A General Trust Framework for Multi-Agent Systems

Mingxi Cheng
University of Southern California
mingxic@usc.edu

Chenzhong Yin
University of Southern California
chenzhoy@usc.edu

Junyao Zhang
University of Southern California
junyaozh@usc.edu

Shahin Nazarian
University of Southern California
shahin.nazarian@usc.edu

Jyotirmoy Deshmukh
University of Southern California
jdeshmuk@usc.edu

Paul Bogdan
University of Southern California
pbogdan@usc.edu

ABSTRACT

Transportation systems of the future can be best modeled as multi-agent systems. A number of coordination protocols such as autonomous intersection management (AIM), adaptive cooperative traffic light control (TLC), cooperative adaptive cruise control (CACC), among others have been developed with the goal of improving the safety and efficiency of such systems. The overall goal in these systems is to provide behavioral guarantees under the assumption that the participating agents work in concert with a centralized (or distributed) coordinator. While there is work on analyzing such systems from a security perspective, we argue that there is limited work on quantifying trustworthiness of individual agents in a multi-agent system. We propose a framework that uses an epistemic logic to quantify trustworthiness of agents, and embed the use of quantitative trustworthiness values into control and coordination policies. Our modified control policies can help the multi-agent system improve its safety in the presence of untrustworthy agents (and under certain assumptions, including malicious agents). We empirically show the effectiveness of our proposed trust framework by embedding it into AIM, TLC, and CACC platooning algorithms. In our experiments, our trust framework accurately detects attackers in CACC platoons; mitigates the effect of untrustworthy agents in AIM; and trust-aware TLC and AIM reduce collisions in all cases compared to the vanilla versions of these algorithms.

KEYWORDS

Trust Framework, Multi-agent Systems, Platoons, Autonomous Intersection Management, Reinforcement Learning

ACM Reference Format:
Mingxi Cheng, Chenzhong Yin, Junyao Zhang, Shahin Nazarian, Jyotirmoy Deshmukh, and Paul Bogdan. 2021. A General Trust Framework for Multi-Agent Systems. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Multi-agent systems (MASs) consist of multiple, interacting, intelligent cyber-agents [2, 8, 26], and the successful behavior of a MAS typically depends on safe coordination between the agents. For autonomous and mobile MASs, such as those found in ground transportation systems or in unmanned aerial vehicles comprising avionic systems, coordination may be used to endow greater safety over human-operated agents or to improve the efficiency of the system (e.g. traffic throughput or increasing the sensing range) or both. For instance, in the context of traffic light control or autonomous intersection management [11], the goal is to improve the throughput of traffic intersections in a safe fashion, for traffic consisting of a mixture of human-driven, semi-autonomous and autonomous vehicles [5, 27]. Similarly, there is work on cooperative adaptive cruise control where the objective is to improve trafficfl ow and fuel consumption while ensuring collision-freedom [15, 23].

An important consideration for MASs is to achieve safe and efficient coordination when the MAS consists of a mixture of trusted and untrusted agents. Here, being trustworthy can encapsulate different things: (i) the agent follows the commands of the coordinator to a high degree of precision, (ii) the agent reports its state (e.g. position, velocity) with consistent accuracy, or (iii) the agent is not malicious, i.e. it does not purposefully engage in behavior that can endanger system safety. For instance, vehicle platooning systems require AI strategies to analyze platoon members and evaluate their degree of trustworthiness in order to avoid attacks that can lead to accidents. In the works of [6, 13, 14], researchers take the front collision warning, lane departure warning, and autonomous braking system into consideration to construct a trust evaluation framework. However, these approaches analyze individual vehicles in isolation and do not account for communication and co-operation among vehicles. Moreover, in these approaches, the system can only react to only one malicious attack. When the platoon system is attacked by multiple malicious agents, the system can be deceived and led to a catastrophic state. Existing trust frameworks are *ad hoc*, which makes it difficult to apply them universally. In this paper, we propose a universal framework based on a logical characterization of trust that allows us to quantify trust in individual agents in a systematic fashion. Our framework considers both short-term and long-term behavioral histories of agents to quantify their trustworthiness.

We envision a cloud-based (or edge-based) architecture where trust values for agents are stored in a secure fashion, AND where authenticated decision-making nodes (such as centralized or distributed coordinators) are able to access trust values for agents to make real-time decisions. Through quantitative trustworthiness scores, we are able to perform trust-aware decision-making, where a coordinator is able to explore trade-offs between safety and efficiency when orchestrating coordination for a mixture of trusted and untrusted agents. Our main contributions are as follows:

- We propose a framework to mathematically quantify trustworthiness of agents in MASs using the formalism of *subjective logic*.

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

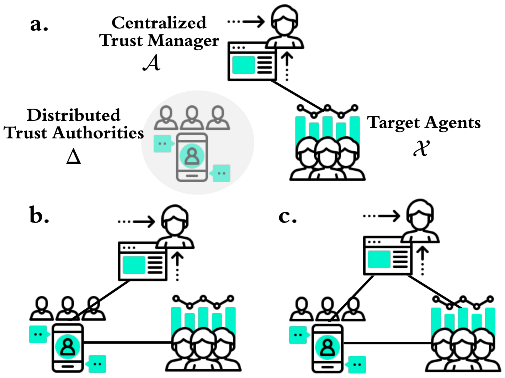


Figure 1: a. A trust framework where the centralized trust manager $\mathcal{A}$ keeps inspecting target agents $\mathcal{X}$. b. $\mathcal{A}$ does not directly inspect $\mathcal{X}$ but relies on distributed trust authorities Δ, which may or may not be trustworthy. c. Both $\mathcal{A}$ and Δ directly inspect $\mathcal{X}$.

We propose that an agent's trustworthiness is updated using long-term and short-term observations of the agent's behavior.

- We provide a trust-aware decision-making framework that uses the quantified trust values to choose coordination policies that achieve the desired trade off between safety and efficiency.
- We demonstrate the feasibility and applicability of the proposed trust framework by applying it in three MASs: cooperative adaptive cruise control (CACC)-based platoons, an autonomous intersection management (AIM) system, and a reinforcement learning-based traffic light control (TLC) system. With minimum modification of existing MASs (e.g., AIM, TLC), we can formulate trust-aware decision-making strategies and achieve better performance.

2 QUANTIFYING TRUST IN MAS

In MASs, such as air/drone traffic control system [7, 17], adaptive cruise control system [16], multi-agent autonomous traffic management [4], and even federated learning [19] in machine learning, the safety and behavior of one or a subset of agents affects the efficiency and safety of the whole system. Such systems usually are vulnerable to agents that are untrustworthy for various reasons including operating defects, uncertain operating environments or purposeful malice. In these cases, a subjective measurement is a must to identify untrustworthy agents. Therefore, we propose a trust quantification framework based on *Subjective Logic* (SL) [18]. Our framework interprets agent-behaviors and assigns a trustworthiness score to the agents.

To explain the basic idea, we assume that the MAS is endowed with a secure and trusted observer known as the *trust manager* (denoted as $\mathcal{A}$) that observes the behavior of agents, extracts knowledge (*opinion* in SL parlance) from observations (*evidence* in SL parlance) and computes the agents' trustworthiness. We now provide the definitions required for the calculation of trustworthiness in our proposed trust framework. Given a trust manager $\mathcal{A}$ and a specified agent X in a MAS, let $b_X^{\mathcal{A}}$ denote the *belief mass* that $\mathcal{A}$ has in X, let $d_X^{\mathcal{A}}$ denote the *disbelief mass*, let $u_X^{\mathcal{A}}$ denote the *uncertainty mass*, and $a_X^{\mathcal{A}}$ denote the *base rate*. Intuitively, the belief and disbelief loosely correspond to the probabilities of an agent being trustworthy and untrustworthy. Uncertainty represents the lack of evidence to support any specific probability, e.g., $u_X^{\mathcal{A}} = 1$ represents we know nothing about agent's behavior and by default, with chance of $a_X^{\mathcal{A}} = 0.5$, it can be trustworthy.

Definition 2.1 (Opinion [18]). In SL, a binomial opinion $\overline{W}_X^{\mathcal{A}} = \{b_X^{\mathcal{A}}, d_X^{\mathcal{A}}, u_X^{\mathcal{A}}, a_X^{\mathcal{A}}\}$ represents the opinion of an observer $\mathcal{A}$ about X, where $b_X^{\mathcal{A}}$, $d_X^{\mathcal{A}}$, $u_X^{\mathcal{A}}$, and $a_X^{\mathcal{A}}$ are as previously defined, and $b_X^{\mathcal{A}} + d_X^{\mathcal{A}} + u_X^{\mathcal{A}} = 1$ for $a_X^{\mathcal{A}} \in [0, 1]$ and base rate is akin to a prior.

Definition 2.2 (Trustworthiness [8, 18]). The *trustworthiness* of X assessed by $\mathcal{A}$ is defined as $p_X^{\mathcal{A}} = b_X^{\mathcal{A}} + u_X^{\mathcal{A}} * a_X^{\mathcal{A}}$, where $b_X^{\mathcal{A}}$, $u_X^{\mathcal{A}}$ and $a_X^{\mathcal{A}}$ are as defined previously, and $p_X^{\mathcal{A}} \in [0, 1]$.

Definition 2.3 (Evidence [18]). Given a behavioral property φ, a *positive evidence* r quantifies the satisfaction of the property φ by a behavior of X as observed by $\mathcal{A}$, a *negative evidence* s quantifies the violation of φ by the observed behavior of X. A binomial opinion is formed using evidences based on the principle that r contributes to the belief mass and s contributes to disbelief mass using the following equations:

$$b_X^{\mathcal{A}} = \frac{r}{r+s+\omega}, \quad d_X^{\mathcal{A}} = \frac{s}{r+s+\omega}, \quad u_X^{\mathcal{A}} = \frac{\omega}{r+s+\omega}, \qquad (1)$$

where $\omega = 2$ is a default non-informative prior weight.

Fig. 1a shows our proposed trust-aware MAS consisting of a centralized manager $\mathcal{A}$ that keeps inspecting agents and updates their trustworthiness $p_X^{\mathcal{A}}, \forall X \in \mathcal{X}$ (where $\mathcal{X}$ represents the set of all agents) based on time-varying $\overline{W}_X^{\mathcal{A}}, \forall X \in \mathcal{X}$ (which are updated based on observed evidences r or s). Instead of keeping a record of all past evidence histories, i.e., r and s, we keep a hash table $\mathcal{H}$ that records (long-term) opinions of $\mathcal{X}$ and use a *cumulative fusion operator* [18] to merge established (long-term) opinions and newly observed (short-term) opinions.[1]

Definition 2.4 (Cumulative Fusion Operator). Let us assume a long-term opinion about agent X, $\overline{W}_X^{\mathcal{A}}$, is calculated based on previous observations $r_{[0,t-\tau]}$ and $s_{[0,t-\tau]}$ from time 0 to $t-\tau$, and newly observed evidences $r_{[t-\tau,t]}$ and $s_{[t-\tau,t]}$ form a short-term opinion $\overline{W}_X^{E}$. The updated opinion takes evidences from time period $[0, t]$, which is equivalent to the *cumulative fusion* of $\overline{W}_X^{\mathcal{A}}$ and $\overline{W}_X^{E}$:

$$\overline{W}_X^{\mathcal{A}} \leftarrow \overline{W}_X^{\mathcal{A}\diamond E} = \overline{W}_X^{\mathcal{A}} \oplus \overline{W}_X^{E}. \qquad (2)$$

See Supplementary Materials Section A for derivation details. [2]

In addition to the centralized trust authority $\mathcal{A}$, there are also distributed trust authorities Δ that help inspect agents and collect evidence as shown in Fig. 1b. The existence of Δ enlarges the observation range, increases the observation frequency, and relaxes the requirement that $\mathcal{A}$ needs to directly inspect agents. Δ keeps local trust records of covered agents and sends updates to $\mathcal{A}$ regularly. For example, in traffic systems, local road side units inspect

[1] Assume short- and long-term opinions are established by evidences (r_1, s_1) and (r_2, s_2), which are observed in non-overlapping time periods. Applying cumulative fusion to combine opinions is equivalent to summing up evidences (r_1, s_1) and (r_2, s_2).
[2] Supplementary Materials can be found in this link https://drive.google.com/drive/folders/1gxEYtS_v3HLyZ7outQiM-bdQIvmt0VFe?usp=sharing.

vehicles and report to the department of motor vehicles. To merge the (long-term) opinions from $\mathcal{A}$ and Δ, we use cumulative fusion operator: $\overline{W}_X^{\mathcal{A}} \leftarrow \overline{W}_X^{\mathcal{A}\diamond\Delta}$. Note that, in our trust framework, $\mathcal{A}$ can operate alone without helpers Δ due to the assumption that $\mathcal{A}$ can directly inspect agents as shown in Fig. 1a.

We assume that $\mathcal{A}$ is always trustworthy, and Δ may or may not be trustworthy. For example, in a traffic system, if road side units serve as Δ, then they usually are trustworthy. However, if vehicles serve as Δ, for example, if the trailing vehicle reports to $\mathcal{A}$ about leader vehicle, then Δ can be untrustworthy and their trust evaluations may not be trustworthy. To deal with such scenarios, $\mathcal{A}$ applies a discounting factor [18] to take Δ's own trustworthiness into consideration when relying Δ's evaluations.[3]

Definition 2.5 (Discounting Operator**).** Assume $\mathcal{A}$ would like to develop trust in X and $\mathcal{A}$ relies on Δ for evidence collection and opinion/trust evaluation. $\mathcal{A}$'s opinion about Δ is represented as $\overline{W}_\Delta^{\mathcal{A}}$, and Δ's opinion about X is $\overline{W}_X^{\Delta}$. Based on the combination of $\mathcal{A}$'s trust in Δ and Δ's opinion about X, $\mathcal{A}$ updates its opinion about X using the *discounting operator* $\otimes$:

$$\overline{W}_X^{[\mathcal{A};\Delta]} = \overline{W}_\Delta^{\mathcal{A}} \otimes \overline{W}_X^{\Delta}. \tag{3}$$

Complete mathematical derivations of discounting operator $\otimes$ is given in Supplementary Material section B. $\overline{W}_X^{[\mathcal{A};\Delta]}$ is a short-term opinion. To merge with the long-term opinion $\overline{W}_X^{\mathcal{A}}$, substituting $\overline{W}_X^{E}$ with $\overline{W}_X^{[\mathcal{A};\Delta]}$ in Eq. 2 generates the designated result. In cases where there are both $\mathcal{A}$ and Δ as shown in Fig. 1c, or there are multiple Δ inspecting the target agent X at the same time, we need to have a way to merge multiple short-term opinions together. We can make use of the *averaging fusion* operator in SL to take the average of two opinions observed at the same time [18].[4]

Definition 2.6. Subject to trust authorities $\mathcal{A}$ and Δ, and a specified agent X in a multi-agent system, assume both $\mathcal{A}$ and Δ inspect X in the same time period $[t-\tau, t]$ and Δ may or may not be trustworthy. $\mathcal{A}$ and Δ develop opinions to X as $\overline{W}_X^{\mathcal{A}}$, and $\overline{W}_X^{\Delta}$, respectively. The short-term opinion about X combines both authorities' opinions via an *averaging fusion operator* $\underline{\oplus}$:

$$\overline{W}_X^{\mathcal{A}\underline{\diamond}[\mathcal{A};\Delta]} = \overline{W}_X^{\mathcal{A}} \underline{\oplus} \overline{W}_X^{[\mathcal{A};\Delta]}. \tag{4}$$

Detailed math equations of averaging fusion operator $\underline{\otimes}$ can be found in Supplementary Materials Section C. If distributed authority Δ is trustworthy then Eq. 4 is simplified as $\overline{W}_X^{\mathcal{A}\underline{\diamond}\Delta} = \overline{W}_X^{\mathcal{A}} \underline{\oplus} \overline{W}_X^{\Delta}$. If the observing authorities are both distributed authorities, namely Δ_1 and Δ_2, then Eq. 4 reads: $\overline{W}_X^{[A;\Delta_1]\underline{\diamond}[A;\Delta_2]}$. Then to merge with long-term history of X, use cumulative fusion operator defined in Definition 2.4. A demonstration example of our proposed trust framework in traffic systems is shown in Fig. 2, which corresponding to the scenario in Fig. 1c.

To demonstrate how the proposed trust framework works in different applications, we first show its feasibility in the context of CACC platoons (Section 3) where the distributed trust authorities are not necessarily trustworthy. We show with simulation results that our trust-based attack detection can accurately detect attackers.

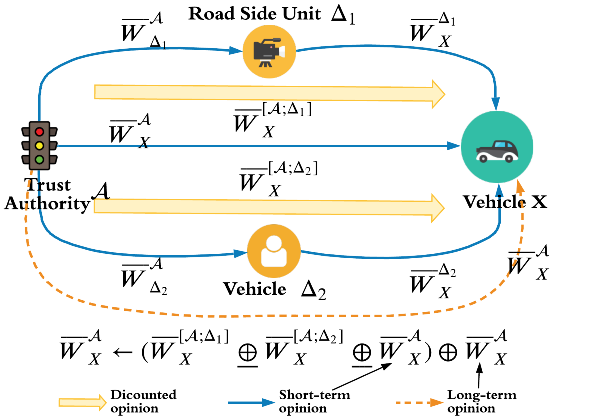

Road Side Unit Δ_1
$\overline{W}_{\Delta_1}^{\mathcal{A}}$
$\overline{W}_X^{\Delta_1}$
$\overline{W}_X^{[\mathcal{A};\Delta_1]}$
$\overline{W}_X^{\mathcal{A}}$
$\overline{W}_X^{[\mathcal{A};\Delta_2]}$
Trust Authority $\mathcal{A}$
Vehicle X
$\overline{W}_{\Delta_2}^{\mathcal{A}}$
Vehicle Δ_2
$\overline{W}_X^{\Delta_2}$
$\overline{W}_X^{\mathcal{A}}$
$\overline{W}_X^{\mathcal{A}} \leftarrow (\overline{W}_X^{[\mathcal{A};\Delta_1]} \underline{\oplus} \overline{W}_X^{[\mathcal{A};\Delta_2]} \underline{\oplus} \overline{W}_X^{\mathcal{A}}) \oplus \overline{W}_X^{\mathcal{A}}$
Dicounted opinion
Short-term opinion
Long-term opinion

Figure 2: A trust framework in traffic systems. $\mathcal{A}$ and Δ keep inspecting the target vehicle X. Both road side units and other vehicles adjacent to X serve as Δ. If we assume road side units are trustworthy, then the opinion updating equation can be simplified as $\overline{W}_X^{\mathcal{A}} \leftarrow (\overline{W}_X^{\Delta_1} \underline{\oplus} \overline{W}_X^{[\mathcal{A};\Delta_2]} \underline{\oplus} \overline{W}_X^{\mathcal{A}}) \oplus \overline{W}_X^{\mathcal{A}}$, where on the left hand side, the first $\overline{W}_X^{\mathcal{A}}$ in the bracket is short-term opinion, and the second $\overline{W}_X^{\mathcal{A}}$ is a long-term opinion extracted from $\mathcal{H}$.

To demonstrate how to extend existing control policies to be trust-aware, we provide two case studies regarding intersection management. We consider situations where trustworthy distributed authorities are appreciated but not necessary. The first case study in Section 4, is of the Autonomous Intersection Management (AIM) protocol [27], where we show how to modify AIM to a trust-aware version called AIM-Trust. Here, we empirically demonstrate that AIM-Trust can achieve an effective trade off between throughput and safety (defined as freedom from collisions). The second case study in Section 5 uses a coordination policy learned using reinforcement learning (RL) for traffic light control (TLC). We augment the TLC policy with trust-awareness in a minimal fashion, and show that the collision rate decreases in all scenarios involving vehicles with mixed trustworthiness values.

3 TRUST-BASED MALICIOUS ATTACKER DETECTION IN CACC PLATOONS

3.1 CACC Platoons

Recent advances in vehicle-to-everything (V2X) communication have enabled the development of platooning to save energy, improve efficiency, and ensure safety [6]. In a platoon, a chain of vehicles equipped with V2X sense the surroundings and maintain a constant inter-vehicle space. The head vehicle controls the platoon by broadcasting its kinematic data, such as its designated *velocity* v, and inter-vehicle *space* d. The member vehicles follow the head vehicle's instructions and use beacons from other platoon members to control velocity and inter-vehicle space. [5]

[3] Assume Δ's trustworthiness is $p_\Delta^{\mathcal{A}}$, then $\mathcal{A}$ discounts Δ's opinions by $p_\Delta^{\mathcal{A}}$.

[4] Assume two short-term opinions are established by evidences (r_1, s_1) and (r_2, s_2), which are observed in the same time periods. Applying averaging fusion to combine opinions is equivalent to take average of evidences (r_1, s_1) and (r_2, s_2).

[5] Beacon messages containing vehicle information are communicated by vehicles to increase cruise stability [14].

3.2 Attacker Model

Various CACC (platoon) attacks have been proposed in the literature, including jamming attacks, V2X data injection [3] and sensor manipulation attacks [28]. Attack defense models such as misbehavior detection has also been studied [14]. In this case study, we aim to detect attackers in platoons, so the core of our attacker model is that attackers gain control over vehicles and their actions are observable by participants. In order to detect adversarial behavior, we focus on V2X data injection attacks: acceleration data injections.

3.3 Trust-based Attacker Detection Model

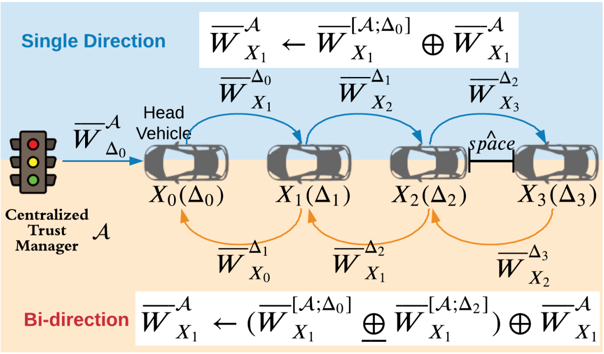

Single Direction
$\overline{W}_{X_1}^{\mathcal{A}} \leftarrow \overline{W}_{X_1}^{[\mathcal{A};\Delta_0]} \oplus \overline{W}_{X_1}^{\mathcal{A}}$
$\overline{W}_{X_1}^{\Delta_0}$
$\overline{W}_{X_2}^{\Delta_1}$
$\overline{W}_{X_3}^{\Delta_2}$
Head Vehicle
$\overline{W}_{\Delta_0}^{\mathcal{A}}$
$\hat{space}$
$X_0(\Delta_0)$
$X_1(\Delta_1)$
$X_2(\Delta_2)$
$X_3(\Delta_3)$
Centralized Trust Manager $\mathcal{A}$
$\overline{W}_{X_0}^{\Delta_1}$
$\overline{W}_{X_1}^{\Delta_2}$
$\overline{W}_{X_2}^{\Delta_3}$
Bi-direction
$\overline{W}_{X_1}^{\mathcal{A}} \leftarrow (\overline{W}_{X_1}^{[\mathcal{A};\Delta_0]} \underline{\oplus} \overline{W}_{X_1}^{[\mathcal{A};\Delta_2]}) \oplus \overline{W}_{X_1}^{\mathcal{A}}$

Figure 3: Trust-based attacker detection model with single and bi-directional trust evaluations.

We now demonstrate how to apply our trust framework to detect attackers in CACC platoons. We assume a centralized trust authority $\mathcal{A}$ that maintains a trustworthiness table $\mathcal{H}$. Such an authority could either be a cloud-based service or an edge computing node. We assume that the head vehicle is the leader and $\mathcal{A}$ only directly inspects leader to reduce inspection intensity. We also assume that each vehicle X serves as a distributed trust authority Δ and reports to $\mathcal{A}$ when evaluating the adjacent vehicles; and it is also a target, when the adjacent vehicles evaluate X. Since Δ can be untrustworthy, when it reports to $\mathcal{A}$, we apply the discounting operator defined in Eq. 3, Definition 2.5. Assume the long-term trust histories of X and and its successor and predecessor vehicles Δ_1 and Δ_2 are $\overline{W}_X^{\mathcal{A}}$, $\overline{W}_{\Delta_1}^{\mathcal{A}}$, and $\overline{W}_{\Delta_2}^{\mathcal{A}}$, respectively. Δ_1 and Δ_2 use sensors to get accurate information including sensed inter-vehicle distances $\boldsymbol{x}_{\Delta_1}^X, \boldsymbol{x}_{\Delta_2}^X$ and sensed X's speed $\boldsymbol{sp}_{\Delta_1}^X, \boldsymbol{sp}_{\Delta_2}^X$. Therefore, they evaluate X and the resulting short-term trust/opinions are $\overline{W}_X^{\Delta_1}$ and $\overline{W}_X^{\Delta_2}$. Then the short-term opinion about X reads:

$$\overline{W}_X^{[\mathcal{A};\Delta_1]} \underline{\oplus} \overline{W}_X^{[\mathcal{A};\Delta_2]} = (\overline{W}_{\Delta_1}^{\mathcal{A}} \otimes \overline{W}_X^{\Delta_1}) \underline{\oplus} (\overline{W}_{\Delta_2}^{\mathcal{A}} \otimes \overline{W}_X^{\Delta_2}). \tag{5}$$

After combining the long-term opinion, the opinion about X reads: $\overline{W}_X^{\mathcal{A}} \leftarrow (\overline{W}_X^{[\mathcal{A};\Delta_1]} \underline{\oplus} \overline{W}_X^{[\mathcal{A};\Delta_2]}) \oplus \overline{W}_X^{\mathcal{A}}$. This bi-directional trust evaluation takes information from both the vehicles that are right before and after the target vehicle as illustrated in Fig. 3. To reduce communication intensity, this trust evaluation can be downgraded to single directional as shown in the top half of Fig. 3. In single-directional trust evaluation, each vehicle is only evaluated by its direct predecessor Δ. Hence, the opinion update equation of vehicle X takes a simplified version: $\overline{W}_X^{\mathcal{A}} \leftarrow \overline{W}_X^{[\mathcal{A};\Delta]} \oplus \overline{W}_X^{\mathcal{A}} = (\overline{W}_\Delta^{\mathcal{A}} \otimes \overline{W}_X^{\Delta}) \oplus \overline{W}_X^{\mathcal{A}}$. We assume that in both single and bi-directional evaluations, the head vehicle's predecessor is the trustworthy $\mathcal{A}$.

In fact, $\overline{W}_X^{\Delta}$ is evidence-based and now we present how to derive evidences. A vehicle gains accurate position and velocity information of its adjacent vehicles via sensors, and the reported information from beacons (of other vehicles). Therefore, vehicles as Δ measure evidence using a set of rules to determine if the adjacent vehicle is trustworthy based on the assumption that the sensor data is always accurate. In what follows, we use φ, ψ, and ξ to denote behavioral properties in an appropriate formalism such as Signal Temporal Logic (STL) [22]. For brevity, we omit a detailed explanation of STL; in our notation, $(\boldsymbol{x}^X, t) \models \varphi$ denotes that starting from time t, the behavior $\boldsymbol{x}^X$ satisfies φ, and $(\boldsymbol{x}^X, t) \models \mathrm{G}_{[t_1,t_2]}\varphi$ indicates that the formula φ holds at all times between $t + t_1$ and $t + t_2$. A set of platoon-specific rules determines positive (r) and negative (s) evidence:

$$\begin{array}{ll} r = r + 1 & \text{if } (\boldsymbol{x}^X, t) \models \varphi \wedge (\boldsymbol{sp}^X, t) \models \psi \wedge (\boldsymbol{jk}^X, t) \models \xi; \\ s = s + 1 & \text{otherwise.} \end{array} \tag{6}$$

$$\varphi \equiv \quad \mathrm{G}_{[t_1,t_2]} \left(|\boldsymbol{x}^X(t) - d| \leq \epsilon_{space} \wedge |\boldsymbol{x}^X(t) - \boldsymbol{x}_\Delta^X(t)| \leq \epsilon_{space} \right) \tag{7}$$

$$\psi \equiv \quad \mathrm{G}_{[t_1,t_2]} \left(|\boldsymbol{sp}^X(t) - v| \leq \epsilon_{speed} \wedge |\boldsymbol{sp}^X(t) - \boldsymbol{sp}_\Delta^X(t)| \leq \epsilon_{speed} \right) \tag{8}$$

$$\xi \equiv \quad \mathrm{G}_{[t_1,t_2]} \left(\boldsymbol{jk}^X(t) \leq \epsilon_{jkness} \wedge |\boldsymbol{jk}^X(t) - \boldsymbol{jk}_\Delta^X(t)| \leq \epsilon_{jkness} \right) \tag{9}$$

Eq. 6 indicates that the reported inter-space of X from beacons, $\boldsymbol{x}^X$, should not deviate from the requested d by more than ϵ_{space} in time interval $[t + t_1, t + t_2]$, where t_1 and t_2 are hyper parameters. Similarly, reported $\boldsymbol{x}^X$ should not deviate from the sensed $\boldsymbol{x}_\Delta^X$ by more than ϵ_{space}. Similar rules apply for speed $\boldsymbol{sp}^X$ and the jerk value $\boldsymbol{jk}^X$, which we estimate by taking the difference between the accelerations values for the last and current beacons. High jerk values or abrupt change in acceleration are a safety risk [14].

3.4 Experiments

3.4.1 Experiment Setup. We experiment with 10-vehicle platoons and there exists $att \in [1, 2, 3]$ attacker(s) that are randomly located in the member vehicles. Attacker setup can be found in Supplementary Materials Section D.1. We test our trust-based attacker detection models (both single and bi-directional) with acceleration injection attacks. We generate synthetic acceleration data for member vehicles and evaluate trust opinions in real time. Evaluated trust values are saved in $\mathcal{H}$ for long-term record after each trip.

3.4.2 Experimental Results. Fig. 4 shows the experimental results of the single-directional trust model. Trust values of attackers decrease when they perform acceleration attacks. Since our trust framework is aware of long-term history, if no record in $\mathcal{H}$, the initial opinion about vehicle is set to $\{0, 0, 1, 0.5\}$, where uncertainty takes its maximum 1 to represent the fact that we don't know anything (before trip 1). If a vehicle has no history and performs an attack in the beginning of its first trip, then its trust value decreases very fast, e.g., vehicle 1, 2, 3 in trip 1. If a vehicle has good history, and performs an attack or behaves dangerously, then its trust value also decreases but with relatively low rate, e.g., vehicle 9 in trip 5. This is because our framework calculates trustworthiness based on both long-term and short-term history. The longer a vehicle keeps

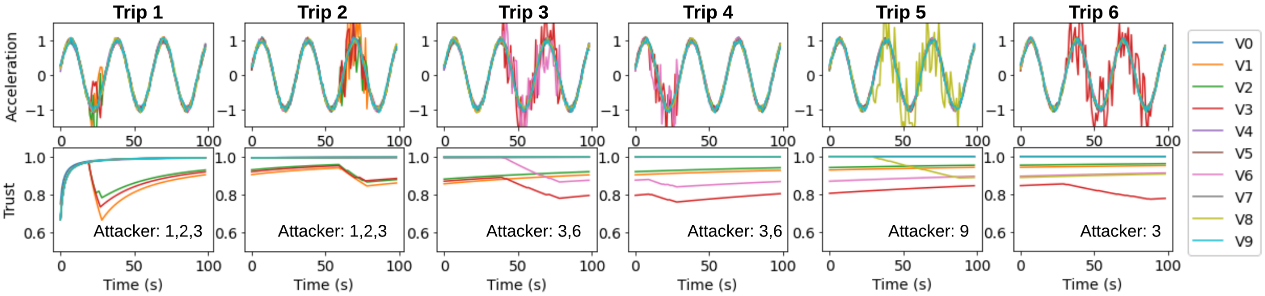


Figure 4: Single-directional attacker detection experimental results. A 10-vehicle platoon completes 6 trips. Assume in thefirst trip all vehicles are new to the trust system and do not have trust record. Their records in $\mathcal{H}$ start building from trip 1 and are used in the following trips. The sine waves are required accelerations, and the fuzzy parts are acceleration attacks performed by vehicles.

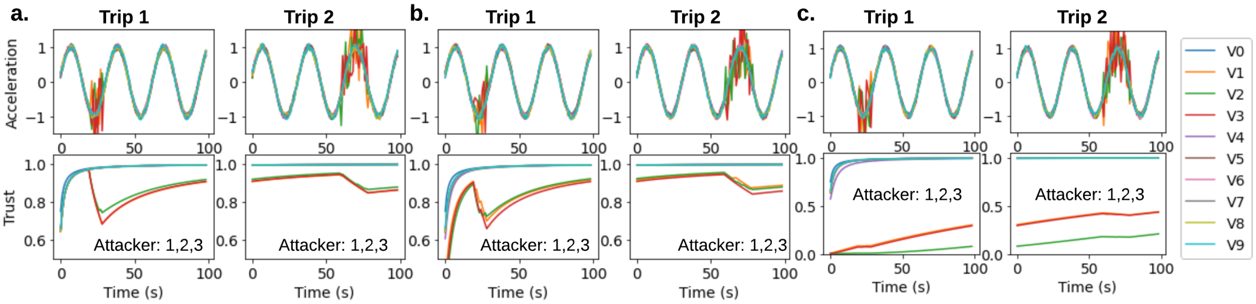


Figure 5: Bi-directional attacker detection experimental results. A 10-vehicle platoon completes 2 adjacent trips and attackers 1, 2, and 3 perform similar accelerations attacks. a. All vehicles do not have trust history in $\mathcal{H}$. b. Only attacker vehicles have moderate histories in $\mathcal{H}$ with trust value 0.25. c. Only attacker vehicles have bad histories with trust value 0.05.

a good record, the slower the penalty comes. On the contrary, when a vehicle with bad history behaves dangerously, its trust value will decrease by a large margin, e.g., vehicle 3 in trip 6.

Fig. 5 shows results of the bi-directional trust model. Different historical trust record of attackers results in different trust values. When attackers with no or moderate histories perform attacks, the trust evaluations and degradation in Fig. 5a-b are similar in Fig. 4. When attackers with bad histories perform good in the current trip, they will gain trust slowly as shown in Fig. 5c. Note that the middle attacker V2 gains trust slower than V1 and V3 because in our bi-directional trust model, V2's evaluators are also untrustworthy, hence their evaluations are discounted by their own trustworthiness.

3.5 Discussion

Our attacker detection model combines long- and short-term trustworthiness history and takes distributed authorities' own trustworthiness into consideration to enable detection of multiple attackers in platoons. One improvement could be differentiating the danger level of attackers by manipulating Eq. 6. A more dangerous behavior (e.g., a behavior leads to crash) should be penalized more than a less dangerous behavior. With this consideration, we will make the trust-based attacker detection be more comprehensive and efficient in follow-up works, such as trust-aware distance control in CACC platoons. In addition, involving trustworthy RSUs is always helpful but costly. With the long-term trust history, vehicles can choose platoons controlled by more trustworthy head vehicles when joining and forming platoons, which is also a meaningful direction in platoon research. In this section we demonstrate the possibility and feasibility of our trust framework, and provide backbones for future works to build on. In the following sections, we will demonstrate how to use the calculated trust values in control policies.

4 TRUST-AWARE AUTONOMOUS INTERSECTION MANAGEMENT (AIM)

4.1 AIM

The intersection traffic in AIM is a simplified version of real-world intersection traffic. Fig. 6a illustrates a four-way intersection example with three lanes in each road leading to the intersection area $\mathcal{I}$ (marked by the white dotted rectangle). A vehicle $X \in \mathcal{X}$ on the road traveling to but not yet entering $\mathcal{I}$ is on the AIM *map* $\mathcal{M}$. The operating procedure of AIM starts from X entering $\mathcal{M}$ and communicating with intersection manager (IM) $\mathcal{A}$ by sending a request Q^X, i.e., vehicle identification number, vehicle size, predicted arrival time, velocity, acceleration, arrival and departure lanes. $\mathcal{A}$ then calculates the trajectory of X and makes a grant or reject decision and sends the decision back to X. $\mathcal{A}$ rejects Q^X if there is a conflict in the simulated trajectories. If $\mathcal{A}$ approves Q^X,

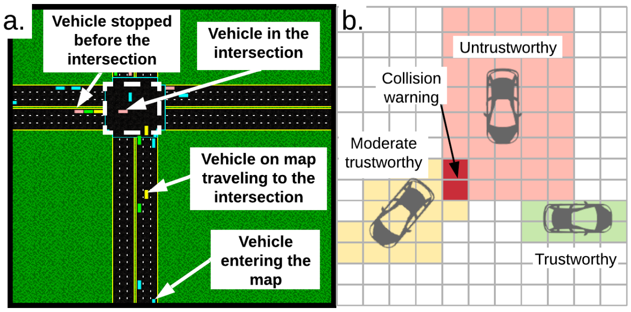

a.
Vehicle stopped before the intersection
Vehicle in the intersection
Vehicle on map traveling to the intersection
Vehicle entering the map
b.
Untrustworthy
Collision warning
Moderate trustworthy
Trustworthy

Figure 6: a. A four-way intersection. b. Space-time buffer examples. Color-shaded areas represent the buffers for each vehicles. Trustworthy vehicles have tight buffer since they are expected to follow the instructions with small error. Untrustworthy vehicles have large buffer because it is highly likely that they would act differently than instructions. Dark red areas represent collision warning in simulated trajectories. In this case the vehicles will not be permitted to enter the intersection and their requests are rejected.

X is responsible for following the instruction to enter and drive through $\mathcal{I}$. In the case of rejection, X has to resend the request and wait for further instructions [12].

Important assumptions that AIM makes are as follows. (i) For all $X \in \mathcal{X}$, they follow the instructions of $\mathcal{A}$ strictly with a range of error tolerance. This restriction guarantees safety by simulating the trajectories and rejecting the conflicted requests. (ii) For all $X \in \mathcal{X}$, they are all attached with a static buffer size, which indicates the time-space reservation of the vehicle. The trajectories are defined as conflicted if the buffers of two vehicles are overlapped (marked as dark red in Fig. 6b, and the shaded area represents the buffer of each vehicle). The larger the buffer size, the higher the safety, and the lower the efficiency or throughput. Note that in conventional AIM, all vehicles' buffer sizes are set to be 1 and this preserves the collision-free because of assumption (i). However, in real world scenarios, assumption (i) is invalid since some of the new drivers or even malicious autonomous agents would act recklessly and not follow the instructions. This leads to collisions in $\mathcal{I}$ and furthermore, small static buffer size in assumption (ii) intensifies the situation.

4.2 RL-based AIM (AIM-RL)

In order to determine optimal buffer sizes for trustworthy and untrustworthy agents to avoid collisions, we use RL to explore the unknown environment. In this section, we define the RL formulation (deep Q-learning [24, 25]) including definitions of states, actions, and rewards, of our proposed RL-based AIM, AIM-RL. In deep Q-learning, the neural network is approximating a Q-learning table, where each entry in the table is updated by $q(s_t, a_t) \leftarrow q(s_t, a_t) + \alpha\,[r_{t+1} + \gamma \max_a q(s_{t+1}, a) - q(s_t, a_t)]$ [10], where s_t is state, a_t is action, r_{t+1} is reward calculated at r_{t+1}, α is learning rate, and γ is discounting factor.

4.2.1 State Space, State Transition, and Action Space. We model a four-way intersection with three lanes in each direction as shown in Fig. 6a. To simulate the real world scenarios, we explicitly allow vehicles on each lane to either go straight, turn left or right. We define our states as $\boldsymbol{s_t} = (v_t^1, e_t^1, q_t^1, ..., v_t^n, e_t^n, q_t^n)^T$, where (v_t^i, e_t^i, q_t^i) are the vehicle identification number, starting point, and requested destination of vehicle $i \in [1, n]$ at time t. In each training time step t, vehicles pass through $\mathcal{I}$ and fully exit $\mathcal{M}$. Within one episode, there are in total τ steps, which represent that the n vehicles pass through τ intersections. $\forall i, t,\ (e_t^i, q_t^i)$ are randomly generated by the simulator. The state transition equations for the environment are defined as: $v_{t+1}^i \leftarrow v_t^i; e_{t+1}^i, q_{t+1}^i \leftarrow Random(v_{t+1}^i)$, where $Random(\cdot)$ is the random starting point and destination generator in simulator.

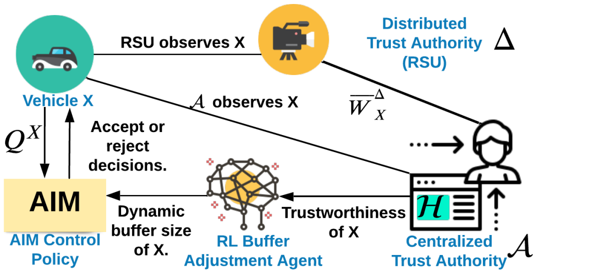


Figure 7: AIM-Trust framework. Combining AIM with RL buffer adjustment agent results in AIM-RL. Adding trust authorities to AIM-RL gives AIM-Trust.

The action at time t is defined as $\boldsymbol{a_t} = (a_t^1, a_t^2, ..., a_t^n)^T$, where a_t^i is the buffer size of vehicle i at time t. Neural network makes prediction by assessing the positions and request of each vehicle.

4.2.2 Reward Function. It is known that throughput is sensitive to buffer size, i.e., large buffer size harms throughput. However, large buffer size leads to low collisions. Therefore, to operate the intersection with low collision rate and maintain throughput, the reward function is defined as:

$$r_t^i = \begin{cases} 1 + \lambda(b_{th} - a_t^i), & \text{if no collision,} \\ -(\tau - 1) * [1 + \lambda(b_{th} - a_t^i)], & \text{if collision happens,} \end{cases} \tag{10}$$

where b_{th} is a hyper parameter indicating a reasonable upper bound buffer size. λ is a hyper parameter to balance the collision and throughput. The vehicle is removed once it collides and not blocking $\mathcal{I}$, and in next steps we put it back in $\mathcal{M}$. A training episode contains τ steps, and an episode ends once reach the maximum τ step.

4.3 Trust-aware RL-based AIM (AIM-Trust)

In this section, we show how to apply our trust framework in AIM and build on AIM-RL to get our trust-aware framework, AIM-Trust. We assume that the IMs serve as the centralized trust manager $\mathcal{A}$, and we have help from RSUs which serve as distributed trust authorities Δ. RSUs cover the places between the intersections while IMs cover intersections. We enhance the AIM framework to be aware of trustworthiness of vehicles and use trust values when inferring the appropriate time-space reservations (buffers) of vehicles to reduce collision rate. AIM-Trust's buffer adjusting agent is similar to AIM-RL and the operation details are shown in Fig. 7.

4.3.1 Trust/Opinion Update. Before entering $\mathcal{M}$, RSUs as distributed Δ update the opinion about X by cumulative fusion: $\overline{W}_X^{\mathcal{A}} \leftarrow \overline{W}_X^{\Delta} \oplus \overline{W}_X^{\mathcal{A}}$. After entering $\mathcal{M}$ and before entering $\mathcal{I}$, IM $\mathcal{A}$ inspects X, observes evidence, and develops short-term opinion about X. Since short-term $\overline{W}_X^{\mathcal{A}}$ is evaluated based on evidence, and to distinguish from long-term $\overline{W}_X^{\mathcal{A}}$, we substitute short-term $\overline{W}_X^{\mathcal{A}}$ with notation $\overline{W}_X^{E_o}$ (E_o represents evidence observed outside $\mathcal{I}$). Then the opinion about X is updated by combining with its long-term opinion: $\overline{W}_X^{\mathcal{A}} \leftarrow \overline{W}_X^{E_o} \oplus \overline{W}_X^{\mathcal{A}}$. Before X entering $\mathcal{I}$, X's trust value is sent to RL-based buffer adjustment agent to calculate the trust-based dynamic buffer. And then AIM control policy uses this dynamic buffer size to determine whether to reject or accept X's request similarly in original AIM. After X exits $\mathcal{I}$, the trust/opinion is updated again based on X's behaviors and collision status in $\mathcal{I}$. Here, $\mathcal{A}$ inspects X's behavior in $\mathcal{I}$ and obtains an evidence-based opinion $\overline{W}_X^{E_i}$, then the opinion about X is updated as: $\overline{W}_X^{\mathcal{A}} \leftarrow \overline{W}_X^{E_i} \oplus \overline{W}_X^{\mathcal{A}}$. $\overline{W}_X^{E_o}$, $\overline{W}_X^{E_i}$, and $\overline{W}_X^{\Delta}$ are evaluated by observations of X's behavior based on Eq. 1. Now we describe how to determine positive (s) and negative (r) evidences. Similarly to Eq. 6, we use a set of AIM-specific rules to specify a driving behavior to be s or r. Before X approaches $\mathcal{M}$, Δ (RSUs) observe X and generate opinion $\overline{W}_X^{\Delta}$. When X arrives $\mathcal{M}$, AIM-Trust uses another set of rules to quantify evidences based on how well X follows instructions and collision status in $\mathcal{I}$.

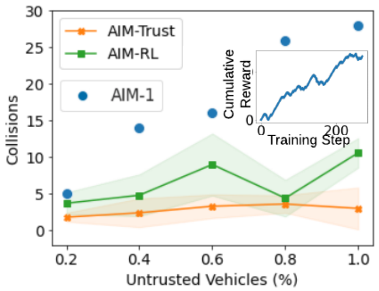
4.3.2 Evidence Evaluation at Road Side Units. Suppose a RSU observes vehicle X's trajectory and velocity, and the desired behavior is defined by a set of rules, e.g., driving within one lane with negligible deviation and under the designated speed limit. Hence, we define these properties formally as follows. Subject to X, suppose the true trajectory is $\boldsymbol{x}^X$, the requested (or predicted) trajectory is $\boldsymbol{y}^X$, the negligible deviation from $\boldsymbol{y}^X$ is $\boldsymbol{\epsilon}$, the speed of the vehicle is $\boldsymbol{sp}^X$, and the designated speed limit is $\hat{sp}$. We quantify r and s as:

$$\begin{aligned} &r = r + 1, \text{ if } (\boldsymbol{x}^X, t) \models \varphi \wedge (\boldsymbol{sp}^X, t) \models \psi; \quad s = s + 1, \text{ otherwise}, \\ &\varphi \equiv \mathrm{G}_{[t_1,t_2]}(|\boldsymbol{x}^X(t) - \boldsymbol{y}^X(t)| \leq \ \boldsymbol{\epsilon}), \psi \equiv \mathrm{G}_{[t_1,t_2]}(\boldsymbol{sp}^X(t) \ \leq \hat{sp}). \end{aligned} \tag{11}$$

This equation indicates that the true trajectory of a vehicle should not deviate from the requested smooth trajectory by more than $\boldsymbol{\epsilon}$ in time interval $[t{+}t_1, t{+}t_2]$, where t_1 and t_2 are hyper parameters. For a RSU, it predicts a smooth trajectory $\boldsymbol{y}^X$ tofi t the true trajectory $\boldsymbol{x}^X$. For AIM-Trust, $\boldsymbol{y}^X$ is the requested and approved trajectory of X. In addition, the speed of a vehicle should never exceed the speed limit in time interval $[t_1, t_2]$.

4.3.3 Evidence Evaluation at $\mathcal{M}$. When vehicles enter $\mathcal{M}$ before entering $\mathcal{I}$, Eq. 11 is used to quantify evidences. If negative evidence observed, it means the vehicle violates the approved trajectory by an intolerable error. Once vehicles enter $\mathcal{I}$, a new set of rules is used to take into account the honest status of vehicles and collisions in $\mathcal{I}$: $r = r + 1$, if the vehicle follows the approved trajectory and no collision happens; otherwise $s = s + 1$.

4.3.4 Trust-based RL Buffer Adjustment Agent. We formulate RL of AIM-Trust similarly to AIM-RL, and add only trustworthiness of vehicles in state space: $\boldsymbol{s_t} = (v_t^1, e_t^1, p_t^1, q_t^1, ..., v_t^n, e_t^n, p_t^n, q_t^n)^T$, where p_t^i is i's trustworthiness at time t. With this minor modification, we achieve trust-aware RL-based intersection management.



UV	AIM-RL	AIM-1
20%	51.35%	64.00%
40%	50.00%	82.85%
60%	64.44%	79.37%
80%	18.18%	83.84%
100%	71.69%	89.28%

Figure 8 & Table 1: Collision comparison between AIM-Trust, AIM-RL and AIM-1. In-plot shows the cumulative reward of AIM-Trust with 20% untrusted vehicles in training. Table 1: Performance improvement of AIM-Trust compared to baselines. UV indicates the untrusted vehicle percentage.

4.4 Experiments

4.4.1 Experiment Setup. We consider in one RL training episode, $n = 10$ vehicles passing through $\tau = 10$ intersections and we monitor the collisions happened within these $n\tau = 100$ intersection crossing events as c. For each intersection (or step) t in an episode, n vehicles enter and leave the intersection following randomly picked starting points, $[e_t^0, e_t^1, ..., e_t^n]$, and destinations, $[q_t^0, q_t^1, ..., q_t^n]$. We compare AIM-Trust with the original AIM algorithm withfi xed buffer size 1, namely AIM-1, and AIM-RL. We insert 20-100% untrusted vehicles in traffic to cause potential collisions. All simulations are done in the AIM simulator [1]. See Supplementary Materials Section D.2 for the selection of hyper-parameters and further experimental setup details. We train both AIM-Trust and AIM-RL 10 times and report the mean-variance results. In addition, we control the training process of AIM-RL and AIM-Trust to be the same to ensure fair comparison.

4.4.2 Experimental Results. As shown in Fig. 8, RL-based AIM-RL and AIM-Trust decrease the collision numbers drastically compared to AIM-1 for two reasons: (i) AIM-1 cannot deal with untrusted vehicles and the small andfi xed buffer size results in high collision rate. (ii) AIM-Trust and AIM-RL takes collision numbers in reward function to penalize collision and allocate appropriate time-space buffer intelligently for different vehicles. Table 1 shows the average improvements of AIM-Trust in terms of percentage. Compared to non-trust AIM-RL, AIM-Trust decreases collisions by at least 19.18% and up to 71.69%. These results demonstrate the effectiveness of the proposed trustworthiness and trust framework in control policies.

Since our formulation of RL reward function considers throughout, and higher throughput leads to lower safety, with Eq. 10, AIM-Trust cannot guarantee collision-free due to the trade-off between safety and throughput. To demonstrate that AIM-Trust can deduce appropriate buffer sizes based on trustworthiness, we suppress the performance on throughput and let $\lambda = 0$ while giving big penalty (-40) when collision happens. (See Supplementary Materials Section D.3 for detailed formulation.) It turns out AIM-Trust with revised reward function learns to select large buffer sizes and achieves collision-free in all scenarios.

5 TRUST-AWARE TRAFFIC LIGHT CONTROL

5.1 RL-based Traffic Light Control (TLC-RL)

Conventional fixed-cycle traffic light control (TLC) has many disadvantages, such as energy waste and long delays [20]. Many RL-based adaptive TLC frameworks have been proposed in recent years to take real-time traffic information as input and adjust traffic light dynamically [9, 20, 21, 29, 30]. Here, we use an example RL formulation of such problems following [20] and later we show how to make it trust-aware by a minimal modification. We use deep Q-learning to dynamically control the traffic light. Detailed RL formulation can be found in Supplementary Material Section D.4.

5.2 Trust-aware TLC (TLC-Trust)

With the assumption that there might be untrustworthy/malicious vehicles, trust evaluation is helpful for intersection management. To enable the traffic light controller with trust information, we utilize our proposed trust framework where the centralized trust manager $\mathcal{A}$ maintains a trustworthiness table $\mathcal{H}$. The trust framework for TLC is similar to AIM-Trust as described in Section 4.3.1. Note that distributed trust authorities, e.g., RSUs in traffic systems, are not necessary in our trust framework, however, they can help $\mathcal{A}$ to enlarge the observation range and maintain an accurate $\mathcal{H}$. With $\mathcal{A}$, we have access to trustworthiness of vehicles. Hence, with a minimum modification to the state space in RL formulation, i.e., adding vehicles trustworthiness q_t^i gives us TLC-Trust.

5.3 Experiments

5.3.1 Experiment Setup. In this case study, we compare TLC-RL and its trust-aware version, TLC-Trust. In addition, we also compare with conventional fixed-cycle traffic light policy (TLC-Fix). Similarly in AIM experiments, we insert $20\% - 100\%$ untrustworthy vehicles in the traffic and we assume that untrustworthy vehicles may turn left even they are at right lane. In one episode, 100 vehicles pass through the intersection $\mathcal{I}$ and we use collision rate as evaluation metric. We run TLC-RL and TLC-Trust 10 times and report mean-variances to ensure fair comparison.

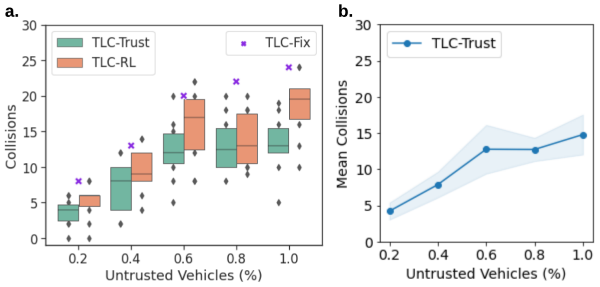


Figure 9: a. Collision comparison between TLC-Trust, TLC-RL and TLC-Fix. b. Mean collisions of TLC-Trust in 10 test cases, each test case runs 10 times to calculate the mean collision. TLC-Trust's collision rates in test and training sets are consistent.

5.3.2 Experimental Results. The collision comparison between TLC-Trust, TLC-RL, and TLC-Fix is illustrated in Fig. 9. Compared to non-trust TLC-RL, TLC-Trust achieves lower collisions in all cases. In addition, TLC-Trust and TLC-RL are more advanced than traditional fixed-length TLC-Fix. Since the experimental setup in TLC and AIM case studies are similar, we compare TLC also with AIM methods. As shown in Fig. 8 and 9, AIM methods are much better at collision reduction compared to TLC methods, this is because AIM-Trust and AIM-RL are designed to do collision avoidance and allocate a trust-based space-time buffer to each vehicle, whereas the traffic light-based control policies are focusing on intersection efficiency. To reduce collision rate of TLC-Trust, we can add penalties in reward function when collision happens. Besides these results, we also provide videos to demonstrate how TLC-Trust works in AIM intersection simulator: https://youtu.be/15heSQbWHtE.

5.4 Discussion

Trust-aware intersection management using traffic signals shows better performance in terms of collision rate compared to the non-trust version. We envision that further advantages of trust-aware intersection management lie in cooperative intersection management where multiple adjacent intersections cooperate and control traffic lights aware of dangerous and untrustworthy vehicles. In addition, emergency vehicles can benefit from this setup since we can give priority through trust values.

6 CONCLUSION

In this work we propose a general trust framework for MASs and demonstrate the feasibility and advantages by applying it to three different systems. We show that with trust evaluations, the designed trust-based CACC platoon attacker detection model accurately detects attackers and the historical trust record can be further useful in future platoon formation and platoon control. In the intersection management case studies (AIM and TLC), we show that by embedding trust quantification into decision-making algorithms, the coordination algorithms can balance safe coordination with system performance even in mixed trust settings. In particular, we show that the trust-aware versions of Autonomous Intersection Management (AIM-Trust) and Traffic Light Control (TLC-Trust) outperform their trust-oblivious counterparts in terms of improving safety of the overall MAS. In future, we will consider applying the trust framework to broader classes of MAS algorithms such as those used for consensus and multi-agent path planning.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the National Science Foundation under the Career Award CPS/CNS-1453860, the NSF award under Grant Numbers CCF-1837131, MCB-1936775, CNS-1932620, and CMMI 1936624 and the DARPA Young Faculty Award and DARPA Director's Fellowship Award, under Grant Number N66001-17-1-4044, a Northrop Grumman grant, and a grant from Toyota R&D North America. The views, opinions, and/or findings contained in this article are those of the authors and should not be interpreted as representing the official views or policies, either expressed or implied by the Defense Advanced Research Projects Agency, the Department of Defense or the National Science Foundation.

REFERENCES

[1] [n.d.]. AIM4 1.0-SNAPSHOT API. http://www.cs.utexas.edu/~aim/aim4sim/aim4-release-1.0.3/aim4-root/target/site/apidocs/index.html. Accessed: 2020-07-26.

[2] Faisal Alkhateeb, Eslam Al Maghayreh, and Shadi Aljawarneh. 2010. A multi agent-based system for securing university campus: Design and architecture. In *2010 International Conference on Intelligent Systems, Modelling and Simulation*. IEEE, 75–79.

[3] Mani Amoozadeh, Arun Raghuramu, Chen-Nee Chuah, Dipak Ghosal, H Michael Zhang, Jeff Rowe, and Karl Levitt. 2015. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine* 53, 6 (2015), 126–132.

[4] Tsz-Chiu Au, Neda Shahidi, and Peter Stone. 2011. Enforcing liveness in autonomous traffic management. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.

[5] Tsz-Chiu Au, Shun Zhang, and Peter Stone. 2015. Autonomous intersection management for semi-autonomous vehicles. In *Routledge Handbook of Transportation*. Routledge, 116–132.

[6] Jakob Axelsson. 2016. Safety in vehicle platooning: A systematic literature review. *IEEE Transactions on Intelligent Transportation Systems* 18, 5 (2016), 1033–1045.

[7] Eric K Butler, Anca A Chandra, Pawan R Chowdhary, Susanne M Glissmann-Hochstein, Thomas D Griffin, Divyesh Jadav, Sunhwan Lee, and Hovey R Strong Jr. 2017. Drone air traffic control andfl ight plan management. US Patent 9,852,642.

[8] Mingxi Cheng, Shahin Nazarian, and Paul Bogdan. 2020. There Is Hope After All: Quantifying Opinion and Trustworthiness in Neural Networks. *Frontiers in Artificial Intelligence* 3 (2020), 54.

[9] Denise de Oliveira, Ana LC Bazzan, Bruno Castro da Silva, Eduardo W Basso, Luis Nunes, Rosaldo Rossetti, Eugénio de Oliveira, Roberto da Silva, and Luis Lamb. 2006. Reinforcement Learning based Control of Traffic Lights in Non-stationary Environments: A Case Study in a Microscopic Simulator.. In *EUMAS*.

[10] Avinash K Dixit, John JF Sherrerd, et al. 1990. *Optimization in economic theory*. Oxford University Press on Demand.

[11] Kurt Dresner and Peter Stone. 2004. Multiagent traffic management: A reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*. 530–537.

[12] Kurt Dresner and Peter Stone. 2008. A multiagent approach to autonomous intersection management. *Journal of artificial intelligence research* 31 (2008), 591–656.

[13] Bill Fleming. 2011. New Innovative ICs [Automotive Electronics]. *IEEE Vehicular Technology Magazine* 6, 2 (2011), 4–8.

[14] Keno Garlichs, Alexander Willecke, Martin Wegner, and Lars C Wolf. 2019. TriP: Misbehavior Detection for Dynamic Platoons using Trust. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 455–460.

[15] Andreas Geiger, Martin Lauer, Frank Moosmann, Benjamin Ranft, Holger Rapp, Christoph Stiller, and Julius Ziegler. 2012. Team AnnieWAY's entry to the 2011 grand cooperative driving challenge. *IEEE Transactions on Intelligent Transportation Systems* 13, 3 (2012), 1008–1017.

[16] Siyuan Gong, Anye Zhou, and Srinivas Peeta. 2019. Cooperative adaptive cruise control for a platoon of connected and autonomous vehicles considering dynamic informationfl ow topology. *Transportation Research Record* 2673, 10 (2019), 185–198.

[17] V David Hopkin. 2017. *Human factors in air traffic control*. CRC Press.

[18] Audun Jøsang. 2016. *Subjective logic*. Springer.

[19] Jakub Konečnỳ, Brendan McMahan, and Daniel Ramage. 2015. Federated optimization: Distributed optimization beyond the datacenter. *arXiv preprint arXiv:1511.03575* (2015).

[20] Xiaoyuan Liang, Xunsheng Du, Guiling Wang, and Zhu Han. 2019. A deep reinforcement learning network for traffic light cycle control. *IEEE Transactions on Vehicular Technology* 68, 2 (2019), 1243–1253.

[21] Weirong Liu, Gaorong Qin, Yun He, and Fei Jiang. 2017. Distributed cooperative reinforcement learning-based traffic signal control that integrates V2X networks' dynamic clustering. *IEEE transactions on vehicular technology* 66, 10 (2017), 8667–8681.

[22] O. Maler and D. Nickovic. 2004. Monitoring Temporal Properties of Continuous Signals. In *FORMATS/FTRTFT*.

[23] Vicente Milanés, Steven E Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, and Masahide Nakamura. 2013. Cooperative adaptive cruise control in real traffic situations. *IEEE Transactions on intelligent transportation systems* 15, 1 (2013), 296–305.

[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013).

[25] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[26] Muaz Niazi and Amir Hussain. 2011. Agent-based computing from multi-agent systems to agent-based models: a visual survey. *Scientometrics* 89, 2 (2011), 479–499.

[27] Guni Sharon and Peter Stone. 2017. A protocol for mixed autonomous and human-operated vehicles at intersections. In *International Conference on Autonomous Agents and Multiagent Systems*. Springer, 151–167.

[28] Rens van der Heijden, Thomas Lukaseder, and Frank Kargl. 2017. Analyzing attacks on cooperative adaptive cruise control (CACC). In *2017 IEEE Vehicular Networking Conference (VNC)*. IEEE, 45–52.

[29] Elise Van der Pol and Frans A Oliehoek. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)* (2016).

[30] Hua Wei, Guanjie Zheng, Huaxiu Yao, and Zhenhui Li. 2018. Intellilight: A reinforcement learning approach for intelligent traffic light control. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2496–2505.