2018 21st International Conference on Intelligent Transportation Systems (ITSC)
Maui, Hawaii, USA, November 4-7, 2018

Robust Deep Reinforcement Learning for Security and Safety in Autonomous Vehicle Systems

Aidin Ferdowsi[1], Ursula Challita[2], Walid Saad[1], and Narayan B. Mandayam[3]

***Abstract*— The dependence of autonomous vehicles (AVs) on sensors and communication links exposes them to cyber-physical (CP) attacks by adversaries that seek to take control of the AVs by manipulating their data. In this paper, the state estimation process for monitoring AV dynamics, in presence of CP attacks, is analyzed and a novel adversarial deep reinforcement learning (RL) algorithm is proposed to maximize the robustness of AV dynamics control to CP attacks. The attacker's action and the AV's reaction to CP attacks are studied in a game-theoretic framework. In the formulated game, the attacker seeks to inject faulty data to AV sensor readings so as to manipulate the inter-vehicle optimal safe spacing and potentially increase the risk of AV accidents or reduce the vehicle flow on the roads. Meanwhile, the AV, acting as a defender, seeks to minimize the deviations of spacing so as to ensure robustness to the attacker's actions. Since the AV has no information about the attacker's action and due to the infinite possibilities for data value manipulations, each player uses long short term memory (LSTM) blocks to learn the expected spacing deviation resulting from its own action and feeds this deviation to a reinforcement learning (RL) algorithm. Then, the attacker's RL algorithm chooses the action which maximizes the spacing deviation, while the AV's RL algorithm seeks to find the optimal action that minimizes such deviation. Simulation results show that the proposed adversarial deep RL algorithm can improve the robustness of the AV dynamics control as it minimizes the intra-AV spacing deviation.**

I. Introduction

Intelligent transportation systems (ITS) will encompass autonomous vehicles (AVs), roadside smart sensors (RSSs), and even drones [1]–[3]. To operate autonomously, AVs must be able to process a large volume of ITS data collected via a plethora of sensors and communication links. However, this reliance on communications and data processing renders AVs highly susceptible to cyber-physical attacks. In particular, an attacker can possibly interject the AV data processing stage, reduce the reliability of measurements by injecting faulty data, and ultimately induce accidents or compromise the traffic flow in the ITS [4]. Such flow disruptions can also cascade to other interdependent critical infrastructure such as power grids or cellular communication systems that provide service to the ITSs [5], [6].

Recently, a number of security solutions have been proposed for addressing intra-vehicle security problems [7]–[10]. In [7], the authors proposed a number of intrusion detection algorithms to secure a vehicle's controller. Further, in [8], the authors demonstrate the impact of long-range wireless attacks on the security of an AV's protocols and controller. Meanwhile, the authors in [9] addressed the security challenges of plug-in electric vehicles, while accounting for their impact on the power system. Moreover, a survey on security threats and protection mechanisms in embedded automotive networks is presented in [10]. The security of vehicular communications has been studied recently in [11]–[14]. The authors in [11] proposed the use of multi-source filters to reduce the vulnerability of a vehicular network, with respect to data injection attacks. The work in [12] introduced a new framework to improve the trustworthiness of beacons by combining two physical measurements (angle of arrival and Doppler effect) from received wireless signals. Moreover, in [13], the authors proposed a collaborative control strategy for vehicular platooning to address spoofing and denial of service attacks. Finally, an overview of current research on advanced intra-vehicle networks and the smart components of ITS and their applications is presented in [14].

However, the solutions in [7]–[15] do not consider the interdependence between the cyber and physical layers of AVs. Moreover, these existing works do not properly model the attacker's actions and goals. In this context, accounting for the cyber-physical interdependence of the attacker's actions and goals will help providing better security solutions. Furthermore, the solutions in [7]–[15], cannot enhance the robustness of AV control to attacks. Nevertheless, designing an optimal safe ITS requires robustness to attacks on intra-vehicle sensors as well as inter-vehicle communication. Moreover, existing works on ITS security often assume a constant attack model, while in many practical scenarios, the attacker might adaptively change its strategy to increase the impact of its attack on the ITS.

The main contribution of this paper is, thus, to propose a novel adversarial deep reinforcement learning (RL) framework that aims at providing robust AV control. We consider a car following model in which we focus on the control of an AV that closely follows another AV. Such a model captures the AV's dynamics control while taking into account AV's sensor readings and beaconing. We consider four sources of information about the leading AV gathered from intra-vehicle sensors such as camera, radar, RSSs, and inter-vehicle beaconing. We consider an attacker which can inject bad data to such information to compromise the system. In contrast, the AV's goal is to optimally control its speed while staying robust to such data injection attacks. To analyze the

This research was supported by the U.S. National Science Foundation under Grants OAC-1541105, IIS-1633363, and OAC-1541069.

[1]Aidin Ferdowsi and Walid Saad are with Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, {aidin, walids}@vt.edu

[2]Ursula Challita is with Ericsson Research, Stockholm, Sweden, ursula.challita@ericsson.com (This work was done when this author was at The University of Edinburgh)

[3]Narayan B. Mandayam is with WINLAB, Dept. of ECE, Rutgers University, New Brunswick, NJ, USA, narayan@winlab.rutgers.edu

interactions between the AV and the attacker, we pose the problem as a noncooperative game and analyze its Nash equilibrium (NE). However, we observe that obtaining the NE will be challenging due to the continuity of the action sets and the AV's speed and spacing. To address this problem, we propose two deep neural networks (DNNs) based on long-short term memory (LSTM) blocks for the AV and the attacker that extract the past AV dynamics and feed them to an RL algorithm for each player. On the one hand, the AV's RL algorithm tries to learn the best estimation from its leading AV's speed by combining the sensor readings. On the other hand, the RL algorithm for the attacker tries to deceive the AV and deviate the inter-vehicle optimal safe spacing. Simulation results show that the proposed deep RL algorithm converges to a mixed-strategy NE and can lead to significant improvement in the AV's robustness to data injection attacks. The results also show that the AV can use the proposed deep RL algorithm to reduce the deviations from the optimal safe spacing.

The rest of the paper is organized as follows. Section II introduces the system model for AV control. Section III formulates the proposed game. Section IV proposes our adversarial deep learning algorithm. Section V analyzes the simulations and conclusions are drawn in Section VI.

II. System Model

Consider a smart road in an ITS consisting of multiple AVs and RSSs. Each AV i is equipped with a camera to take images from the environment, a radar to measure distances from objects in the vicinity of the AV, and a transceiver device to communicate important position-speed-acceleration (PVA) beacons with nearby AVs and sensors over a cellular network. One challenging area in such ITSs is the optimal and safe flow control of AVs by using the collected measurements and received beacons. Moreover, the presence of an adversary might induce faulty decisions to the ITS and result in accidents or reduce the vehicle flow. Thus, the AVs' control on the roads must be robust to faulty data injected to the measurements and beacons by a malicious attacker. Next, we present an estimation model at each AV to observe the speed of its leading AV using sensor and beacon fusion and we model the adversary. Then, we define a dynamic process to capture the spacing between the AVs as a function of the attacker and AV actions.

A. Autonomous Vehicle Cyber-physical System

In order to drive safely and prevent accidents, each AV i must acquire information about its own position, speed v_i as well as the distance and speed of some nearby objects such as the immediately leading AV, $i-1$. One framework to analyze the speed of an AV i is by using the so called *car-following* models that is popular in the literature [16]. Here, we use the General Motors' first car-following model to analyze the speed update at each AV i as a function of AV $i-1$'s speed as follows [16]:

$$\dot{v}_i(t) = \lambda(\hat{v}_{i-1}(t) - v_i(t)), \tag{1}$$

where λ is a reaction parameter and $\hat{v}_{i-1}(t)$ is the estimated speed of AV $i-1$ at AV i. As we can see from (1), each vehicle must estimate $\hat{v}_{i-1}(t)$ at each time step in order to control its dynamics. To this end, each AV must use its own built-in sensors such as camera, radar as well as periodic reports from AV $i-1$ and the closest RSS. Thus, at each AV i, AV $i-1$'s speed, $v_{i-1}(t)$, must be estimated from AV $i-1$'s measured speed using a camera image, c_i, and radar reading, r_i, on AV i, AV $i-1$'s speed report u_{i-1}, and closest RSS's reported speed s_i. Therefore, the relationship between AV $i-1$'s exact speed and the measurements can be expressed using a *generic linear model* as follows:

$$\boldsymbol{z}_i(t) = \boldsymbol{H}_i v_{i-1}(t) + \boldsymbol{e}_i(t), \tag{2}$$

where $\boldsymbol{H} \in \mathbb{R}^{4\times 1}$ is the measurement Jacobian matrix, $\boldsymbol{z}_i \triangleq [c_i, r_i, u_{i-1}, s_i]^T$, and $\boldsymbol{e}_i \in \mathbb{R}^{4\times 1}$ is a random error vector. Now, assuming complete information about $\boldsymbol{H}$ and with a condition that $\boldsymbol{H}$ is full rank, we can estimate v_i as follows:

$$\begin{aligned}
&\boldsymbol{H}_i^T \boldsymbol{z}_i(t) = \boldsymbol{H}_i^T \boldsymbol{H}_i v_{i-1}(t) + \boldsymbol{H}_i^T \boldsymbol{e}_i(t) \\
&\Rightarrow \left[\boldsymbol{H}_i^T \boldsymbol{H}_i\right]^{-1} \boldsymbol{H}_i^T \boldsymbol{z}_i(t) = \left[\boldsymbol{H}_i^T \boldsymbol{H}_i\right]^{-1} \boldsymbol{H}_i^T \boldsymbol{H}_i v_{i-1}(t) \\
&+ \left[\boldsymbol{H}_i^T \boldsymbol{H}_i\right]^{-1} \boldsymbol{H}_i^T \boldsymbol{e}_i(t) \\
&\Rightarrow v_{i-1}(t) = \underbrace{\left[\boldsymbol{H}_i^T \boldsymbol{H}_i\right]^{-1} \boldsymbol{H}_i^T \boldsymbol{z}_i(t)}_{\hat{v}_{i-1}(t)} - \left[\boldsymbol{H}_i^T \boldsymbol{H}_i\right]^{-1} \boldsymbol{H}_i^T \boldsymbol{e}_i(t),
\end{aligned} \tag{3}$$

where $\hat{v}_{i-1}$ is the estimated velocity. Now, by defining $\hat{\boldsymbol{z}}_i \triangleq \boldsymbol{H}_i \tilde{v}_{i-1}$ as the estimated measurement vector, we can find the measurement estimation error or residual as $\tilde{\boldsymbol{z}}_i \triangleq \boldsymbol{z} - \hat{\boldsymbol{z}}_i$. Next, we can define a weighted cost function for the measurement residual as follows:

$$J_i(\tilde{\boldsymbol{z}}_i) \triangleq \tilde{\boldsymbol{z}}_i^T \boldsymbol{W}_i \tilde{\boldsymbol{z}}_i = \left[\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\right]^T \boldsymbol{W}_i \left[\boldsymbol{z}_i - \hat{\boldsymbol{z}}_i\right], \tag{4}$$

where $\boldsymbol{W}_i$ is a positive definite square matrix. If the measurements are not dependent, a typical choice for $\boldsymbol{W}_i$ is to have positive diagonal components while the non-diagonal components are zero. Since in our model the sensor error are independent, we consider $\boldsymbol{W}_i$ to be a diagonal matrix in which w_k^i on the k-th row and column of $\boldsymbol{W}_i$ is the weight of measurement k. The estimator at each AV i must minimize the cost function in (4). It can be proven that the solution of this problem is given by [17]:

$$\bar{v}_{i-1}(t) = \left[\boldsymbol{H}_i^T \boldsymbol{W}_i \boldsymbol{H}_i\right]^{-1} \boldsymbol{H}_i^T \boldsymbol{W}_i \boldsymbol{z}_i(t). \tag{5}$$

Since we know that all the sensors can directly measure the speed, we can consider $\boldsymbol{H} = [1,1,1,1]^T$. Moreover, since the diagonal entities of $\boldsymbol{W}_i$ are weights assigned to each sensor reading, thus we can consider $\sum_{k=1}^{4} w_k^i = 1$. Now, (5) can be simplified to:

$$\bar{v}_{i-1}(t) = \frac{\sum_{k=1}^{4} w_k^i(t) z_k^i(t)}{\sum_{k=1}^{4} w_k^i(t)} = \sum_{k=1}^{4} w_k^i(t) z_k^i(t) = \boldsymbol{w}_i^T(t) \boldsymbol{z}_i(t), \tag{6}$$

where $z_k^i(t)$ is the k-th element of $\boldsymbol{z}_i(t)$, and $\boldsymbol{w}_i(t)$ is a vector with $w_i^k(t)$ as its element k.

B. Attack Model

In the studied system, an attacker is able to inject faulty data to any of the aforementioned sensor readings. Such

attacks can take place using special lasers to alter camera and radar readings as well as man in the middle attacks to inject bad data into the input of the AV and RSS beacons. We define $\tilde{\boldsymbol{z}}_i(t)$ as an "under attack sensor vector" which can be defined as $\tilde{\boldsymbol{z}}^i(t) \triangleq \boldsymbol{z}^i(t) + \boldsymbol{a}^i(t)$, where $\boldsymbol{a}^i(t)$ is the injected faulty data vector at time t to the sensor vector $\boldsymbol{z}_i(t)$. Thus, such attack will induce a deviation in the value of the speed estimation which can be derived from (6) as follows:

$$\tilde{v}_{i-1}(t)=\boldsymbol{w}_i^T(t)\tilde{\boldsymbol{z}}_i(t)=v_{i-1}(t)+\boldsymbol{w}_i^T(t)\boldsymbol{e}_i(t)+\boldsymbol{w}_i^T(t)\boldsymbol{a}_i(t). \quad (7)$$

Hence, the attacker can change AV $i-1$'s estimated speed at AV i by injecting faulty data. However, to stay stealth, the attacker cannot inject any arbitrary data due to the physical limitations of the system. For instance, at each time step the attacker cannot report a very high or low speed to AV i. Moreover, due to the difference in the sensor types (camera image, radar reading, beacons), the attacker cannot manipulate the sensors equally. Thus, we consider threshold levels for each sensor k such that $|a_k^i(t)| < \tau_k^i$ where $a_k^i(t)$ is the data injected to AV i's sensor k.

III. Cyber-physical Security Problem and Game Formulation

From (7), we can see that the AV i's estimated speed at each time step is a function of the actual AV i's speed, $v_{i-1}(t)$, as well as the noise, $\boldsymbol{e}_i(t)$, the weighting $\boldsymbol{w}_i(t)$, and the attack $\boldsymbol{a}_i(t)$ vectors. Thus, using (1) we can see that each AV i's speed $v_i(t)$ is also a function of $v_{i-1}(t)$, $\boldsymbol{e}_i(t)$, $\boldsymbol{w}_i(t)$, and $\boldsymbol{a}_i(t)$. Here, we analyze the spacing $d_i(t)$ between AVs i and $i-1$, before we subsequently investigate the optimal safe spacing for the AVs. It can be shown that the derivative of $d_i(t)$ is the difference between the AVs' speed, as follows:

$$\dot{d}_i(t) = v_{i-1}(t) - v_i(t, \boldsymbol{a}_i(t), \boldsymbol{w}_i(t), \boldsymbol{e}_i(t)). \quad (8)$$

Thus, the spacing at each time step is a function of the AVs' speed as well as the sensor readings and the attack vector. Such attack vector can manipulate the spacing $d_i(t)$ yielding two effects on the ITS: a) if $d_i(t)$ decreases, the risk of collision between AVs increases and b) if $d_i(t)$ increases, the traffic flow will reduce, which will be non-optimal and ineffective for the ITS operation. Therefore, the attacker's goal is to manipulate the spacing and deviate it from the optimal safe state while staying stealthy. In contrast, AV i tries to optimize its operation while staying robust to such sensor manipulations to minimize the spacing deviations. Formally, the attacker's goal is to find an attack vector $\boldsymbol{a}_i^*(t)$ at each time step t such that:

$$\boldsymbol{a}_i^*(t)=\max_{\boldsymbol{a}_i(t)} R_i(\boldsymbol{a}_i(t), \boldsymbol{w}_i(t), \boldsymbol{e}_i(t)) \triangleq (d_i(t) - o_i(v_{i-1}(t)))^2, \quad (9)$$

$$\text{s.t. } |a_k^i(t)| < \tau_k^i \ \forall k = 1, \ldots, 4 \quad (10)$$

where $R_i(t)$ is AV i's *regret* function which quantifies the deviation from the optimal safe spacing and $o_i(v_{i-1}(t))$ is the optimal safe spacing at time t. Conversely, AV i's objective is to find a weighting vector $\boldsymbol{w}_i^*(t)$ to minimize the defined regret function as follows:

$$\boldsymbol{w}_i^*(t) = \min_{\boldsymbol{w}_i(t)} R_i(\boldsymbol{a}_i(t), \boldsymbol{w}_i(t)), \ \text{ s.t. } \sum_{k=1}^{4} w_k^i(t) = 1. \quad (11)$$

The optimization problems in (9) and (11) are dependent on the actions of both the attacker and the AV. Solving such problem requires taking into account the interdependence of AV and the attacker's actions. In the following we analyze the interdependence of the attacker and the AV's actions to each other and their previous actions and we formulate such problem in a game-theoretic framework [18].

To this end, we first derive the impact of past attacker and AV actions on their future actions. Next, we analytically derive a limit on the number of past regret samples which are enough to take future actions with T being the sampling period of the sensors.

Theorem 1. *The attacker and the AV can optimally choose their future actions if: (i) $\lambda T < 2$ and (ii) they have information about the regret for at least $\bar{n}$ past time steps, where $\bar{n}$ is the smallest integer that satisfies:*

$$\bar{n} \leq \frac{\log(\epsilon)}{\log(|1-\lambda T|)}, \quad (12)$$

where ϵ is a small value.

Proof. The proof is omitted due to space constraints. Please see [19] for the details. ■

Theorem 1 proves that in order to solve the optimization problems in (9) and (11), the AV and the attacker can only use their past $\hat{n}$ actions.

Proposition 1. *The spacing between AVs i and $i-1$ converges to an optimal safe spacing if AV $i-1$ must start following AV $i-1$ when the spacing is:*

$$d^*(\nu)=o(\nu) - (\hat{n}+2)T\nu + T\nu \sum_{p=0}^{\hat{n}-1} \left(1-(1-\lambda T)^p\right), \quad (13)$$

where $d^*(\nu)$ is the spacing when the AV i starts following AV i, and ν is the expectation of v_{i-1}, $\mathbb{E}\{v_{i-1}\} = \nu$.

Proof. The proof is omitted due to space constraints. Please see [19] for the details. ■

Proposition 1 shows that AV i must start following AV $i-1$ when it reaches a distance of $d^*(\nu)$ from AV $i-1$ while AV $i-1$'s speed is ν. Now, if $d_i(0) = d_i^*(\nu)$, we can formally define the regret function of (9) as follows:

$$R_i(n)=\lambda^2 T^4 \Big[\sum_{p=0}^{n} \sum_{l=0}^{\min\{\bar{n},p\}} (1-\lambda T)^l \Big(\boldsymbol{w}_i^T(p-l+1)\boldsymbol{e}_i(p-l+1) + \boldsymbol{w}_i^T(p-l+1)\boldsymbol{a}_i(p-l+1) \Big) \Big]^2, \quad (14)$$

where for notational simplicity we use $R_i(n)$ instead of $R_i(\boldsymbol{a}_i(n), \boldsymbol{w}_i(n), \boldsymbol{e}_i(n))$ as defined in (9). We can see that, at each time step n, the regret function accumulates the errors from the initial time step till n. Thus, if we define the *deviation* from optimal safe spacing at time step n as:

$$\delta_i(n) \triangleq \sum_{p=0}^{n} \sum_{l=0}^{\min\{\bar{n},p\}} (1-\lambda T)^l \Big(\boldsymbol{w}_i^T(p-l+1)\boldsymbol{e}_i(p-l+1) + \boldsymbol{w}_i^T(p-l+1)\boldsymbol{a}_i(p-l+1) \Big). \quad (15)$$

Then, we can derive the deviation as a process as follows:

$$\delta_i(n)=\delta_i(n-1)+\underbrace{\sum_{l=0}^{\min\{\bar{n},n\}}(1-\lambda T)^l\Big(\boldsymbol{w}_i^T(\min\{\bar{n},n\}-l+1)\boldsymbol{e}_i(\min\{\bar{n},n\}-l+1)+\boldsymbol{w}_i^T(\min\{\bar{n},n\}-l+1)\boldsymbol{a}_i(\min\{\bar{n},n\}-l+1)\Big)}_{\theta(\boldsymbol{w}_i,\boldsymbol{a}_i,\boldsymbol{e}_i)}. \quad (16)$$

Thus, we can write the regret function as follows:

$$R_i(n)=\lambda^2T^4\Big[\delta_i(n-1)+\theta\left(\boldsymbol{w}_i(n),\boldsymbol{a}_i(n),\boldsymbol{e}_i(n)\right)\Big]^2. \quad (17)$$

Then, at each time step n, the attacker and the AV must choose their associated vectors using their past $\hat{n}$ actions and the deviation from last step, $\delta_i(n-1)$.

We now formally define a noncooperative game where the players are the attacker and the AV, the AV's action $\alpha^{\text{AV}}(n)$ is to choose a weighting vector at each time step, $\boldsymbol{w}_i(n)$, and the attacker's action, $\alpha^{\text{att}}(n)$ is to choose a data injection vector at each time step, $\boldsymbol{a}_i(n)$. Moreover, the AV's utility function is $U^{\text{AV}}(n) = -R_i(n)$ while the attacker's utility function is $U^{\text{att}} = R_i(n)$. A suitable solution concept for the defined game is the so-called NE which a stable game state at which the AV cannot reduce the regret by unilaterally changing its action $\boldsymbol{w}_i(n)$ given that the action of the attacker is fixed. Moreover, at the NE, the attacker cannot increase the regret by changing its action $\boldsymbol{a}_i(n)$ while the AV keeps its action fixed. Since the players utility at each time step sum up to zero, the game is zero-sum and is guaranteed to admit at least one *mixed-strategy Nash equilibrium (MSNE)* [20]. A *mixed strategy* is a randomization between the available actions of the AV which satisfy $\sum_{i=1}^{n} w_i^k = 1$ and the available actions of the attacker which satisfy $|a_i^k| < \tau_k$, $\forall k = 1,\ldots,4$. Even though the MSNE exists for our game, it is analytically challenging to derive the equilibrium strategies. Thus, we next propose a deep RL algorithm for this game in which the AV and the attacker learn their optimal actions based on their time-varying observations of each others' actions.

IV. Adversarial Deep Reinforcement Learning for Optimal Safe AV Control

The proposed deep RL algorithm in Fig. 1 has two components: (i) A DNN that summarizes the past actions and spacing deviations and (ii) an RL component, which can be used by each player to decide on the best action.

To derive the AV and the attacker's actions that maximize their expected utility using RL, we use a Q-learning algorithm [21]. In this algorithm, we define a state-action value Q-function $Q^j(s^j,\alpha^j)$ which is the expected return of player j when starting at a *state* s^j and performing action α^j. To derive the maximizer action at each time step for each player, we use the following update rule for the Q function [21]:

$$Q^j_{n+1}(s^j(n),\alpha^j(n))=Q^j_n(s^j(n),\alpha^j(n))+\beta\Big[U^j(n+1)+\gamma\max_{\alpha^j}Q^j_{n+1}(s^j(n+1),\alpha^j)-Q^j_n(s^j(n),\alpha^j(n))\Big], \quad (18)$$

where β is the learning rate and γ is the discount factor. In our problem, $\alpha^{\text{att}} = \boldsymbol{a}_i(n)$ is the attacker's action while, $\alpha^{\text{AV}} = \boldsymbol{w}_i(n)$ is the AV's action. Moreover, since the players have no information about the other player's past actions and noise vector, the observed state for the AV is $s^{\text{AV}}(n) = \{\boldsymbol{w}_i(n-\hat{n}),\ldots,\boldsymbol{w}_i(n-1);\delta_i(n-\hat{n}),\ldots,\delta_i(n-1)\}$ while the observed state for the attacker is $s^{\text{att}}(n) = \{\boldsymbol{a}_i(n-\hat{n}),\ldots,\boldsymbol{a}_i(n-1);\delta_i(n-\hat{n}),\ldots,\delta_i(n-1)\}$. From (18), we can see that at each step, the players must find the action which maximizes Q^j_n. However, to find such action, each player must know all the possible states, and at each time step find the optimal action. However, in our problem, all the available states cannot be stored since, $\boldsymbol{w}_i(n)$, $\boldsymbol{a}_i(n)$, $\boldsymbol{e}_i(n)$, and $\delta_i(n)$ have continuous values, resulting in an infinite state space.

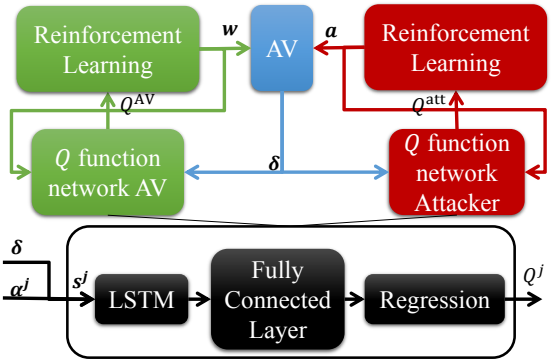

Fig. 1: Proposed adversarial deep RL algorithm.

To solve such a challenging problem, we use DNNs which are very effective at extracting features from large data sets. Particularly, we use LSTM blocks which are deep recurrent neural networks (RNNs) that can store information for long periods of time and, thus, can learn long-term dependencies within a given sequence [22]–[24]. Essentially, an LSTM algorithm processes an input sequence $s^j(n)$ by adding new information into a memory, and using gates which control the extent to which new information should be memorized, old information should be forgotten, and current information should be used. Therefore, the output of an LSTM algorithm will be impacted by the network activation in previous time steps. Thus, LSTMs are suitable for our problem in which we want to extract useful features from actions and deviation of previous time steps and reduce our state space. Thus, the proposed deep RL algorithm will use a DNN as shown in Fig. 1 to approximate the Q function for each player and using this Q-function we will choose optimal actions for each player from (18). Algorithm 1 summarizes the proposed adversarial deep RL approach that is used by each player to learn its optimal action vectors. Moreover, Fig. 1 shows the DNN architecture for the proposed adversarial deep RL algorithm. Using the proposed algorithm, we can find the optimal actions for the players which will converge to one of the MSNE points [21].

Algorithm 1 Adversarial Deep RL for Robust AV Control

1: Initialize two *replay memory* M^j that stores the past experiences of the players and two DNNs for Q^j.
2: Observe initial state $s^j(0)$ for both players.
3: **Repeat:**
4: Select an action α^j for each player j:
5: with probability ε select a random action,
6: otherwise select $\alpha^j = \arg\max_{\alpha'^j} Q^j(s^j(n), \alpha'^j)$.
7: Perform action α^j for both players simultaneously.
8: Observe utility $U^j(n+1)$ and new state $s^j(n+1)$.
9: Store *experience* $\{s^j(n), \alpha^j(n), U^j(n+1), s^j(n+1)\}$ in replay memory D^j for each player j.
10: Sample a random experience $\{\hat{s}^j(\eta), \hat{\alpha}^j(\eta), \hat{U}^j(\eta+1), \hat{s}^j(\eta+1)\}$ from the replay memory D^j for each player.
11: Calculate the *target* value t^j for each player j:
12: If the sampled experience is for $n = 0$ then $t^j = \hat{U}^j$,
13: Otherwise $t^j = \hat{U}^j + \gamma \max_{\alpha'^j} Q^j(\hat{s}^j(n+1), \alpha'^j)$.
14: Train the network Q^j for each player using: $[t^j - Q^j(\hat{s}^j(n), \hat{a}^j(n))]^2$.
15: $n = n + 1$.
16: **Until** convergence to an MSNE

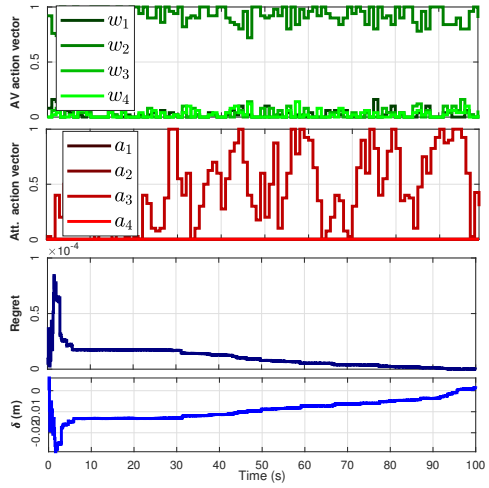

Fig. 2: The AV and the attacker's action, regret, and deviation for our proposed algorithm in the case where the attacker attacks only to the beacon information.

V. Simulation Results and Analysis

For our simulations, we choose a reaction parameter $\lambda = 1$ and a sampling period $T = 1$. Using Theorem 1, by choosing $\epsilon = 0.001$, we find $\hat{n} = 66$ which is equivalent to 6.6 seconds. As a result, each AV only needs the information about the past 6.6 seconds to be able to carry out an optimal safe action. Moreover, we consider that the sensor noise powers are arranged in a descending order as follows: RSS, radar, camera and beacon. This is due to the fact that the RSS might have the highest error for speed measurement while the beacon is sending the exact speed information from AV $i-1$ to AV i. In addition, we do not supply the information about noise statistics to the AV and the attacker. Thus, they both must learn such information during the interaction with each other. Moreover, we consider that the attack threshold levels are $\tau_1 = 0.5$, $\tau_2 = 1$, $\tau_3 = 1$, and $\tau_4 = 1.5$ (m/s).

First, we consider a case in which the attacker can only attack beacon values as it is one of the most studied attacks in the literature. Also, since the beacon has the lowest error power, the ideal case for the AV is to put the highest weight on beacon information in the absence of the attacker. Fig. 2 shows the action vector for the AV and the attacker when they interact for the first 100 seconds during which AV i follows $i-1$. From Fig. 2, we can see that, even though the beacon has the lowest error power, since the attacker attacks the beacons, the AV decides to put more weight on other sensors. Also, Fig. 2 shows that, although the attacker can always have a data injection that is equal to the threshold level, $\tau_3 = 1$, it can sometimes decide to inject lower values, to maximize the expected deviation. In addition, Fig. 2 shows that, in the first learning steps, the attacker can inject deviations in the spacing thus increasing the regret for the AV. However, our proposed algorithm enables the AV to mitigate the error on the estimation and thus stay robust to the data injected attack. Hence, after 100 seconds, we can see that the regret reduces to zero and the attacker cannot force the AV to deviate from its optimal safe spacing.

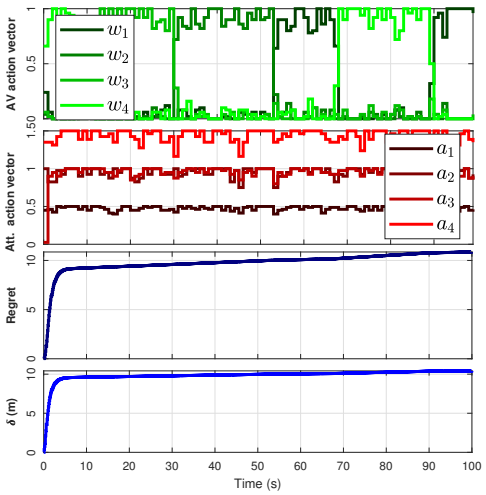

Fig. 3: The AV and the attacker's action, regret, and deviation for our proposed algorithm for the case in which the attacker attacks all the sensors.

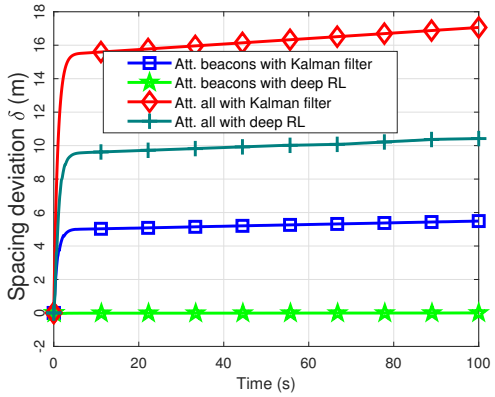

Fig. 4: Comparison of the proposed deep RL algorithm with a baseline that does not use any learning process.

Next, we consider the worst case security scenario in which the attacker can attack all of the sensor readings. Fig. 3 shows the AV and the attacker action process during the first 100 seconds of car-following. In this case, we can see from Fig. 3 that the attacker can attack to all the sensor values and

thus, the AV cannot prioritize between the sensor readings as in the previous simulation. Thus, Fig. 3 shows that the AV tries to assign higher weights to one step in small time periods to deceive the attacker. In contrast, the attacker tries to maximize the value of injected data as seen from Fig. 3 that the injected data values are close to the threshold level. Moreover, Fig. 3 shows that in the first 10 seconds the value of regret has an abrupt increase, while in the remaining time the regret stays almost constant. Also, the spacing deviation reaches a value close to 10 meter. This means that, when the attacker can attack all of the sensor values, the AV cannot make the estimation robust to the injected attack, however the regret stays approximately constant. Thus, the AV can feedback the spacing deviation δ to its car following model to compensate the deviation from the optimal safe spacing by changing the speed and thus make the AV resilient to such data injection attacks.

In Fig. 4, we show the spacing deviation as a function of time. In this figure, we compare our proposed deep RL algorithm with a baseline scenario, where the AV knows the noise distributions and choose a static weighting vector $\boldsymbol{w}_i$ using a *Kalman* filter. Fig. 4 shows that, even though the used Kalman filter converges to a constant spacing deviation, however, our proposed deep RL algorithm has a lower steady state deviation than the Kalman filter. This is due to the fact that the Kalman filter only takes into account the noise power, however, our proposed algorithm uses an adversarial approach to learn the attacker's action. This, indeed, enables the AV to minimize the deviation from the optimal safe spacing and remain more robust to the attacker.

VI. Conclusion

In this paper, we have proposed a novel deep RL method which enables a robust dynamics control for AVs in presence of data injection attacks on their sensor readings. To analyze the incentives of attacker to attack on the AV data and address the AV's reaction to such attacks, we have formulated a game-theoretic problem between the attacker and the AV. We have then shown that deriving the mixed strategies at Nash equilibrium is analytically challenging. Thus, we have used our proposed deep RL algorithm to learn the optimal sensor fusion for the AV at each time step that results in minimizing the deviation from an optimal safe inter-vehicle spacing. In the proposed deep RL algorithm, we have used LSTM blocks which can extract temporal features and dependence of AV and attacker actions and deviation values and feed them to a reinforcement learning algorithm. Simulation results have shown that, using the proposed deep RL algorithm, an AV can mitigate the effect of data injection attacks on the sensor data and thus stay robust to such attacks.

References

[1] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems," *arXiv preprint arXiv:1712.04135*, 2017.

[2] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 3949–3963, June 2016.

[3] T. Zeng, O. Semiari, W. Saad, and M. Bennis, "Joint communication and control for wireless autonomous vehicular platoon systems," *arXiv preprint arXiv:1804.05290*, 2018.

[4] F. Kargl, P. Papadimitratos, L. Buttyan, M. Mter, E. Schoch, B. Wiedersheim, T. V. Thong, G. Calandriello, A. Held, A. Kung, and J. P. Hubaux, "Secure vehicular communication systems: implementation, performance, and research challenges," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 110–118, November 2008.

[5] A. Ferdowsi, W. Saad, and N. B. Mandayam, "Colonel Blotto game for secure state estimation in interdependent critical infrastructure," *arXiv preprint arXiv:1709.09768*, 2017.

[6] A. Ferdowsi, W. Saad, B. Maham, and N. B. Mandayam, "A Colonel Blotto game for interdependence-aware cyber-physical systems security in smart cities," in *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering*, ser. SCOPE '17. Pittsburgh, Pennsylvania: ACM, 2017, pp. 7–12.

[7] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Proc. of IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, June 2011, pp. 528–533.

[8] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, April 2015.

[9] H. Chaudhry and T. Bohn, "Security concerns of a plug-in vehicle," in *Proceedings of IEEE PES Innovative Smart Grid Technologies (ISGT)*, Washington, DC, USA, Jan 2012, pp. 1–6.

[10] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaniche, and Y. Laarouchi, "Survey on security threats and protection mechanisms in embedded automotive networks," in *Proc. of IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, Budapest, Hungary, June 2013, pp. 1–12.

[11] T. Kim, A. Studer, R. Dubey, X. Zhang, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Vanet alert endorsement using multi-source filters," in *Proceedings of the Seventh ACM International Workshop on VehiculAr InterNETworking*, Chicago, IL, USA, September 2010, pp. 51–60.

[12] M. Sun, M. Li, and R. Gerdes, "A data trust framework for vanets enabling false data detection and secure vehicle tracking," in *Proc. of IEEE Conference on Communications and Network Security (CNS)*, Las Vegas, NV, USA, Oct 2017, pp. 1–9.

[13] A. Petrillo, A. Pescap, and S. Santini, "A collaborative approach for improving the security of vehicular scenarios: The case of platooning," *Computer Communications*, vol. 122, pp. 59 – 75, 2018.

[14] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 534–545, April 2015.

[15] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "On the performance of secure vehicular communication systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 6, pp. 898–912, Nov 2011.

[16] M. Brackstone and M. McDonald, "Car-following: a historical review," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 2, no. 4, pp. 181 – 196, 1999.

[17] R. C. Dorf and R. H. Bishop, *Modern control systems*. Pearson, 2011.

[18] Z. Han, D. Niyato, W. Saad, T. Başar, and A. Hjørungnes, *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge University Press, 2012.

[19] A. Ferdowsi, U. Challita, W. Saad, and N. B. Mandayam, "Robust deep reinforcement learning for security and safety in autonomous vehicle systems," *arXiv preprint arXiv:1805.00983*, 2018.

[20] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.

[21] J. Heinrich and D. Silver, "Deep reinforcement learning from self-play in imperfect-information games," *arXiv preprint arXiv:1603.01121*, 2016.

[22] A. Ferdowsi and W. Saad, "Deep learning for signal authentication and security in massive Internet of Things systems," *arXiv preprint arXiv:1803.00916*, 2018.

[23] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, 2017.

[24] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013.