MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM) | 979-8-3503-7423-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/MILCOM61039.2024.10773966

# Intrusion Response System for In-Vehicle Networks: Uncertainty-Aware Deep Reinforcement Learning-based Approach

Han Jun Yoon*, David Soon*, Terrence J. Moore†, Seunghyun Yoon‡, Hyuk Lim‡,
Dongseong Kim§, Frederica F. Nelson†, Jin-Hee Cho*
* Department of Computer Science, Virginia Tech, USA
†US DEVCOM Army Research Laboratory, USA
‡School of Energy Engineering, Korea Institute of Energy Technology (KENTECH), Republic of Korea
§School of Electrical Engineering and Computer Science, University of Queensland, Australia

***Abstract*—Modern vehicles use the Controller Area Network (CAN) bus system to manage communication between electronic control units (ECUs). The CAN bus lacks authentication and authorization mechanisms, and cryptographic protections are rarely used. This creates vulnerabilities to attacks such as Denial of Service, spoofing, and fuzzing, which can disrupt ECU functionality. Injection attacks through external connections (e.g., telematic unit, head unit, OBD-II port) can cause ECU malfunctions, leading to abnormal vehicle behavior or catastrophic events like car crashes. To address these issues, we propose a deep reinforcement learning (DRL)-based intrusion response system (IRS). Upon detecting an attack with an intrusion detection system (IDS), our IRS responds with the optimal defense strategy, maximizing defense utility. Our goal is to minimize the attack's success while ensuring mission completion within deadlines. We define an action space representing defense strategies for detected intrusions. Our extensive experiments prove that our IRS significantly outperforms state-of-the-art and baseline counterparts in minimizing the attack's success by up to 60% and maximizing the mission performance by up to 70%. This work is the first to propose a DRL-based IRS for managing multiple attacks in in-vehicle networks under ML-based IDS alerts.**

***Index Terms*—In-vehicle network, Intrusion detection system, Intrusion response system, Deep reinforcement learning**

## I. Introduction

The increasing integration of electronic control units (ECUs) in modern vehicles, managed through Controller Area Network (CAN) bus systems, has significantly enhanced vehicle functionality but has also introduced critical cybersecurity vulnerabilities [1]. Despite advancements in intrusion detection systems (IDSs) for in-vehicle networks, there remains a notable gap in the research and development of effective intrusion response systems (IRSs). This gap leaves vehicles vulnerable to sophisticated cyber threats, compromising safety and reliability. Military vehicles and unmanned systems, like their civilian counterparts, utilize CAN buses, which are susceptible to cyber-attacks such as Denial of Service (DoS), spoofing, and fuzzing [2]. A deep reinforcement learning (DRL)-based IRS offers an effective method for protecting these networks, playing a crucial role in ensuring the success and security of military missions in the face of cyber threats. **The aim of this work** is to address this critical need by developing an IRS that leverages deep reinforcement learning (DRL) to select the optimal defense against detected attacks autonomously. By integrating DRL, our system will dynamically and efficiently respond to cyber threats, ensuring robust protection and maintaining the integrity of vehicle operations. This innovative approach not only fills the existing void in in-vehicle security research but also sets a new standard for resilience against automotive cyber threats.

This work makes the following key contributions: First, our work pioneers a DRL-based Intrusion Response System (IRS) that effectively responds to multiple attack types using uncertainty-aware DRL with entropy regularization [3] to enhance vehicle security and mitigate cyber threats. Second, we introduce a sub-action space for the IRS, featuring discrete defensive actions tailored to each detected intrusion type, improving the efficiency of defense strategy selection. Finally, through extensive experiments, we demonstrate the superior efficacy of our DRL-based IRS over baseline defenses (random or none), reducing the attack success ratio (ASR) by up to 60% and improving the mission success ratio (MSR) by up to 70%.

## II. Related Work

### A. Cybersecurity in In-Vehicle Networks

Han et al. [4] introduced ID-Anonymization for CAN (IA-CAN), a novel protocol mitigating Denial-of-Service (DoS) attacks and securing in-vehicle and external communication. Wu et al. [5] found that the CANoe framework and Genuino UNO boards, combined with machine learning (ML) algorithms, can accurately detect irregular activities within in-vehicle network systems. Kim and Shrestha [6] proposed cybersecurity layers including network access control, real-time anomaly detection, and encryption protocols. El-Rewini et al. [7] suggested System of Systems (SoS) strategies using cryptographic methods and authentication protocols to safeguard essential components.

One of the key tools for in-vehicle network cybersecurity is an IDS. Song et al. [8] developed a lightweight IDS that improved significantly over previous rate-based methods. Seo et al. [9] created GIDS (GAN-based IDS) using Generative Ad-

versarial Nets to detect both known and unknown attacks. Kang and Kang [10] developed a deep neural network (DNN)-based IDS enhancing detection capabilities while maintaining efficient real-time response. Despite various cybersecurity measures for in-vehicle networks, there is a notable lack of research on IRS.

### B. Intrusion Response Systems (IRSs)

Cheng et al. [11] used a zero-sum stochastic game and a Bayesian attack graph to simulate network intrusions, incorporating various levels of Theory of Mind in attacker and defender strategies. Ullah et al. [12] designed an IRS for targeted attacks that balance security and operational efficiency by considering attack likelihood and functional dependencies, thereby extending attack durations to deter threats. Nespoli et al. [13] introduced an immune-inspired IRS employing a Genetic Algorithm to optimize countermeasures, similar to antibodies, to efficiently reduce security risks. DRL has been crucial for developing model-free IRSs. Hughes et al. [14] applied deep Q-network learning to create an automated IRS with 21 actions, achieving impressive results with optimized hyperparameters. Iannucci et al. [15] developed a model-free IRS using Q-Learning and DQN, with system designs based on node characteristics and a reward function to minimize costs and response times. However, these IRS technologies [11–15] have not been tailored for the cybersecurity of autonomous vehicles.

For in-vehicle network IRS, Hamad et al. [16] explored the overall structure of IRS in in-vehicle networks, noting that their approach did not directly activate response systems following IDS alerts. Kwon et al. [17] designed a solution to mitigate network intrusions in vehicles by reconfiguring the ECU and neutralizing malicious packets. However, IRS research for in-vehicle security is extremely rare. We address this gap by proposing a DRL-based IRS for in-vehicle cybersecurity.

## III. Problem Statement

We consider an in-vehicle network consisting of a CAN bus designed to assist the ECU in communicating with the outside world. Given a detected intrusion (i.e., a specific attack vector), the system must respond properly to counteract the intrusion. The attacker's aim is to inject random or malicious messages to disrupt the system's mission execution, where the mission is to reach a particular destination within a certain time constraint. Fig. 1 describes the network model considered in this work.

The given in-vehicle network aims to maximize the effectiveness of the IRS in terms of minimizing an attack success ratio (ASR) while completing the trip within the given deadline under adversarial attacks considered in this work (see our Attack Model in Section IV-C). To formally put it, we aim to:

$$\arg\min_{\mathbf{d}} = d_c + \frac{\sum_{t=0,\, d_t \in \mathbf{d}}^{T_c} \mathrm{AS}_t(d_t)}{N_A}, \quad \text{subject to} \quad T_c \leq T$$

where $\mathrm{AS}_t(d_t)$ refers to attack success (or failure), returning 1 or 0, respectively, when $d_t$ is a chosen defense response at round $t$, $T_c$ is the time taken to complete the mission, $T$ is a mission deadline, $d_c$ is the defense cost, and $N_A$ is the total number of attacks performed during the period of $T_c$. Section IV-D describes the set of defense responses. The $\mathrm{AS}_t(d_t)$ returns 1 if an attack is successful; 0 otherwise. If $T_c > T$, the mission is considered failed, and we set the total mission time, $T_c = T$).

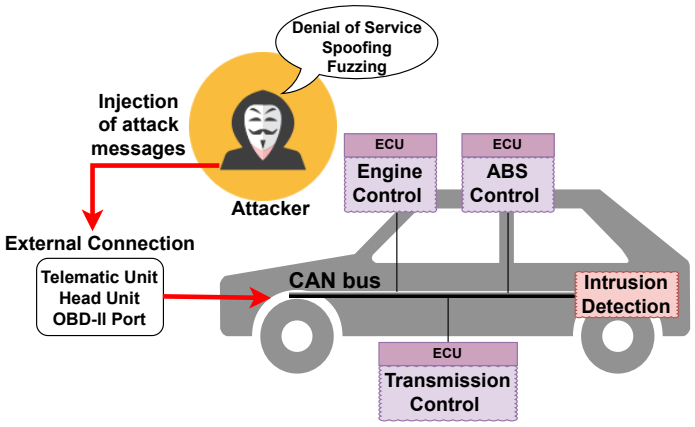



Fig. 1. The considered in-vehicle network.

## IV. System Model

### A. Network Model

We consider an in-vehicle network where the CAN protocol facilitates communication between ECUs and the external environment. Each ECU manages specific car functions, such as drive gear and Revolutions-Per-Minute (RPM) gauge. In-vehicle networks lack robust authentication and authorization due to their original design focus on performance, cost, and real-time communication over security, making them vulnerable to cyber threats like message spoofing and denial of service [18]. Resource constraints, such as limited computational power and memory, further complicate implementing cryptographic solutions [19]. Therefore, protocols like CAN have vulnerabilities that attackers can exploit. When the IDS detects an intrusion, the system aims to implement the most appropriate response to counter the attacker and protect the in-vehicle network, ensuring the vehicle can reach its destination safely and timely.

### B. Node Model

This work considers the following types of nodes:

- The *CAN bus network* is a message-based protocol enabling reliable, priority-driven communication among vehicle ECUs.
- ECUs are compact devices managing specific vehicle functions. Our study focuses on the engine control, transmission control, and ABS modules.
- The IDS oversees CAN messages and identifies attacks. We utilize a machine learning-based IDS to notify the system of the specific type of attack.
- The *Telematic Unit (TU)* is a communication device for two-way data exchange between a vehicle and the external environment via wireless modules. It enhances vehicle functions and comfort, such as navigation, and supports safe driving.
- The *Head Unit (HU)* provides a unified hardware interface for the system, including screens, buttons, and system controls for various integrated information and entertainment functions.

TABLE I
ATTACKER STRATEGIES, AND ATTACK IMPACT

| AS | Attack strategies | Attack impact |
|---|---|---|
| $AS_1$ | Denial of Service | The ABS Control ECU will be flooded with a large number of messages, leading to sudden brake or brake malfunction. |
| $AS_2$ | Spoofing | Change RPM randomly or switch drive gear for a given time-step depending on which spoof message was injected (i.e., spoofing messages are either related to RPM or drive gear) |
| $AS_3$ | Fuzzing | Successful injection of fuzzing attack will result in the stop of the given vehicle. |

### C. Attack Model

The attackers inject attack messages through external connections such as the telematic unit, head unit, and OBD-II port. We summarize attacker strategies and their impact in Table I. This work considers the following attack behaviors:

- *Denial of Service (DoS) ($AS_1$)*: This attack aims to consume CAN bus bandwidth by sending a massive amount of messages, allowing an ECU node to dominate the CAN bus resources. We model the DoS attack by injecting large messages with the CAN ID set to 0×000 into the vehicle networks. CAN-ID 0x000 has the highest priority on the CAN bus, allowing it to dominate communication and block lower-priority messages. This overloads the network, preventing the ABS ECU from receiving timely messages and potentially causing braking control failure.
- *Spoofing ($AS_2$)*: CAN messages are injected to control specific functions. Spoofing messages target the engine control ECU with RPM-related CAN IDs and the transmission control ECU with gear-related CAN IDs. Successful spoofing can increase or lower RPM and switch the drive gear to neutral or park.
- *Fuzzing ($AS_3$)*: This attack sends random CAN IDs and data, which can lead to a sudden vehicle stop if critical ECUs such as the engine, brakes, or transmission are targeted.

These attacks can disrupt the vehicle's mission of reaching its destination within a limited timeline by causing malfunctions related to RPM, drive gear, and the braking system. To assess the impact of attack severity, we use the probability $P_A$, which models the frequency of attacks launched by an attacker.

### D. Defense Model

We consider the following defense strategies to counteract the attacks in the Attack Model (Section IV-C):

- *Rate Limiting ($DS_1$)* limits the rate of messages transmitted or received by ECUs to prevent damage from flooding attacks.
- *Software Update ($DS_2$)* releases new software versions for ECUs to patch known vulnerabilities or update software. The TU manages external communication for updates, while the CAN bus distributes them internally to the ECUs.
- *Access Control List (ACL) ($DS_3$)* specifies which entities (e.g., ECUs, sensors, devices) are granted or denied access to network resources based on identity, role, or other attributes.
- *Network Filtering ($DS_4$)* limits traffic from suspicious sources by controlling access to network resources.
- *Input Validation ($DS_5$)* implements robust input validation and error-handling mechanisms to manage malformed or unexpected inputs, reducing the impact of fuzzing attacks.

Defense cost is assigned as low, medium, or high based on the presumed complexity and resource demands of each strategy. Table II summarizes defender strategies, success conditions, and associated implementation costs.

## V. UNCERTAINTY-AWARE DRL-BASED IRS

We consider the defense strategies listed in Table III. Upon detecting an attack, we identify a subset of effective defenses. For efficiency, we introduce the concept of a subgame in Game Theory [20] to narrow down the defense strategies for each detected attack. For example, if $AS_1$ is detected, the defender will consider $DS_1$, $DS_3$, $DS_4$, and $DS_6$ rather than all six strategies. This approach reduces the cost of calibrating the probability distribution for the defense strategies. $AS_4$ (no attack) represents the full action space, enabling the system to consider all defense options without the limitations imposed by a detected threat. This unrestricted approach allows for comprehensive evaluation and deployment of defense strategies.

### A. Entropy Regularization

To incorporate uncertainty into DRL, we use *entropy regularization* [3]. This technique encourages more exploratory policies by adding a penalty based on the entropy of the policy distribution. Entropy measures the unpredictability of the agent's actions in a given state. The entropy of the action probability distribution is calculated as:

$$\mathrm{H}(\pi_\theta(a_1, a_2, \ldots, a_n|s_t)) = -\sum_i^n \pi_\theta(a_i|s_t)\log(\pi_\theta(a_i|s_t)), \tag{1}$$

where $H(\pi_\theta(a_1, a_2, \ldots, a_n|s_t))$ is the entropy at state $s_t$, $a_i$ represents an action, and $\pi_\theta(a_i|s_t)$ is the probability of taking action $a_i$ given state $s_t$ under the policy parameterized by $\theta$. This entropy term is added to the objective function of a policy-based DRL algorithm (e.g., PPO) with a coefficient $\beta$ to control the regularization strength.

Without entropy regularization, a DRL agent may quickly converge to a deterministic policy, limiting its exploration. By encouraging a more exploratory and stochastic policy, entropy regularization introduces uncertainty into the learning process. This penalizes overly deterministic policies and promotes thorough exploration of the state-action space.

### B. DRL-based Response Selection

We formulate the proposed optimization problem for the defender to select the best defense response, $\mathbf{d}_t$, maximizing its net effectiveness using DRL. The problem is based on a *Markov Decision Process* (MDP) with the following components:

- **States**: The set of states is defined as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_t, \ldots, \mathbf{s}_T\}$, where $\mathbf{s}_t$ represents the state at time $t$. At time $t$, $\mathbf{s}_t$ is given by:

$$s_t = (AT_t), \tag{2}$$

TABLE II
DEFENDER STRATEGIES, CONDITIONS FOR SUCCESSFUL DEFENSE, AND DEFENSE COST
(DEFENSE COST: LOW – 1, MEDIUM –2, HIGH – 3)

| DS | Defense strategies | Successful defense condition | Defense cost |
|---|---|---|---|
| $DS_1$ | Rate Limiting | Applied on a targeted ECU node with a success probability of $P_{rl}$ | Low |
| $DS_2$ | Software Update | Applied on a targeted ECU node with success probability of $P_{su}$ | Low |
| $DS_3$ | Access Control List | $P_{acl}$ % success probability of blacklisting the attacker node | Medium |
| $DS_4$ | Network Filtering | $P_{nf}$ % success probability of blacklisting the attacker node | High |
| $DS_5$ | Input Validation | Applied on a targeted ECU node with success probability of $P_{iv}$ | High |
| $DS_6$ | No Defense | N/A | N/A |

TABLE III
CONSIDERED DEFENSE ACTION SPACE UNDER EACH INTRUSION

| AS | Action Space |
|---|---|
| $AS_1$ | $DS_1$, $DS_3$ $DS_4$, $DS_6$ |
| $AS_2$ | $DS_3$, $DS_4$, $DS_6$ |
| $AS_3$ | $DS_2$, $DS_5$, $DS_6$ |
| $AS_4$ | $DS_1$, $DS_2$, $DS_3$, $DS_4$, $DS_5$, $DS_6$ |

where $AT_t$ is the detected intrusion type at time $t$.

- **Actions**: The set of actions is defined as $\mathbf{A} = \{\mathbf{a}_1, \ldots, \mathbf{a}_t, \ldots, \mathbf{a}_T\}$, where $\mathbf{a}_t$ represents the actions available at time $t$. At time $t$, $\mathbf{a}_t$ is:

$$\mathbf{a}_t = (\mathbf{d}_t^*), \tag{3}$$

where $\mathbf{d}_t^*$ is the best defense response to counteract the attack and minimize ASR.

- **Rewards**: The reward function at time step $t$, denoted as $\mathbf{R}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$, for an action taken is calculated with two key objectives in mind: (1) to maximize the reduction in the ASR between two consecutive time points, $t$ and $t+1$, which assesses the effectiveness of the selected defense action against an attack; and (2) to maximize defense efficiency by minimizing the cost associated with the defense action. The formulation of the reward at time $t$ is:

$$\mathbf{R}(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1}) \tag{4}$$

$$= \alpha \times \underbrace{\left( \sum_{i=0,\ d_i \in \mathbf{d}}^{i=t} \mathrm{AS}_i(d_i) - \sum_{i=0,\ d_i \in \mathbf{d}}^{i=t+1} \mathrm{AS}_i(d_i) \right)}_{\text{difference in the number of successful attacks between two consecutive states}}$$

$$+\beta \times \underbrace{\frac{1}{1+d_c^t}}_{\text{defense efficiency}} . \tag{5}$$

The first term related to attack success (AS) is negative when the number of successful attacks increases from $t$ to $t+1$, and zero otherwise. The second term, associated with defense costs, decreases as defense costs rise and increases when they fall. We employ weights, $\alpha$ and $\beta$, to balance these objectives and prevent one from overshadowing the other with $\alpha+\beta = 1$. Here $\mathrm{AS}_i(d_i)$ returns 1 when a launched attack is successful at time $i$ when a defense action $d_i$ is taken at time $i$.

- **Transition Probabilities**: The transition probability $T(\mathbf{s}_t, \mathbf{a}_t, \mathbf{s}_{t+1})$ represents the likelihood of moving from state $\mathbf{s}_t$ to state $\mathbf{s}_{t+1}$ via action $\mathbf{a}_t$.
- **Reward Accumulation**: The accumulated reward is given by:

$$\mathbf{G}(t) = \sum_{k=0}^{T_c} \gamma^k R(t+k), \tag{6}$$

where $\gamma$ is the discount factor in the range $[0, 1]$, with lower values favoring immediate rewards.

The policy function, $\pi : s \rightarrow a$, maps states to a probability distribution of actions. Given an MDP episode of length $T_c$, the sequence of states, actions, and rewards forms the policy's trajectory. The goal of RL is to identify the optimal policy that maximizes the expected reward.

## VI. EXPERIMENTAL SETUP

**Datasets.** We use open-source car-hacking datasets [21], which include DoS attacks, fuzzy attacks, drive gear spoofing, and RPM gauge spoofing. These datasets were created by logging CAN traffic from a real vehicle via the OBD-II port during message injection attacks. We use these datasets to train our IDS to detect intrusion types.

**ML-based IDS Setup.** We use an existing ML classifier to build predictive models for IDS implementation. Using open-source car-hacking datasets [21], we develop a Random Forest-based IDS with nearly 97% accuracy.

**Metrics.** Our experiments use the following metrics:

- *Attack Success Ratio ($\mathcal{ASR}$)* measures the ratio of successful attacks to the total number of attacks launched.
- *Attack Success Impact ($\mathcal{ASI}$)* measures the change in throttle, brake, and gear value due to successful attacks.
- *Mission Success Ratio ($\mathcal{MSR}$)* refers to the ratio of successful missions to the total number of missions attempted.
- *Defense Cost ($\mathcal{DC}$)* indicates the total defense cost incurred during mission execution.
- *Route Completion ($\mathcal{RC}$)* refers to the percentage of the route distance completed by the vehicle.
- *Infraction Score ($\mathcal{IS}$)* sums all infractions as a geometric series, with each rule violation or unsafe behavior contributing less to the total score.
- *Driving Score ($\mathcal{DS}$)* is the product of route completion and the infraction penalty.

**Comparing schemes.** To select a defense strategy, we use the following algorithms for extensive experimental validation:

- **Proximal Policy Optimization (PPO)** [22] is an RL algorithm that enhances training stability and reliability by using a clipped objective function to prevent large policy updates, ensuring controlled and effective learning.
- **Deep Q Learning (DQN)** [23] utilizes neural networks parameterized by $\theta$ to represent the action-value function, assuming the agent observing the environment fully.

- **Sub-action-based PPO (S-PPO)** uses PPO with a sub-action space, where a subset of the full action space is employed based on the detected attack type.
- **Sub-action-based DQN (S-DQN)** applies DQN with a sub-action space upon detecting an attack.
- **Random** is a method that selects a defense strategy at random from all available strategies.
- **No Defense** is a baseline approach with no defense strategy to counter the detected attacks.

Table V summarizes the design parameters and their meanings.

## VII. Numerical Results & Analyses

**Analysis of Security and Mission Performance**: Fig. 2 shows experimental results on how attack severity ($P_A$) affects key metrics: attack success ratio ($\mathcal{ASR}$), mission success ratio ($\mathcal{MSR}$), defense cost ($\mathcal{DC}$), and attack success impact ($\mathcal{ASI}$). Fig. 2(a) demonstrates that `S-PPO` and `S-DQN` consistently outperform `PPO` and `DQN`, maintaining stable $\mathcal{ASR}$ despite $P_A$ variations, due to the DRL agent's dynamic learning and strategy adjustment. In Fig. 2(b), `S-PPO` and `S-DQN` use a sub-action space to effectively enhance defense actions, unlike the full action space, which limits efficient exploration and leads to poorer defense decisions. Fig. 2(c) shows optimal defense resource allocation by the agent, resulting in stable $\mathcal{DC}$. Fig. 2(d) confirms that sub-action schemes consistently surpass conventional methods.

**Analysis of Performance of Autonomous Vehicle**: Fig. 3 explores the impact of varying attack severities ($P_A$, probability of launching an attack) on different schemes in terms of route completion ($\mathcal{RC}$), Infraction Score ($\mathcal{IS}$), and driving score ($\mathcal{DS}$). As observed in Fig. 3(a), the strategies `S-PPO` and `S-DQN` demonstrate superior performance in route completion ($\mathcal{RC}$) compared to `PPO` and `DQN`. This advantage is attributed to the reduced complexity of the action space, which enables our proposed DRL-based schemes to more effectively complete mission routes than other baseline approaches.

Further analyses in Figs. 3(b) and 3(c) reveal that `S-PPO` and `S-DQN` also excel in reducing the Infraction Score ($\mathcal{IS}$) and enhancing the driving score ($\mathcal{DS}$), outperforming `PPO` and `DQN`. The performance of these models is followed sequentially by `Random`, and `No Defense`, showcasing the effectiveness of the sophisticated control strategies employed by `S-PPO` and `S-DQN` in maintaining safe and efficient driving behaviors under varying attack conditions.

**Empirical Training Time Analysis**: Table IV demonstrates that using a sub-action space significantly reduces training time compared to a full action space for both PPO and DQN algorithms. The sub-action space enhances efficiency and speeds up convergence by reducing computational complexity, thus providing timely results. We observe that DQN requires more training time than PPO. This extended duration is attributed to DQN's use of a large replay buffer, increasing the time spent sampling and utilizing experiences for training. While the larger buffer improves learning stability, it slows training. The analysis was performed on a system with a 1.4 GHz Quad-Core Intel Core i5 CPU, 8 GB of RAM, and an Intel Iris Plus Graphics GPU with 1536 MB of memory.

TABLE IV
Training Time in Seconds

| DRL-based IRS Schemes | Training time in seconds |
|---|---|
| `PPO` | 1197 |
| `S-PPO` | 778 |
| `DQN` | 23172 |
| `S-DQN` | 22624 |

TABLE V
Design parameters, their meaning, and default values

| Par. | Meaning | Value |
|---|---|---|
| $\alpha, \beta$ | Weight for reward function | 0.67/0.33 |
| $\gamma$ | Discount rate | 0.9 |
| $N$ | PPO/DQN network Size | 256 |
| $l_{\text{actor}}$ | Learning rate for actor-network | 0.00005 |
| $l_{\text{critic}}$ | Learning rate for critic network | 0.0005 |
| $l$ | DQN Learning rate | 0.0001 |
| $\epsilon$ | Exploration rate | 1.0 |
| $\epsilon_{\text{min}}$ | Minimum exploration rate | 0.1 |
| $\epsilon_{\text{decay}}$ | Epsilon decay | 0.9 |

## VIII. Conclusion & Future Work

Modern vehicles use the CAN bus system for communication between ECUs, but it lacks security features, making it vulnerable to attacks like DoS, spoofing, and fuzzing. These attacks can disrupt ECU functionality, leading to serious issues such as vehicle malfunctions or crashes. In-vehicle networks typically lack authentication and authorization, as they were originally designed with the assumption that all devices were trustworthy, prioritizing functionality and efficiency over security.

To address these vulnerabilities, we proposed a DRL-based IRS that responds optimally to detected attacks, minimizing the attack's success and ensuring mission completion within deadlines. The proposed DRL-based approaches outperform baseline methods, reducing the ASR by up to 60%, improving MSR by up to 70%, and optimizing other security and performance metrics of the autonomous vehicle. Leveraging a sub-action space-based design introduced efficiency for tailored defense strategies. Additionally, employing uncertainty-aware DRL with entropy regularization has improved solution quality by fostering greater diversity in solutions.

**Future Work.** We plan to conduct the following future work: (1) Consider fault-tolerant networks with less effective IDS. (2) Incorporate human input to develop human-in-the-loop reinforcement learning. (3) Develop bundle-based defense strategies to handle multiple simultaneous attacks.

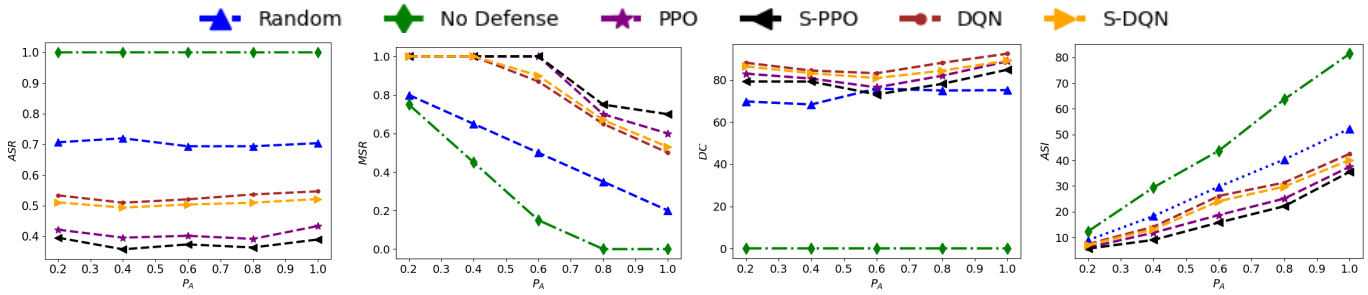
## Acknowledgment

This research was partly supported by Army Research Office grant #W911NF-20-2-0140 and NSF grant #2107450.

## References

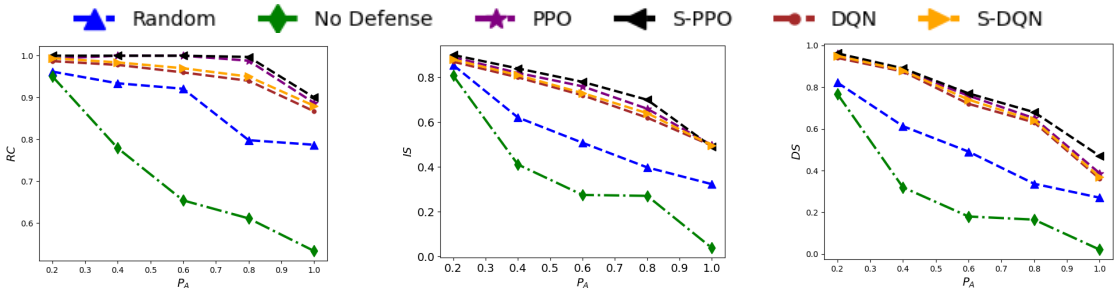
[1] R. S. Rathore, C. Hewage, O. Kaiwartya, and J. Lloret, "In-vehicle communication cyber security: Challenges and solutions," *Sensors*, vol. 22, no. 17, p. 6679, Sep 2022. [Online]. Available: http://dx.doi.org/10.3390/s22176679

[2] ——, "In-vehicle communication cyber security: challenges and solutions," *Sensors*, vol. 22, no. 17, p. 6679, 2022.




(a) $P_A$ vs. $\mathcal{ASR}$ (b) $P_A$ vs. $\mathcal{MSR}$ (c) $P_A$ vs. $\mathcal{DC}$ (d) $P_A$ vs. $\mathcal{ASI}$

Fig. 2. Effect of different attack severity ($P_A$, probability of launching an attack) to varying schemes in terms of attack success ratio ($\mathcal{ASR}$), mission success ratio ($\mathcal{MSR}$), defense cost($\mathcal{DC}$), and attack success impact ($\mathcal{ASI}$).




(a) $P_A$ vs. $\mathcal{RC}$ (b) $P_A$ vs. $\mathcal{IS}$ (c) $P_A$ vs. $\mathcal{DS}$

Fig. 3. Effect of different attack severity ($P_A$, probability of launching an attack) to varying schemes in terms of route completion ($\mathcal{RC}$), Infraction Score ($\mathcal{IS}$), and driving score ($\mathcal{DS}$).


[3] S. Zhao, M. Gong, T. Liu, H. Fu, and D. Tao, "Domain generalization via entropy regularization," *Neurips*, vol. 33, pp. 16 096–16 107, 2020.

[4] K. Han, A. Weimerskirch, and K. G. Shin, "Automotive cybersecurity for in-vehicle communication," *IQT QUARTERLY*, vol. 6, no. 1, pp. 22–25, 2014.

[5] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 3, pp. 919–933, 2020.

[6] S. Kim and R. Shrestha, *In-Vehicle Communication and Cyber Security*. Springer Singapore, 2020, pp. 67–96.

[7] Z. El-Rewini, K. Sadatsharan, D. F. Selvaraj, S. J. Plathottam, and P. Ranganathan, "Cybersecurity challenges in vehicular communications," *Vehicular Communications*, vol. 23, p. 100214, 2020.

[8] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 Int'l Conf. Information Networking (ICOIN)*, 2016, pp. 63–68.

[9] E. Seo, H. M. Song, and H. K. Kim, "GIDS: Gan based intrusion detection system for in-vehicle network," in *2018 16th Annual Conf. Privacy, Security and Trust (PST)*, 2018, pp. 1–6.

[10] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PLOS ONE*, vol. 11, no. 6, pp. 1–17, Jun. 2016.

[11] Q. Cheng, C. Wu, B. Hu, D. Kong, and B. Zhou, "Think that attackers think: Using first-order theory of mind in intrusion response system," in *2019 IEEE Global Communications Conf. (GLOBECOM)*, 2019, pp. 1–6.

[12] S. Ullah, S. Shelly, A. Hassanzadeh, A. Nayak, and K. Hasan, "On the effectiveness of intrusion response systems against persistent threats," in *2020 Int'l Conf. on Computing, Networking and Communications (ICNC)*. IEEE, 2020, pp. 415–421.

[13] P. Nespoli, F. Gomez Marmol, and G. Kambourakis, "AISGA: Multi-objective parameters optimization for countermeasures selection through genetic algorithm," in *The 16th Int'l Conf. Availability, Reliability and Security*, 2021, pp. 1–8.

[14] K. Hughes, K. McLaughlin, and S. Sezer, "A model-free approach to intrusion response systems," *J. Information Security and Applications*, vol. 66, p. 103150, 2022.

[15] S. Iannucci, V. Cardellini, O. D. Barba, and I. Banicescu, "A hybrid model-free approach for the near-optimal intrusion response control of non-stationary systems," *Future Generation Comp. Sys.*, vol. 109, pp. 111–124, 2020.

[16] M. Hamad, M. Tsantekidis, and V. Prevelakis, "Intrusion response system for vehicles: Challenges and vision," in *Smart Cities, Green Technologies and Intelligent Transport Systems*. Springer, 2019, pp. 321–341.

[17] H. Kwon, S. Lee, J. Choi, and B.-h. Chung, "Mitigation mechanism against in-vehicle network intrusion by reconfiguring ecu and disabling attack packet," in *2018 Int'l Conf. Information Technology (InCIT)*, 2018, pp. 1–5.

[18] M. De Vincenzi, G. Costantino, I. Matteucci, F. Fenzl, C. Plappert, R. Rieke, and D. Zelle, "A systematic review on security attacks and countermeasures in automotive ethernet," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, 2024.

[19] Z. Petho, I. Khan, and Á. Torok, "Analysis of security vulnerability levels of in-vehicle network topologies applying graph representations," *Journal of Electronic Testing*, pp. 1–9, 2021.

[20] S. Tadelis, *Game Theory: an Introduction*. Princeton University Press, 2013.

[21] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Comms.*, vol. 21, p. 100198, 2020.

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

[23] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A theoretical analysis of deep Q-learning," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 486–489.