8322–8335, Sep. 2019.

[18] X. Huang, R. Yu, D. Ye, L. Shu, and S. Xie, "Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3773–3787, Apr. 2021.

[19] B. Wu, T. Chen, K. Yang, and X. Wang, "Edge-centric bandit learning for task-offloading allocations in multi-RAT heterogeneous networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3702–3714, Apr. 2021.

[20] S. Dhelim, N. Aung, M. T. Kechadi, H. Ning, L. Chen, and A. Lakas, "Trust2Vec: Large-scale IoT trust management system based on signed network embeddings," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 553–562, Jan. 2023.

[21] Y. Wang, Y. Tian, X. Hei, L. Zhu, and W. Ji, "A novel IoV block-streaming service awareness and trusted verification scheme in 6G," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5197–5210, Jun. 2021.

[22] J. Liu and Q. Zhang, "Offloading schemes in mobile edge computing for ultra-reliable low latency communications," *IEEE Access*, vol. 6, pp. 12825–12837, 2018.

[23] X. Hou, Z. Ren, J. Wang, W. Cheng, Y. Ren, K.-C. Chen, and H. Zhang, "Reliable computation offloading for edge-computing-enabled software-defined IoV," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7097–7111, Aug. 2020.

[24] S. Xie, H. Li, L. Li, Z. Chen, and S. Li, "Reliable and energy-aware job offloading at terahertz frequencies for mobile edge computing," *China Commun.*, vol. 17, no. 12, pp. 17–36, Dec. 2020.

[25] G. Zhang, Y. Lu, X. Wang, and X. Li, "Trust-based intermediary vehicle election provisioning with resilience under information asymmetry," in *Proc. IEEE 96th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2022, pp. 1–5.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.

[27] M. Mahzarnia, M. P. Moghaddam, P. T. Baboli, and P. Siano, "A review of the measures to enhance power systems resilience," *IEEE Syst. J.*, vol. 14, no. 3, pp. 4059–4070, Sep. 2020.

[28] N. Aung, S. Dhelim, L. Chen, A. Lakas, W. Zhang, H. Ning, S. Chaib, and M. T. Kechadi, "VeSoNet: Traffic-aware content caching for vehicular social networks using deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, early access, Mar. 14, 2023, doi: 10.1109/TITS.2023.3250320.

[29] X. Li, T. Jing, R. Li, H. Li, X. Wang, and D. Shen, "BDRA: Blockchain and decentralized identifiers assisted secure registration and authentication for VANETs," *IEEE Internet Things J.*, vol. 10, no. 14, pp. 12140–12155, Jul. 2023.

[30] S. Suhail, R. Hussain, A. Khan, and C. S. Hong, "On the role of hash-based signatures in quantum-safe Internet of Things: Current solutions and future directions," *IEEE Internet Things J.*, vol. 8, no. 1, pp. 1–17, Jan. 2021.

[31] X. Chen, J. Ding, and Z. Lu, "A decentralized trust management system for intelligent transportation environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 558–571, Jan. 2022.

[32] J. Jiang, X. Zhu, G. Han, M. Guizani, and L. Shu, "A dynamic trust evaluation and update mechanism based on C4.5 decision tree in underwater wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 9031–9040, Aug. 2020.

[33] Z. Ying, M. Ma, Z. Zhao, X. Liu, and J. Ma, "A reputation-based leader election scheme for opportunistic autonomous vehicle platoon," *IEEE Trans. Veh. Technol.*, vol. 71, no. 4, pp. 3519–3532, Apr. 2022.

[34] X. Huang, R. Yu, J. Kang, and Y. Zhang, "Distributed reputation management for secure and efficient vehicular edge computing and networks," *IEEE Access*, vol. 5, pp. 25408–25420, 2017.

[35] M. Shinozuka, S. E. Chang, T. C. Cheng, M. Feng, T. D. O'Rourke, M. A. Saadeghvaziri, X. Dong, X. Jin, Y. Wang, and P. Shi, "Resilience of integrated power and water systems," *Seismic Eval. Retrofit Lifeline Syst.*, vol. 1, pp. 65–86, Jan. 2003.

[36] W. Zhan, C. Luo, J. Wang, C. Wang, G. Min, H. Duan, and Q. Zhu, "Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5449–5465, Jun. 2020.

[37] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13124–13138, Nov. 2020.

[38] R. Chai, M. Li, T. Yang, and Q. Chen, "Dynamic priority-based computation scheduling and offloading for interdependent tasks: Leveraging parallel transmission and execution," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10970–10985, Oct. 2021.

[39] T. D. T. Nguyen, V. Nguyen, V.-N. Pham, L. N. T. Huynh, Md. D. Hossain, and E.-N. Huh, "Modeling data redundancy and cost-aware task allocation in MEC-enabled Internet-of-Vehicles applications," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1687–1701, Feb. 2021.

[40] S. Gyawali, Y. Qian, and R. Q. Hu, "Deep reinforcement learning based dynamic reputation policy in 5G based vehicular communication networks," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6136–6146, Jun. 2021.

YANFEI LU received the Ph.D. degree from the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing, China, in 2014. He is currently a Professor with the School of Electronics and Information Engineering, Beijing Jiaotong University. His current research interests include network protocols, vehicular networks, and algorithm design.

GUIYU ZHANG (Student Member, IEEE) received the B.S. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China, in 2021, where she is currently pursuing the M.S. degree. Her research interests include vehicular edge computing and reliable offloading.

XIAOXUAN WANG (Member, IEEE) received the Ph.D. degree in traffic control and information engineering from Beijing Jiaotong University, Beijing, China, in 2020. He was a Visiting Ph.D. Student with Virginia Tech, from 2017 to 2018. He is currently a Faculty Member with Beijing Jiaotong University. His general research interests mainly lie in intelligent transportation and industrial internet including T2W, T2T, V2V communications, and cognitive control. He has served as a Reviewer for a number of journals, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY.

XUEHAN LI received the B.S. degree in electronic and information engineering from Beijing Jiaotong University, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree. His research interests include security and privacy-preserving in VANETs, blockchain technology, and vehicular edge computing.