Detecting and Mitigating Colluding Attacks in Connected Vehicles using Reinforcement Learning

Pranay Chowdary Jasti
Department of Computer Science
Texas State University
San Marcos, Texas
xxp12@txstate.edu

Henry Griffith
Department of Engineering
San Antonio College
San Antonio, Texas
hgriffith5@alamo.edu

Heena Rathore
Department of Computer Science
Texas State University
San Marcos, Texas
heena.rathore@txstate.edu

2024 IEEE International Conference on Consumer Electronics (ICCE) | 979-8-3503-2413-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCE59016.2024.10444185

***Abstract*—As the vehicles are interconnected and share infrastructure information among each other, the adoption of connected vehicles (CVs) continues to surge. However, CV's introduce vulnerabilities related to the integrity of the shared data, which may be compromised by either malicious attacks or sensor failures. Detecting these vulnerabilities in CVs has become paramount to provide safety to the passengers and pedestrians in the network. Trust and reputation-based reinforcement learning (RL) algorithms are one way for the detection and handle these vulnerabilities. These algorithms fail to work when the number of vehicles collude together to disrupt the CV network. In response to this, we propose a trust and reputation based RL along with multi level dempster shafer technique to deal with colluding attacks in CVs. This integration involves fusing reputation-based trust management on a vehicle level with a RL agent running on road side unit. We conduct performance analysis on different RL algorithms namely deep Q-networks, actor-critic and proximal policy optimization.**

***Index Terms*—connected vehicles, security, reinforcement learning, colluding attacks.**

I. Introduction

Connected vehicles (CVs) show promising results in improving road safety, efficient fuel consumption, and significantly decreasing traffic congestion, representing a pivotal development in modern transportation [1]. However, network vulnerabilities can give rise to attacks in CV which can lead to adverse impacts, posing a viable risk to passengers' lives [2]. Attacks such as information manipulation, data injection, and data falsification are classified as internal attacks and can be particularly challenging to detect and mitigate due to the attackers' deeper knowledge of the system's internals [3]. External attacks include denial of service attacks, spoofing attacks, man-in-the-middle attacks, and others. Existing models to overcome external attacks are effective [4], but internal attacks can remain undetected within these existing models due to their increased sophistication [5].

Internal attacks such as information falsification have several solutions including trust reputation models [3], there is limited work for colluding vehicle attacks [6]. Colluding attacks involve the cooperation of multiple malicious vehicles. State-of-the-art solutions for these attacks employ block-chain based approach [7], trust management system [8] and others [9]. While well-crafted colluding attacks have the potential to compromise most of the existing state-of-the-art frameworks for CV, there is a notable lack of research efforts focused on their identification and mitigation. A well-crafted attack can stay undetected in CV environments [6], thus, there is a need for an efficient framework for the detection and mitigation of such colluding attacks.

Recently, reinforcement learning (RL) models [10] have demonstrated their superiority in adapting to CV dynamic environments [11]. RL along with trust and reputation models are designed to handle dynamic vehicle traffic and incentivize good behavior while penalizing undesirable conduct. This can be achieved by tracking and updating reputation based on current trust scores. In our work, we propose the utilization of a multi-level Dempster-Shafer (DS) approach with a dynamic reputation update policy using RL for the detection and mitigation of colluding attacks.

This paper serves as an extension of our prior work [12], where we employed a Deep Q-based dynamic reputation update policy in conjunction with a multi-level Dempster-Shafer model. We implemented several enhancements compared to our prior work: (1) The attacker model initiated attacks when the reputation policy favored trust over reputation. (2) We equipped the attacker with the ability to emulate colluding attacks instead of traditional data falsification attacks. (3) We adopted the final malicious score as a state value for training the RL agent. (4) Our study encompassed a comprehensive comparison among various RL algorithms, including deep Q-networks (DQN), actor-critic (AC), and proximal policy optimization (PPO).

II. Background and Related Work

A. Related Work

Authors in [6] investigates the effectiveness of deep RL based adaptive traffic controller system for mitigating colluding attack. The authors used deep RL with negative waiting time as a reward function to reduce the overall wait time of the colluding vehicles. The simulations were carried out with the SUMO traffic simulation tool [13] to generate traffic resembling the Monaco city traffic and used ablation study and sensitivity analysis to evaluate the average travel time and average wait time for both colluding and normal vehicles. The findings reveal that the proposed framework was able to reduce the total wait time of colluding vehicles by 92% and increased the total travel time of normal vehicles by 62% causing them to spend 12% more time on travel.

In [7], the authors investigate the use of ledger-based blockchain aiming at building a trustworthy CV system against threats targeting human safety. The authors proposed an efficient interaction verification system to avoid system attacks using proof of interaction with a lightweight mining algorithm. Gathering the values of average number of blocks and the time required for miner detection values, generated from simulations resembling an urban CV environment, proves that the model is resilient to different threats such as black hole miners, bad mouthing, and malicious miners. In their discussion authors described that the proposed model will not allow a change that's greater than 10% in one mining cycle which protects the system from the substantial effect of a colluding attack.

In [8], the authors proposed the use of interaction provenance as a proof of interaction considering the chronological order of historical interactive events along with fuzzy ranges, aggregated weights, and trustworthiness profile mapping. The author provides information related to the effectiveness of the proposed algorithm and how adjusting the fuzzy ranges, can be used to mitigate colluding attacks.

In [9], the authors investigate wormhole protocol detector (WPD), a lightweight protocol for detecting and mitigating wormhole attacks. The author's WPD monitors, detects out-of-range packets, identifies the nodes participating in the wormhole connection, avoids the wormhole links, and obtains secure routing paths between CVs. The findings reveal that the module can effectively identify wormhole attacks using the WPD with 100% accuracy in different length of wormhole link lengths and it failed to identify the encapsulation wormhole attack, as the attacker have the ability to tamper with the information. In this paper, we propose a novel approach for detecting and mitigating colluding attacks in CV. The following are the contributions:

- Current state-of-the-art solutions lack a clearly defined profile for colluding attacks. In our research, we have established a distinct and well-defined profile for colluding attacks.
- We propose a multi-level DS technique along with RL algorithm for mitigating colluding attacks which fit well for dynamic CV networks.
- Existing solutions employ DQN for reputation-based security. We conducted a comprehensive comparative analysis among state-of-the-art RL algorithms. For this, we simulated our work on RL based environment for CV networks [15], [16].

III. Proposed Work

A. Dataset for Vehicular Network Simulation

We used an open-source vehicle navigation model controlled by multi-agent RL framework [15], [16] where the navigation of the vehicles is governed in an on-ramp merging scenario simulating general traffic conditions. Each vehicle shares basic safety messages (BSM) which contain details about its position, speed, headed direction, and acceleration. Each vehicle calculates its peer reputation by comparing the BSM reported by the vehicle with the reports generated through its own sensor. We assume that each vehicle is equipped with necessary sensors which help them to make peer reports.

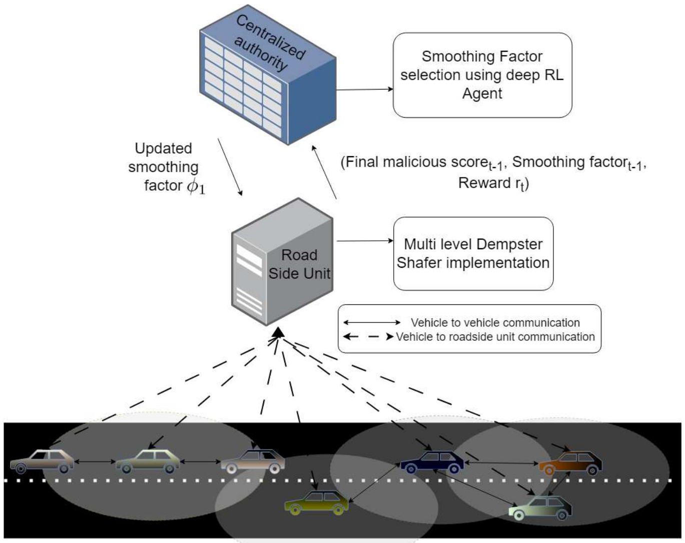

Centralized authority
Smoothing Factor selection using deep RL Agent
Updated smoothing factor ϕ_1
(Final malicious score$_{t-1}$, Smoothing factor$_{t-1}$, Reward r_t)
Road Side Unit
Multi level Dempster Shafer implementation
Vehicle to vehicle communication
Vehicle to roadside unit communication

Fig. 1: System Model

B. System Model

The system architecture (Figure 1) combines decentralized validation with centralized feedback, integrating reputation-based trust management with an RL at the centralized authority (CA). A multi-level implementation of DS at the Roadside Unit (RSU) predicts false reports from vehicles, influencing the Deep RL agent's rewards. It starts with inter-platoon communication, where vehicles transmit BSM, sensed by peer vehicles. Decentralized reputation calculation at the vehicle level sends reputation scores to the RSU, with dynamic reputation updates (denoted as smoothing factor ϕ_1) assigned by the CA. The RSU employs DS theory to combine vehicle reputation reports, considering both peer-reported behavior and self-reported values with centralized reputation and predict the final malicious score. An average final malicious score, previous smoothing factor ϕ_1, and previous-time rewards are fed into the RL agent, which determines an optimal smoothing factor, which is disseminated across the network. This factor encourages malicious vehicles to report truthfully, while a robust reputation update policy continuously evaluates each vehicle's overall behavior.

C. Attacker model

In our model, we implement a colluding attack by falsifying the data across a group of vehicles. This is accomplished by adding white Gaussian noise to the sensed kinematics of the peer vehicles as shown in Figure 2. Here, vehicles within

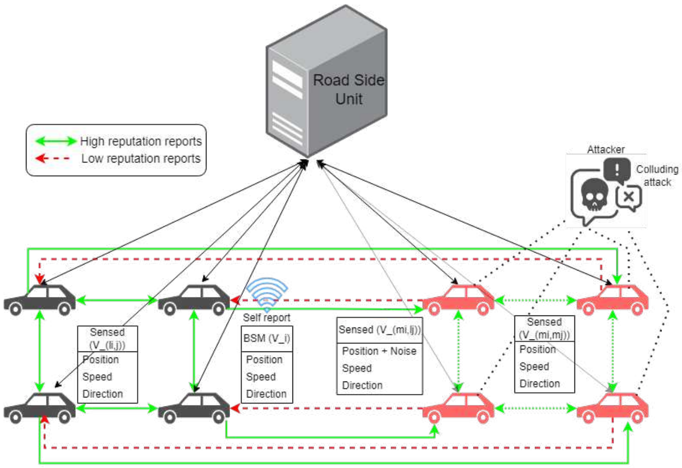


Fig. 2: Colluding attack where black vehicles are normal vehicles and red are malicious vehicles forming two groups.

the colluding group (shown in red) give high reputation to other vehicles participating in the same group. However, they deliberately inject noise into the sensed coordinates of the normal vehicles that are in close proximity to them (shown in black). This strategic action aims to diminish the reputation of the normal vehicles at the RSU, with the underlying goal of falsely implicating them as malicious vehicles. The white Gaussian is generated with $\mu = 2$ and $\sigma = 0.9$. The noise is introduced by intercepting the smoothing function $(\phi_1) \geqq 0.4$, at RSU.

D. Dempster-Shafer Technique

The DS theory of evidence is a mathematical framework for quantifying belief in statements by combining independent evidence from multiple sources using belief functions. It handles uncertainty by assigning degrees of belief to subsets of possible events and uses the DS rule to aggregate belief functions.

a) Frame of discernment Θ*:* Frame of discernment is defined as $\Theta = \{\Theta_1, \Theta_2 \ldots\ldots \Theta_n\}$ that covers individual, mutually exclusive, discretized values of all viable outcomes of Θ. In our approach Θ contains two elements $\Theta = \{M, N\}$; where M= malicious vehicle, N= normal vehicle.

b) Power Set P(Θ*):* The power set P(Θ) of aforesaid random variable Θ is a set of all subsets of Θ including the individual elements, and represents the DS frame of Θ. For our model it contains P(Θ)=$\{\phi, M, N, MN\}$.

c) Evidence: Evidences are events/symptoms and one evidence maps to single hypothesis or set of hypotheses.

d) Mass Function (m-value): Our trust regarding the truth value of a proposition 'A' is dependent on the evidence that supports the proposition which is denoted as mass function (m-value). It relates to the weights of the elements in the P(Θ), $m : 2^{\Theta} \to [0, 1]$, where weight of the null set is 0, $m(\phi) = 0$ and $m(A) \geq 0$. The overall sum of the mass functions of all elements in the P(Θ)=1 or $\sum\{m(A) \ \forall \ A \ \in \ 2^{\Theta}\} = 1$. Thus, $m(A)$ is a measure of belief assigned by a given evidence to A, where A is any element of 2^{Θ}, $\forall A \in 2^{\Theta}$, and non belief is forced by the lack of knowledge. We can get the lower and upper bound of an interval from the mass function. The lower bound is used as the belief function and the upper bound is used to calculate the plausibility function.

e) Plausibility function (Pl): The upper bound of the interval is called plausibility, and it is determined by taking the sum of all the mass function of the subsets (B) that intersects (A) where $(B \cap A \neq \phi)$, $Pl(A) : 2^{\Theta} \to [0, 1]$ [14].

$$Pl(A) = \sum_{B \cap A \neq \phi} m(B) \tag{1}$$

f) DS Rule of Combination: The data collected from the different sources are combined rationally, to focus on the consensus opinion and use normalization to ignore all the conflicting evidences. A cartesian product of two mass functions is employed for the combination of evidence. The DS combination rule determines the joint m_{12} from the combination of two mass functions using equation:

$$m_{12}(A) = \frac{\sum_{B \cap C = A}\{m_1(B) m_2(C)\}}{1 - K} \tag{2}$$

when $A \neq \phi$, $m(\phi) = 0$ and $K = \sum_{B \cap C = \phi} m_1(B) m_2(C)$

E. Multi Level DS

1) Level-1 Plausibility Calculation: Level 1 corresponds to decentralized reputation calculation at the vehicle level. Each vehicle is equipped with the capability to sense the kinematics of its peers. Each vehicle calculates the difference between the sensed kinematics and the broadcasted BSMs of peer vehicle (j). If this disparity exceeds a predefined threshold noise, the vehicle (i) assigns a trust value of 0.1 to j; otherwise, it assigns a trust value of 0.9.

$$T^i_{j,t} = \begin{cases} 0.1 & If \quad \Delta > 0.2 \\ 0.9 & Else \end{cases} \tag{3}$$

where $\Delta = x^j_j - x^i_j$ and x is the position reported. These trust values are then integrated with the current reputation score with the current smoothing factor (denoted as ϕ_1):

$$R^i_{j,d_t} = \phi_1 R^i_{j,d_{t-1}} + (1 - \phi_1) T^i_{j,t} \tag{4}$$

Each vehicle then share the calculated reputation scores to RSU. The reputation scores shared by peer vehicles serve as the basis for building mass functions. The RSU treats the reputation reports provided by different vehicles as individual pieces of evidence. Specifically, one mass function characterizes the likelihood of a vehicle being normal (N), while the other, its complement (1 - reputation score), represents the likelihood of a vehicle being malicious (M). For each vehicle's reputation report, the RSU generates a mass function tuple consisting of $m(N)$ (the mass function for normal behavior), $m(M)$ (the mass function for malicious behavior):

$$m_{1i}(N)^{RSU} = R^j_{i,d_t} \tag{5}$$

$$m_{1i}(M)^{RSU} = 1 - R^j_{i,d_t} \tag{6}$$

The generated mass functions from the reputation reports shared by the peer vehicles are combined using Eq. 2 which is later used to calculate the plausibility values of vehicles being malicious and normal. The plausibility values generated at this stage are referred to as level-1 plausibility values where $Pl_{1,j}(M)$ and $Pl_{1,j}(N)$ represent the vehicle being malicious and normal respectively.

2) Level-2 Plausibility Calculation: In this step, RSUs generate mass functions by calculating the difference between the reputation report submitted by the vehicle i for peer vehicle j and the level-1 plausibility value of the peer vehicle j being normal indicating the vehicle's malicious behavior.

$$m_{2i}(M)^{RSU} = \begin{cases} |R^i_{j,d_t} - Pl_j(N)^{RSU}| \quad If \quad |R^i_{j,d_t} - Pl_j(N)^{RSU}| > 0.2 \\ 0.1 \quad Else \end{cases} \tag{7}$$

$$m_{2i}(N)^{RSU} = 1 - m_{2i}(M) \tag{8}$$

This validation process is iteratively applied to each report submitted by a vehicle for its peer vehicles. The plausibility values from this step are considered as the level-2 plausibility values denoted as $Pl_{2,i}(M)$ and $Pl_{2,i}(N)$.

3) Random Validation by RSU:

When the proportion of malicious vehicles surpasses 50%, attackers succeed in their objective of diminishing the reputation of peer vehicles at the CA. To counteract this vulnerability, RSU intermittently intercepts vehicle communications every 20 discrete intervals and maintains this policy for the next 20 iterations. RSU validates vehicle self-reports against their own sensed kinematic data to establish vehicle reputations.

$$R^{RSU}_{i,c_t} = \phi_2 R^{RSU}_{i,c_{t-1}} + (1 - \phi_2) T^{RSU}_{i,c_t} \tag{9}$$

Here, ϕ_2 is used with a default constant value of 0.2. RSU accumulates reputation reports that vehicles share regarding their peers. The acceptance of reputation reports despite variance is grounded in the disparate nature of trust update policies employed by vehicles and RSU. When the variance follows a descending order, a vehicle is attributed a trust level of 0.9; otherwise, a trust level of 0.1 is assigned:

$$T^{RSU}_{i,c_t} = \begin{cases} 0.1 \quad If \quad \Delta > 0.2 \\ 0.9 \quad Else \end{cases} \tag{10}$$

$\Delta = x_i^{RSU} - x_i^i$. Later using Eq. (5) and (6) the centralized mass functions $(m_{3,i}(M))$ and $(m_{3,i}(N))$ are calculated.

F. Combination of Level-1, Level-2 and RSU Validated Scores

RSU merges information from Level 1, Level 2, and RSU calculated reputation to predict vehicle behavior. Employing the DS theory of combination, RSU fuses these mass functions to determine the final plausibility of a vehicle being malicious and normal using Eq. 1. When the final plausibility of a vehicle's malicious behavior exceeds 0.3, the vehicle is identified as malicious, rendering all its transmitted reputation reports as malicious. These predictions serving as the foundation, contribute to the formulation of rewards for the RL agent located at CA. This multi-stage process not only bolsters the integrity of the vehicular network against malicious behavior but also showcases the integration of RL to incentivize reliable reporting. The final plausibility of the vehicle's malicious behavior is tracked using a robust trust update policy (11) as described in [17]. This value serves as the final malicious score of the vehicle which is used as an indicator of vehicle behavior at the RSU level. $M^{RSU}_{i,t}$ is the cumulative malicious score, $Pl^{RSU}_{i,t}$ is the final plausibility score of the vehicle being malicious and the values of D, θ and σ are set as 200, 2, 20 respectively.

$$M^{RSU}_{i,t} = M^{RSU}_{i,t-1} + \frac{1}{\theta}\Phi(M^{RSU}_{i,t-1}) D (Pl^{RSU}_{i,t} - E^{RSU}_{i,t-1}) \tag{11}$$

$$\Phi(M^{RSU}_{i,t-1}) = 1 - \frac{1}{1 + \exp\left(\frac{-(M^{RSU}_{i,t-1} - D)}{\sigma}\right)} \tag{12}$$

$$E^{RSU}_{i,t-1} = (M^{RSU}_{i,t-1})/D \tag{13}$$

G. RL model at CA

In our model we have tailored the RL agent on CA to select the optimal smoothing factor. The state s_t contains previous smoothing factor ϕ_1, average final plausibility score of the vehicles being malicious (obtained from Level 3 DS at RSU) and previous reward value (re).

$$s_t = \phi_{1,t-1}, Pl^{RSU}_{t-1\ avg}, re_{t-1} \tag{14}$$

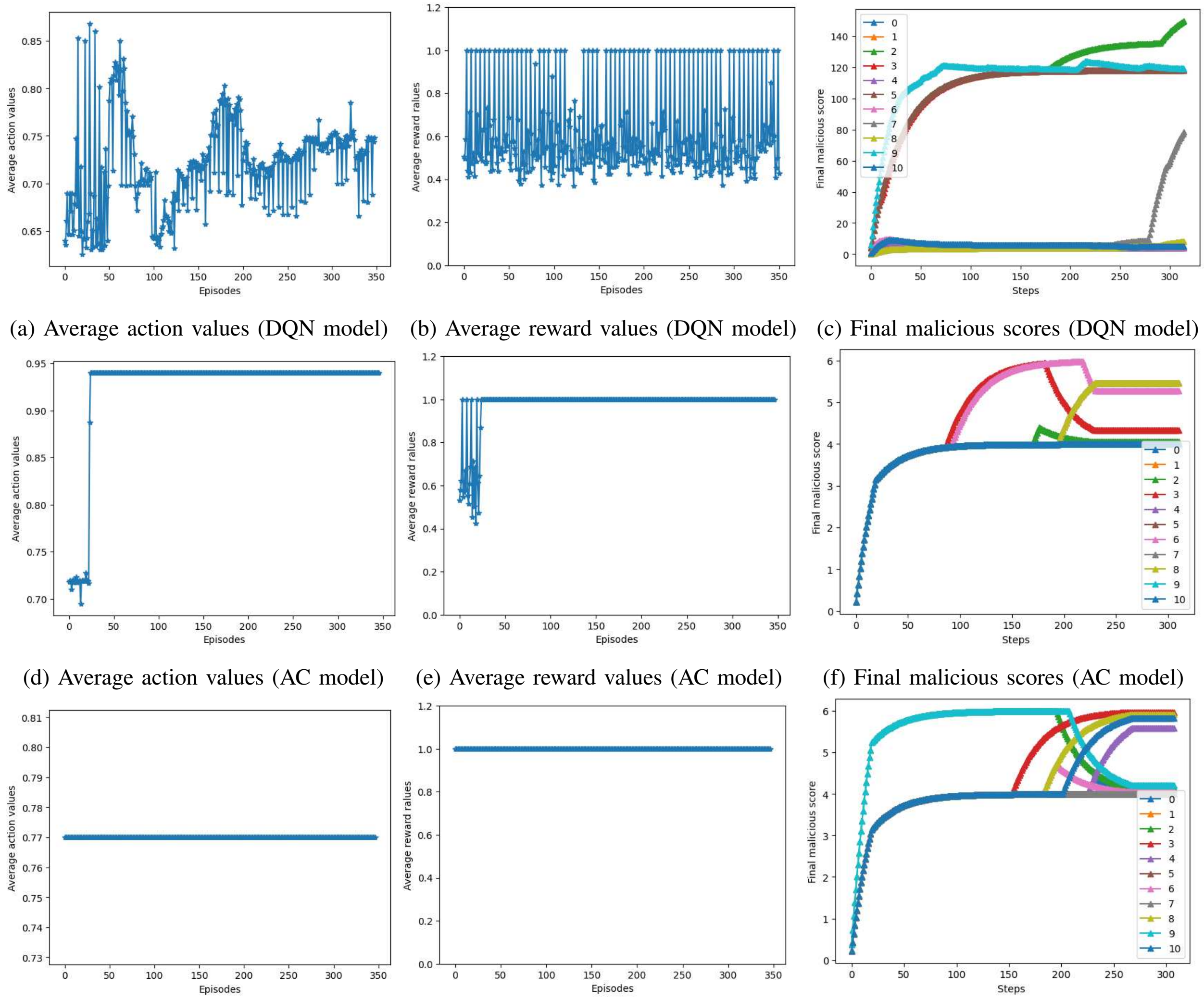
The reward is computed as the ratio of the number of true reputation reports to the total number of reputation reports computed by RSU. Subsequently, this reward is furnished to the RL agent located in CA, which leverages it to make informed selections of actions, represented as smoothing factors.

$$a_t = \phi_1 = \{\phi_{1_1}, \phi_{1_2}, \phi_{1_3}, ..., \phi_{1_n}\} \tag{15}$$

The overarching aim of these actions is to maximize the cumulative reward. In a comprehensive loop, the actions determined by the RL agent are disseminated to the vehicles, effectively compelling them to transmit accurate reports. For our work, the AC is a fully connected network with same lower layers with two output layers one for each actor and critic. We used softmax activation function for actor layer to model probabilities.

IV. Simulations and Performance Results

The simulations are carried out on a Lambda GPU workstation AMD(R) Ryzen threadripper pro 3955wx 16 cores x32 with 128 GB RAM on a Ubuntu 20.04.5 LTS. In this section, we discuss the performance of DQN, AC and PPO based RL models in an environment where 50% of the vehicles are participating in a colluding attack as described in Fig 2. For visualization purposes, we have selected 10 vehicles where 2, 3, 5, 7, 9 numbered vehicles are malicious colluding vehicles and the other five are normal vehicles. We have plotted the values of average action values, reward values and final plausibility scores between 1-350 episodes.



(a) Average action values (DQN model) (b) Average reward values (DQN model) (c) Final malicious scores (DQN model)

(d) Average action values (AC model) (e) Average reward values (AC model) (f) Final malicious scores (AC model)

(g) Average action values (PPO model) (h) Average reward values (PPO model) (i) Final malicious scores (PPO model)

Fig. 3: Performance Analysis (Malicious colluding vehicles are numbered as 2, 3, 5, 7, 9)

1) DQN model: Figures 3(a) and 3(b) represents the average action and reward values respectively of a DQN based RL agent over time. Here, we can see that with average action values ranging from 0.45 - 0.85 and average reward values ranging from 0.4 - 1.0. This shows that the DQN agent was unable to find an optimal smoothing factor to mitigate the attacker thus affecting its reward values. Figure 3(c) represents the final malicious scores of vehicles in 300^{th} episode. High final malicious scores of vehicles represent that the colluding vehicles are successfully attacking the system even with DQN model in place.

2) AC model: Figures 3(d) and 3(e) represent the average action and reward values of a AC RL agent over time. Here we can see that AC agent was able to find a optimal smoothing factor of 0.94 to mitigate the colluding attack after approximately 120^{th} episode and was successful in forcing the colluding vehicles to send true reputation reports. The AC agent was able to achieve high reward value of 1.0 which can be seen in the reward value plot (See Figure 3(e)). Figure 3(f) represent the final malicious scores of vehicles with AC model in 300^{th} episode. The values ranging from 0-6 represents that colluding vehicles were never able to send false reputation reports because of the AC model based mitigating agent.

3) PPO model: Figures 3(g) and 3(h) represent the average action and reward values of a PPO agent. Here we can see that PPO agent achieved high reward values from 100^{th} episode by selecting average smoothing factor as 0.77 mitigating the colluding vehicles malicious behavior by forcing them to send true feedbacks. We can see that the reward values are 1 from which we can infer that the colluding vehicles are sending true reputation reports when PPO model was implemented at the CA. A final malicious score ranging from 0-6 in 3(i) represents the behavior of vehicles with PPO agent in place. As the malicious vehicles were forced to send true reputation reports right from the 100^{th} episode their respective malicious scores are low.

A. Comparative Analysis

We further describe the performance of different deep RL agents in the selection of smoothing factors in mitigating malicious colluding attacks across episodes. Table I contains the average action, reward and final malicious values for DQN, AC, and PPO RL agents between 100-150 episodes. From the data, it is evident that the PPO agent was able to achieve high rewards whereas AC model performed moderately by achieving 0.84. DQN failed to mitigate the malicious vehicles from sending the false reputation scores which can be observed from high final malicious scores and low reward values.

Model	Action	Reward	Average final malicious score
DQN	0.68	0.63	54.56
AC	0.84	0.84	4.19
PPO	0.77	1	4.18

TABLE I: Values between 100-150 episodes

Table II contains the average action, reward and final malicious values from DQN, AC, PPO RL agents between 200-250 episodes. From the data we can see that AC and PPO based agents were able to achieve same reward values and were successful in mitigating the malicious vehicles from sending false reputation scores. We can observe that AC agent chose 0.94 as optimal smoothing factor which was around 0.84 during 100-150 episodes to achieve high reward. From the low reward values and high final malicious scores we can infer that DQN based agent still suffers to mitigate malicious vehicles.

Model	Action	Reward	Average final malicious score
DQN	0.74	0.61	49.19
AC	0.94	1	4.02
PPO	0.77	1	3.72

TABLE II: Values between 200-250 episodes

Table III contains the values of average action, reward and final malicious score from DQN, AC and PPO RL agents between 500-750 episodes. It is evident from the table III that the AC and PPO continued to mitigate the colluding vehicles where as DQN agent was unable to find the optimal smoothing factor which mitigates the malicious vehicles.

Model	Action	Reward	Average final malicious score
DQN	0.72	0.61	46.8
AC	0.94	1	4.19
PPO	0.77	1	3.95

TABLE III: Values between 500-750 episodes

V. Conclusion

This paper proposed multi-level DS technique in conjunction with RL for detecting and mitigating colluding attacks in CV. From the simulation it was observed that the PPO-based RL agent showed its superiority by mitigating the influence of colluding malicious vehicles. Preventing them from sending false reputation scores right from the 100th episode, whereas the AC-based agent was able to find the optimal smoothing factor after the 120th episode. In contrast, the DQN agent struggled to find an optimal smoothing factor even after 750 episodes. From the simulation, we can conclude that the AC and PPO-based agents performed efficiently in mitigating malicious vehicles quickly, whereas the DQN-based model taking more time than these agents to find an optimal smoothing factor.

VI. Acknowledgment

This publication has been supported by NSF CISE Research Initiation Initiative (CRII) grant #2153510 and #2313351.

References

[1] M. Hijji et al. “6G connected vehicle framework to support intelligent road maintenance using deep learning data fusion”. *IEEE Transactions on Intelligent Transportation*, 2023.

[2] M. Amoozadeh et al., “Security vulnerabilities of connected vehicle streams and their impact on cooperative driving”. *IEEE Communications Magazine*, vol. 53, no. 6, pp.126-132, 2015.

[3] H. Rathore and H. Griffith, “Leveraging Neuro-Inspired Reinforcement Learning for Secure Reputation-based Communication in Connected Vehicles”, CPS-Sec Workshop, IEEE CNS, 2023.

[4] Y. Li et al., “TSP security in intelligent and connected vehicles: Challenges and solutions”. *IEEE Wireless Communications*, vol. 26, no. 3, pp.125-131, 2019.

[5] H. Griffith, M. Farooq and H. Rathore, “A Data Generation Workflow for Consensus-Based Connected Vehicle Security,” *IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, NV, USA, 2023, pp. 1-2, 2023.

[6] Qu, Ao, Yihong Tang, and Wei Ma. “Attacking deep reinforcement learning-based traffic signal control systems with colluding vehicles,” arXiv preprint arXiv:2111.02845 (2021).

[7] F. Kandah, B. Huber, A. Altarawneh, S. Medury and A. Skjellum, “BLAST: Blockchain-based Trust Management in Smart Cities and Connected Vehicles Setup,” *IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, pp. 1-7, 2019.

[8] M. A. Hoque and R. Hasan, “An Interaction Provenance-based Trust Management Scheme For Connected Vehicles,” in *IEEE 19th Annual Consumer Communications and Networking Conference (CCNC)*,, pp. 731-732, 2022.

[9] SS. Albouq and EM. Fredericks. “Detection and Avoidance of Wormhole Attacks in Connected Vehicles,” In *Proceedings of the 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (DIVANet ’17), Association for Computing Machinery*, pp. 107–116, 2017.

[10] S. Gyawali, Y. Qian and R. Q. Hu, “Deep Reinforcement Learning Based Dynamic Reputation Policy in 5G Based Vehicular Communication Networks,” in *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 6136-6146, 2021.

[11] Chen et al, “Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles,” in *Comput Aided Civ Inf.* vol 36, pp 838– 857, 2021.

[12] PJ Chowdary, H. Griffith, H. Rathore, “A Multi-Level Dempster-Shafer and Reinforcement Learning-Based Reputation System for Connected Vehicle Security”, *IEEE CCNC*, 2024.

[13] P.A. Lopez et al., “Microscopic traffic simulation using sumo”. In 2018 *21st international conference on intelligent transportation systems (ITSC)*, pp. 2575-2582, IEEE, 2018.

[14] Sentz, K. and Ferson, “Combination of evidence in Dempster-Shafer theory”. 2002

[15] Dong Chen et al, “Deep Multi-agent Reinforcement Learning for Highway On-Ramp Merging in Mixed Traffic,” 2022.

[16] https://github.com/DongChen06/MARL_CAVs [accessed on august 31, 2023].

[17] D. Suo and S. E. Sarma, “Proof-of-Travel: A Protocol for Trustworthy V2I Communication and Incentive Designs,” in *IEEE Vehicular Networking Conference (VNC)*, pp. 1-4, 2020.