This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

Adversarial Attacks and Defense in Deep Reinforcement Learning (DRL)-Based Traffic Signal Controllers

Ammar Haydari, *Student Member, IEEE,* Michael Zhang, Chen-Nee Chuah, *Fellow, IEEE*

***Abstract*—Security attacks on intelligent transportation systems (ITS) may result in life-threatening situations. Combining deep neural networks with reinforcement learning (RL) models called DRL shows promising results when applied to urban Traffic Signal Control (TSC) for adaptive adjustment of traffic light schedules. In this paper, first, we explore the security vulnerabilities of DRL-based TSCs in the presence of adversarial attacks. We investigate the impact of the two distinct threat models with two state-of-the-art adversarial attacks using white-box and black-box settings. The attacks are simulated on different DRL-based TSC algorithms in a single intersection and multiple intersections. The results show that the performance of the DRL learning agent decreases in both adversarial attack models with white-box and black-box settings resulting in higher levels of traffic congestion. After analysing the adversarial attack models, we explored several sequential anomaly detection models. While sequential anomaly detection models minimizes the detection delays, it also achieves lower false alarm rates due to cumulative anomaly inspection. We also proposed an ensemble model that works with all the attack models without any model assumption. The results of anomaly detectors indicates that low-cost ensemble model achieves the best anomaly detection performance in all attack models and DRL settings.**

***Index Terms*—Deep reinforcement learning, Statistical Anomaly Detection, Traffic signal control, Adversarial attack, Security.**

I. Introduction

In recent years, data-driven approaches are often used to drive the design and performance evaluation of different control algorithms in Intelligent Transportation System (ITS). With the proliferation of such data-driven models and communication technologies, Information and Communication Technology (ICT) have revolutionized ITS by connecting different components: vehicles, road-side units and sensors, cameras, loop detectors and control modules such as ramp meters, traffic signal controllers via vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. In addition, some in-vehicle and road-side units are also connected to wide-area Internet via 4G/5G cellular technologies.

This work is supported by Department of Transportation Center for Transportation, Environment, and Community Health (CTECH) and NSF HDR:TRIPODS grant CCF-1934568.

Ammar Haydari and Chen-Nee Chuah are with the Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616 USA (e-mail:ahaydari@ucdavis.edu; chuah@ucdavis.edu).

Michael Zhang is with the Department of Civil and Environmental Engineering, University of California, Davis, CA 95616 USA (e-mail: hmzhang@ucdavis.edu).

Learning-based control mechanisms in ITS, such as traffic flow control systems, travel demand prediction, and autonomous vehicles, take action based on real-time data from the environment. Traffic signal controller (TSC), which schedules the green/yellow/red phases at road intersections, plays a critical role in ITS, especially in busy urban settings. Control loops like TSCs often use real-time traffic information (e.g., captured by local cameras/sensors or broadcast messages from vehicles) to perform intelligent control decisions. This opens up the attack surface. Cybersecurity attacks such as falsified data may lead to erroneous control decisions, jeopardizing the safety and efficient operation of the transportation system corridor. Mitigating risks due to those issues remains an open and active research area.

Machine learning (ML)-based learning models are classified into supervised learning, semi-supervised learning, unsupervised learning, and reinforcement learning (RL). The first three approaches use labeled or unlabeled training datasets to identify patterns and create models to discriminate between different output classes. On the other hand, RL learns by interacting with the environment and the actions are rewarded or penalized. The environment is typically stated in the form of a Markov decision process (MDP). RL agents exploit the knowledge to make cognitive choices, such as decision making and scheduling [1]. Today, popular learning-based controller approaches combine deep neural networks (DNN) with RL, referred as DRL, in which policy estimation is performed by neural networks. One good example application of such methods in ITS is estimating the optimal light schedules of TSCs. In general, learning-based TSCs perform better than standard dynamic TSCs in terms of delay and throughput for isolated single-intersection and multi-intersection settings [2].

Learning based intelligent TSC agent collects messages from environment and schedules the traffic according to demand. Recently, many DRL-based data driven solution methods are proposed in the literature for controlling TSCs in a network of intersections and a successful cyber-attack targeting such TSCs can cause chaos in cities. Regardless of the underlying technology (WAVE or 5G) for V2V or V2I communications, the defense mechanisms of learning based TSCs needs thorough investigation.

Learning-based TSCs may make wrong decisions or take wrong actions in the presence of adversarial attacks. In more advanced attack models known as insider attacks, attacker falsifies the data input by considering the target DNN structure of the learning model. There are two distinct adversarial attack

settings on learning agents: white-box attack where attackers have access to the training model of learning agent and interacts with target model for generating adversarial inputs, and black-box attack where malicious inputs are generated from an estimated training model which is close to the true target model of learning agent [3]. In this paper, we thoroughly investigate security vulnerabilities of DRL based TSCs under two adversarial attack models namely Fast Gradient Sign Method (FGSM) [4] and Jacobian-based Saliency Map Attack (JSMA) [5] with white-box and black-box settings. We, then, propose an online anomaly detection algorithm for detecting such adversarial attacks.

A. Adversarial Attacks on DRL-TSCs

The falsified data attacks generally designed with optimization techniques to identify which feature to perturb [6]. Similar to this analogy, the attack strategy in DRL targets DNN structures where policy of learning agent is calculated to find the minimum perturbation amount. There are two possible threat models for DRL-based TSCs; attack may be carried out in the cyber domain by directly accessing the input pipeline of DRL agent or attack may be launched over the communication network by releasing falsified data from actual devices or Sybil devices to mislead the learning agent. Since FGSM attack perturbs all the input features only a slight amount, this attack can be launched purely in the cyber-domain without considering physical traffic conditions by accessing the input gate of the DRL agent. On the other hand, JSMA adversarial attack selects specific feature dimensions to perturb based on the constructed saliency map. JSMA can achieve this by using compromised vehicles or creating Sybil vehicles to send falsified data to TSCs.

In order to assess the impact of these adversarial attacks on different DRL-based TSCs, we consider both value-based, namely Deep Q Network (DQN), and policy-gradient with actor-critic-based, advantage actor-critic (A2C), DRL algorithms. We simulate the following: (i) single-intersection TSC scenario trained with DQN and A2C approaches, and (ii) multi-agent grid like 4-intersection TSC scenario trained with A2C approach, referred to as MA2C-DRL. Since the black-box attack assumes attacker does not have access to the actual target DNN model, we trained a separate DRL agent with different traffic demands and DNN settings for black-box attack. All the experiments are performed using a realistic SUMO traffic simulator. Detailed analysis shows that DRL-based TSCs are vulnerable to cyber-attack with or without knowledge of the trained DNN models.

B. Defense Mechanisms Against Adversarial attacks on DRL-TSCs

Adversarial attack surface for targeting DRL agents is very broad. Therefore protecting DRL agents against adversarial attacks is a challenging task. There are two general protection mechanism for DRL agents: (i) the agent builds a defense mechanism within the agent model that increases the robustness of DRL agent against the attacks, (ii) the agent is equipped with an external detection mechanism that detects the anomalies and raises an alarm. One possible mitigation strategy for external anomaly detectors is changing the controller model from learning-based one to another model such as max-pressure TSC [7] or actuated TSC [8]. Since gradient-based adversarial attacks such as FGSM and JSMA generally have a minimal perturbation on the data, it is also hard to differentiate adversarial samples from real samples with standard anomaly detectors.

Given the adversarial attacks FGSM and JSMA for single intersection and multi-intersection scenarios discussed in the previous subsection, we studied the performance of statistical anomaly detectors to detect even infinitesimally small anomalies. An ensemble anomaly detector that combines two sequential anomaly detection models and an autoencoder-based anomaly detection model with CUSUM-like detection model is evaluated on the gradient-based adversarial attacks. The experiments show that proposed ensemble sequential anomaly detection model achieves the best detection rate with different DRL agents and TSC scenarios.

C. Contributions

In this paper, we characterize the impact of two state-of-the-art adversarial attack models on DRL-TSCs and evaluate multiple statistical anomaly-based detection techniques. Our ensemble detection mechanism outperforms the other statistical anomaly detection models. The contributions of this paper can be summarized as follows.

- We demonstrate experimentally that both FGSM and JSMA adversarial attacks degrade the performance of DRL-based TSC agents as long as attack continues. White-box and black-box FGSM attacks have similar effects on TSC. However, black-box JSMA attack is less effective compared to white-box JSMA attacks.
- We developed and applied a sequential anomaly detection mechanism to the FGSM and JSMA adversarial attack on DRL-TSC scenarios with single intersection and multiple intersection models. The method combines multiple detection models in a computationally efficient method.
- The ensemble anomaly detection method is agnostic to both the model of the neural network policy and the type of adversary. Hence, the detection algorithm protects the DRL-TSC agents against different adversarial attack models.
- While different sequential anomaly detection models achieve the best performance on different attacks and DRL settings, our proposed ensemble model achieves the best detection performance on all the scenarios.

The rest of the paper is organized as follows. Section II discusses related work while Section III provides background for DRL learning agents and TSC settings. We present our adversarial attack models in Section IV and statistical anomaly detection model in V. We discuss our adversarial attack and defense results in Section VI and Section VII, respectively. Finally, Section VIII concludes the paper.

II. Related Work

Adversarial machine learning is an active research field for data scientists. Many attack models and defense mechanisms

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

have been studied by researchers for different ML models including DNNs [9]. DRL agents are vulnerable to different kind of adversarial attacks and detecting such adversarial attacks is a challenging task. In this section, we review the existing works on security of TSCs, DRL adversarial attacks and potential detection models.

A. Security of TSCs

Initial studies on adaptive TSC methods are rule-based or threshold-based control methods where predefined values of different traffic parameters such as queue or delay can trigger adaptive rules [10]. Lately, many machine learning-based TSC control mechanisms have been proposed. One such approach leverages DNN in a RL agent referred to as DRL and applies it to a network of traffic intersections [2]. The performance of learning based TSCs are generally better than standard TSC controllers.

There are many security analysis papers in literature for different type of TSCs. In [11], the authors identified some of the underlying threats against TSCs and proposed a game-theoretic risk minimization model without specifying the type of TSC. The study assumes that attacker has access to the control center and manipulates the traffic lights directly. Security of single intersection and multiple intersection back-pressure based TSCs is studied in [12]. The same group later extended their study with multiple attack strategies with several protection algorithms [13]. With the advanced vehicular and communication technologies, vehicles expected to communicate with the TSCs through Vehicular Ad Hoc Network (VANET). The security vulnerabilities of such VANET-based TSCs are investigated without considering a signal control mechanism in [6] where adversary uses decision three ML model to find the optimum perturbation. Although machine learning-based, especially DRL TSCs, offer promising performance gain, their security vulnerabilities need to be studied carefully. Apart from TSCs, there are various other studies on assessing the vulnerability of different ML-based ITS control mechanisms. Autonomous vehicles need to have a perfect perception while driving. Hence, deep learning has been exploited to process high-dimensional data. Since securing autonomous vehicles against malicious activities is an important and challenging task [14], the effects of adversarial attacks on DNN structures are studied in [15] where LIDARs of autonomous vehicles are under attack.

B. Adversarial attacks on DRL

There have been numerous studies on the adversarial attack models on the DNN policies of DRL agents. Adversarial attacks targeting DNNs are generally applicable to DRL agents. However, most of the DRL attack models are not applicable to DRL-TSC settings because it requires access to multiple parts of learning agent such as state, action and rewards and directly accessing the DRL-TSC components are challenging.

One of the earlier generative adversarial attack [16] targets the DNN classifier by perturbing the input data. The attack model is designed with constrained minimization approach using L_2 norm. Another constraint optimization adversarial attack for image classification task is proposed in [17]. Gradient-based adversarial attack models have promising results on DNN classifiers. Two well know gradient based adversarial attacks are FGSM [4] and JSMA [5] which deteriorate the performance of DNNs by crafting data input geared towards confusing the neural networks. These discussed adversarial attacks are know as the state of the art sequential adversarial attacks mainly proposed for DNNs.

Authors, in [18], presented a strategic attack reducing the number of attack times for DRL agents using random noise and FGSM attack strategies. With the transferability of neural networks, similar attack concepts can be extended to black-box attacks [19] and can target directly the DRL agents [20]. Since DRL agents estimate state values or policy values using DNNs, they are also vulnerable to adversarial attacks with white-box attack settings [21] and black-box attack settings [20]. A sequential adversarial attack for DRL agents is proposed in [22] in which adversarial samples are generated using adversarial transformer networks [23] on white-box attack strategy. Another strategic timing and target specific adversarial attack model for DRL agents is presented in [24]. The authors perturbed the input states selectively to reduce the visibility of attacker while achieving higher attack performance. Similar to our black-box attack settings, the authors in [25] injects perturbations from imitatively learned black-box model. There are also other adversarial attack models which are specific to application areas such as multi-agent robot interactions and path findings [26], [27].

C. Defense models for DRL

There are multiple defense options for the DRL agents including adversarial training, defensive distillation and adversarial detection. Adversarial training idea trains the learning model with adversarial samples that makes the learning model more robust. Several adversarial training-based defense mechanisms exist in literature for DRL agents [18], [28], [29]. However, adversarial training is attack dependent and it is easy to fool the model with a different attack strategy. Another defense model is called defensive distillation that trains the DRL policy with a different DNN model and transfers pre-trained soft-max layer from the other trained model to increase the robustness of DRL agents [30]. However it is already proven that bypassing the defensive distillation method is easy with various techniques [17]. The other security model, which is more aligned with our proposed detection model, is adversarial detection that distinguishes the adversarial samples from the clean samples without modifying the DRL model. One of the earlier adversarial attack detection mechanism for DRL agents is proposed in [31] where a defense mechanism detects the adversarial samples and suggests alternative actions for the DRL agent instead of the wrong action. A DNN-based adversarial sample detection model for DNNs is presented in [32]. The adversarial samples are classified and rejected by DNN models using the autoencoder reconstruction error similar to the robust autoencoder model [33].

Statistical properties of input data are susceptible to divergence after the perturbation. The study in [34] analyzes

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

two statistical distance measures, maximum mean discrepancy and energy distance, for detecting adversarial samples against several adversarial attacks including FGSM and JSMA. There are several adversarial detection models for DNN classifiers applicable to DRL agents [35], [36]. Sophisticated adversarial detection models for DRL agents are also proposed in literature [37], [38].

D. Summary

To date, there remains a limited understanding of the security vulnerabilities of learning-based ITS controllers and their impact on various operational performance metrics. In our paper, we experimented another research direction of ITS security where we characterize the security vulnerabilities of TSCs when implemented with DRL model and proposed a novel statistical detection model. Main-stream adversarial attack models continuously inject adversarial samples to the learning models and expects to fool the model quickly. To protect the DRL-TSC learning model we propose to use statistical sequential detection models with a novel ensemble detection algorithm that achieves the best detection performance in all cases.

III. Overview of DRL-based Traffic Signal Controllers

A. Deep Reinforcement Learning

Reinforcement learning (RL) is a trial-and-error based learning algorithm where agent interacts with the environment and takes action to maximize cumulative reward. Mathematical formulation of RL is based on Markov Decision Process (MDP). In general RL agent interacts with the environment and receives a numerical positive reward (penalty if it is negative). Continuously observing the state of the environment defined by s_t, taking action a_t, and receiving reward (or penalty) from the environment r_t, RL agent learns an action policy which defines how to behave by computing action value function $Q(s_t, a_t)$ after each iteration. In high dimensional environments, RL agent cannot estimate this action value functions easily. Through non-linear approximation, deep learning can easily estimate this function. Controlling RL agents with deep neural network based function approximations is called DRL. In this section, we explain how two popular DRL algorithms, DQN and A2C, work in the context of TSCs.

1) Deep Q-Network: Deep learning extracts the features from data with multi-layered neural networks. Tabular Q-learning method stores every state-action pair in a q-table, however, controlling agents in high dimensional systems with tabular methods is not tractable. The pioneering algorithm called Deep Q-Network (DQN) approximates state-action value function $Q(s_t, a_t)$ using non-linear DNN models, which maps N dimensional state inputs to M dimensional actions (output). An RL agent selects the best action from the outputs of DNNs [39] using Q-learning concept. Using DNNs for function approximation sometimes result in unstable learning performance. To ease this problem, temporal difference and batch learning techniques are used. A DRL agent is controlled with target network every k steps by updating the main network with respect to the target network. The agent may get stuck in a local optimal point due to recent trajectories but by randomly sampling stored experiments, DRL agent learns how to behave from a broad range of experiences.

2) Advantage Actor-Critic: Another main approach estimates policy function with gradient methods instead of estimating value function. However, policy gradient algorithms are not effective in large-scale applications due to high variance of the policy estimation. A general solution to this problem is to combine policy and value functions with an advantage function using two individual estimators, where the agent's behaviour is controlled with policy and the actions are balanced with value functions. These models are referred to as actor-critic RL. Synchronously updating both actor and critic estimators is known as advantage actor-critic (A2C) RL.

B. Deep Reinforcement Learning for TSC

In this section, we will discuss relevant DRL settings for single-agent and multi-agent settings. First, we will explain state, action and reward definitions, and then we will explain our collaboration technique for the multi-agent RL model.

In this application, the state of the environment is described as a vector of values for each incoming lane of the intersection. For one intersection, we created two valued vectors for each lane: one is average speed and the other is total number of vehicles. Position and speed of each vehicle can be collected from individual vehicles for calculating average speed and number of vehicles using V2I communication. Based on the information received from vehicles, the DRL agent in the TSC selects a green phase from among possible green phases. The TSC at a single intersection (such as Fig. 1) has four possible green phases: North-South Green (NSG), East-West Green (EWG), North-South Advance Left Green (NSLG), and East-West Advance Left Green (EWLG). Each selected green phase is executed after a yellow phase transition. With the objective of maximizing cumulative reward, a scalar reward is computed after each action (phase selection in this case). There are several reward definitions for TSC settings such as vehicle waiting time, cumulative delay, and queue length. In our DRL-based TSC, we used the change of the vehicle waiting time at an intersection for one cycle as a reward function.

As mentioned earlier, applying deep learning techniques to RL can help compute the action value functions more efficiently. For DRL models, designing a neural network structure for better performance is another critical step. Multi-layer perceptron (MP), i.e., the standard fully connected neural network model, is a useful tool for classic data classification. In this project, we used MP with 4 layers in DQN and 5 layers in A2C with relu and softmax activation functions for policy estimations of learning agents.

To test more general cases in DRL-based TSCs, we also studied a multiple intersection scenario with multi-agent A2C (MA2C) settings where the interaction among agents is necessary to reach a global optimal performance. In multi-agent settings, each agent updates its policy by including the current state and reward functions of neighbor TSCs as well to decrease the overall delay in traffic. For this purpose, global state

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

is found with concatenation of the local states of neighboring intersections and reward is generated by summing the local rewards of neighboring intersections.

IV. Adversarial attacks on DRL

In data-driven learning algorithms, a function estimator tunes the parameters precisely and carefully with respect to the training set. An adversary can manipulate the training set by injecting falsified data into the system. A smart way of attacking the learning agent is to inject carefully-crafted fake data that has very similar patterns with actual data. In the white-box attack model, the adversary has knowledge of the exact learning model and the corresponding output classes, and will manipulate the input to mislead the model. In the black-box attack model, the exact learning model is not known but the adversary can estimate a similar learning model to help generate input perturbation that can affect the target learning model.

In a DRL controller, the DNN function estimator, which estimates the action with respect to a given state, is the most probable adversarial target. The objective of the adversary is to craft the data input in order to lead DNN to a wrong action. When the DNN of DRL is under attack, it may select an incorrect action. For targeting DRL-based controllers, adversarial attacks can be launched sequentially at every time step to mislead the system as quickly as possible or strategically at specific time steps to hide itself from the controller center. In this study, we simulated sequential FGSM and JSMA attack strategies on DRL-based TSCs, which plays a critical role in traffic management systems. The threat model of adversarial attacks on DRL-TSCs is shown in Fig 1.

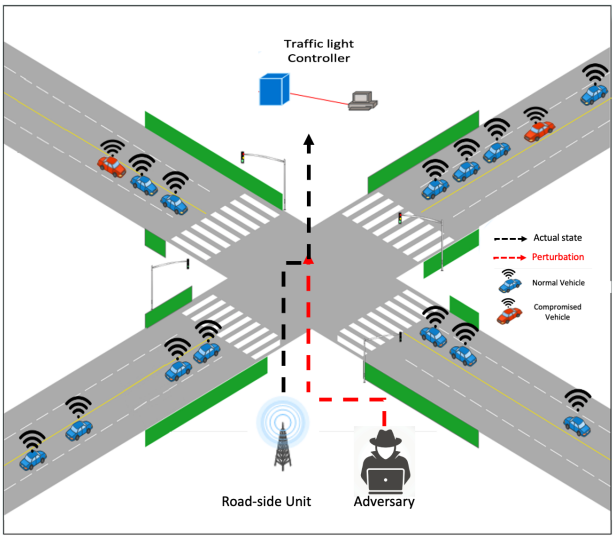


Fig. 1: TSC is controlled with a DRL agent and an adversary that can attack the agent with falsified data which perturbs the input state. While adversary can input directly for FGSM attack, it can use compromised vehicles for JSMA attack.

A. Fast Gradient Sign Method

A clever attack model, fast gradient sign method (FGSM) introduced in [4], calculates the gradient of the cost function with respect to DNNs to maximize the perturbation using the L_∞ distance. Adversarial input is generated by adding generated adversarial data to the input state as follow:

$$\eta = \epsilon * \text{sign}(\nabla_x J(\theta, \boldsymbol{x}, a)) \tag{1}$$

where ϵ is the attack magnitude, J is the cost function of DNN, and θ is the model parameters. ∇_x refers to the gradient of the cost function related to model input state $\boldsymbol{x}$, and true action a.

The FGSM attack designed to be fast and effective by generating infinitesimal perturbation that is close to the true input with perturbation parameter e.g., $\epsilon = 0.007$. FGSM attack model is untargeted where the attacker does not specify the target action when FGSM is launched. The optimal perturbation η satisfies $||\eta||_\infty < \epsilon$.

The perturbation amount η is added to the input data x:

$$\boldsymbol{x}_{adv} = \boldsymbol{x} + \eta. \tag{2}$$

In DRL-TSC, FGSM attack perturbs all the input features with very low values, therefore, launching this attack from the communication network requires to modify all the state dimensions that corresponds to each traffic lanes. The attack model assumes that the attacker has access to the input gate of DRL agents. By using this gate, an attacker perturbs the input state $\boldsymbol{x}$ right before it goes into the DNN where Q values for each action is estimated. Launching FGSM with the black box settings is also possible. In this case, the attacker will not be able to access to the DRL agent directly, it only has access to the data pipeline of the DRL agent.

B. Jacobian-based Saliency Map Attack

Another attack model, jacobian based saliency map attack (JSMA), utilizes forward derivative is presented in [5]. The intuition of JSMA attack is to find the influence of each state feature $\boldsymbol{x}_i$ to a specified output action a and then perturb only those specified feature dimensions. This influence relies on the jacobian matrix of outputs with respect to each action taken by the DRL agent using the forward gradient of the DNNs to construct adversarial saliency maps.

The adversary can control which input feature to perturb with respect to constructed saliency maps to achieve the desired goal. In this attack model, an attacker selects a target action for the DRL agent where the output of the DNN is Q values for each action. With a greedy mechanism, an action is selected from the DNN during the test phase with respect to given state $\boldsymbol{x}$ as:

$$a_t = \underset{a}{\text{argmax}}\, Q(\boldsymbol{x}, a) \tag{3}$$

where a_t refers to the selected action by thr DRL agent at time t.

In our case, an adversary tries to mislead the DRL agent to select a wrong action and for this purpose, the output Q value for the desired action should be increased. The Q values are the probabilities of corresponding actions. The adversary

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

can increase the desired Q values estimated through DNNs by using the saliency map:

$$S^{+}(x_{(i)},a)=\begin{cases}0 & \text{if } \frac{\partial f(\boldsymbol{x})_{(a)}}{\partial x_{(i)}}<0 \text{ or } \sum\limits_{a'\neq a}\frac{\partial f(\boldsymbol{x})_{(a')}}{\partial x_{(i)}}>0\\ \left(\frac{\partial f(\boldsymbol{x})_{(a)}}{\partial x_{(i)}}\right)\left|\sum\limits_{a'\neq a}\frac{\partial f(\boldsymbol{x})_{(a')}}{\partial x_{(i)}}\right| & \text{otherwise}\end{cases} \tag{4}$$

where i is the input feature of state $\boldsymbol{x}$, a is the action corresponding to the input, and a' is the other actions of DRL agent. In Equation 4, the first line of the expression rejects the negative target derivative with respect to action a and positive derivatives with respect to other actions a' of input state $\boldsymbol{x}$ feature i. The second line of Equation 4 extracts the positive forward derivative of state $\boldsymbol{x}$ of feature i given the action a. Based on the constructed saliency map, an adversary selects which input feature to perturb in order to mislead the agent for selecting the wrong action. Higher $S^{+}(x_{(i)},a)$ values mean the attacker can more easily determine if increasing this feature either increase the Q value of the target action a or decrease the Q values of other actions. In the JSMA model, the attacker first selects which action to perturb randomly then based on that selected action it creates the saliency map. Using the saliency map the attacker finds the best features to perturb.

The threat model for JSMA attack is different from the FGSM attack. Since JSMA perturbs specific features based on the saliency map, it is possible to launch this attack by compromising the communication between vehicles and TSC units. In this attack model, an attacker can use compromised vehicles and/or Sybil vehicles to broadcast falsified information in order to increase or decrease the corresponding feature dimension values.

V. Sequential anomaly detection for DRL-TSCs

The attackers can exploit a wide range of vulnerabilities in DRL-TSCs, and attack patterns are generally unpredictable. Therefore, it is hard to model a defense mechanism for a broad range of anomalies. Besides, defining a parametric model, which tries to fit a probability distribution to the data, is not practical. Due to life threatening effect of misbehaved DRL-TSCs, it is critical to detect and mitigate adversarial attacks in a timely manner. Considering the major challenges in DRL-TSC, non-parametric sequential anomaly detectors are suitable for detecting streaming anomalies in online settings. There are three main reasons why we employed a non-parametric sequential statistical anomaly detector for adversarial attacks on DRL-TSCs: (i) consecutive adversarial samples are more harmful for DRL controllers and need to be detected quickly, (ii) standard outlier detectors are susceptible to false alarms due to not considering temporal correlations in data, (iii) non-parametric sequential detectors have less miss-match error that results in lower detection error.

Statistical anomaly detectors operate by comparing the summary statistics extracted from the training set in offline phases and summary statistic of data in online phases for detecting potential anomalies. Since no single statistical property captures all anomaly types, we present a sequential anomaly detection model that extracts multiple summary statistics and leverages an ensemble model for the online test phase. In this section, we first explain three summary statistic extraction models that are distance-based, PCA-based and Robust Autoencoder-based and present the online sequential detection algorithm.

Let us first explain the data representation that is used for the rest of the paper. The monitoring system observes d dimensional data instance $\{\boldsymbol{x}_i^1,\ldots,\boldsymbol{x}_i^d\}$ that forms a set of nominal streaming data $\mathcal{X}=\{\boldsymbol{x}:j=1,2,\ldots,N\}$. Depending on the TSC setting and DRL model the size of d can change. In our experiments, DRL collects the summary statistics from each lane and forms the d dimensional state $\boldsymbol{x}_t$ at time t.

A. GEM-based Summary Statistic

The Geometric Entropy Minimization (GEM) method defines an acceptance region for the offline training set based on the nearest neighbor statistics with respect to significance level α [40]. A GEM-based computationally efficient summary statistic extraction method using bipartite kNN graph is presented in [41]. In the training phase, summary statistics extracted as described in the following.

We begin with randomly partitioning the anomaly free dataset $\mathcal{X}_N$ into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ with sizes N_1 and N_2 where $N=N_1+N_2$. Then, for each data point $\boldsymbol{x}_j\in\mathcal{S}_2$, we find the kNN euclidean distance e_j from $\mathcal{S}_1$. Sum of the distances of $\boldsymbol{x}_j$ to its nth nearest neighbor in $\mathcal{S}_1$ can be denoted as:

$$d_j=\sum_{i=1}^{k}e_j(i). \tag{5}$$

Once $\{d_j:\boldsymbol{x}_j\in\mathcal{S}_2\}$ is computed and sorted in ascending order, we refer to this baseline set as $\boldsymbol{D}_{GEM}$.

B. PCA-based Summary Statistic

High dimensional observation may exhibit sparse data structure so the underlying independent data dimension can be lower than the actual data dimension. When we represent data $\boldsymbol{x}_j$ in lower dimension as $\boldsymbol{y}_j$, the remaining parts $\boldsymbol{r}_j$ is the residuals. Adversarial noise injected to the actual data is mainly represented in residuals $\boldsymbol{r}_j$, hence the magnitude of the residuals $\|\boldsymbol{r}_j\|_2$ are expected to be higher than normal data. Recently a PCA-based online anomaly detection model is proposed in [42]. Based on this intuition, and the same partitioning strategy, we follow the PCA-based training steps for set $\mathcal{S}_1$.

1) Compute the sample mean $\overline{\boldsymbol{x}}$ and sample covariance matrix $\mathcal{Q}$
2) Then, compute the eigenvalue $\{\lambda_j:j=1,2,...,p\}$ and the eigenvectors $\{\boldsymbol{v}_j:j=1,2,...,p\}$ of $\mathcal{Q}$
3) Determine the dimension of $\boldsymbol{y}_t$, r, with respect to the desired level of data variance γ,
4) Form the eigenmatrix corresponding the largest r eigenvalues $\lambda_1,\lambda_2,...,\lambda_r$: $\boldsymbol{V}\triangleq[\boldsymbol{v}_1,\boldsymbol{v}_2,...,\boldsymbol{v}_r]$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

5) Compute the residual term $\boldsymbol{r}_{j-PCA}$ for every sample $\boldsymbol{x}_j$ in set $\mathcal{S}_2$ as follows:

$$\begin{aligned} \boldsymbol{y}_j &= \overline{\boldsymbol{x}} + \boldsymbol{V}\boldsymbol{V}^T(\boldsymbol{x}_j - \overline{\boldsymbol{x}}) \\ \boldsymbol{r}_{j-PCA} &= \boldsymbol{x}_j - \boldsymbol{y}_j \\ &= (\boldsymbol{I}_p - \boldsymbol{V}\boldsymbol{V}^T)(\boldsymbol{x}_j - \overline{\boldsymbol{x}}) \end{aligned} \tag{6}$$

6) Finally form the residual term vector $\boldsymbol{D}_{PCA}$ with $\{\|\boldsymbol{r}_{j-PCA}\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$ in ascending order.

C. Robust Deep Autoencoder Summary Statistic

A deep autoencoder-based noise and outlier extraction technique is proposed in [28] as an unsupervised Robust Deep Autoencoder (RDA) anomaly detection algorithm. The proposed RDA learns the normal data behaviours with a regularization penalty term using different norms. The idea of the RDA combines the powerful nature of the Robust PCA model [43] with autoencoders that recover low dimensional $\boldsymbol{y}_t$ iteratively by removing the residuals $\boldsymbol{r}_t$ from the data $\boldsymbol{x}_t$.

The training procedure of the RDA-based summary extraction model starts with pre-training the model with the sample set $\mathcal{S}_1$. After pre-training the model with certain number of episodes, which is 10 in our experiments, RDA is trained with sample set $\mathcal{S}_2$ and summary statistic $\boldsymbol{D}_{RDA} = \{\|\boldsymbol{r}_{j-RDA}\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$ is formed as a baseline .

D. Sequential Anomaly Detector

In the test phase, summary statistics d_{t-GEM}, $\|\boldsymbol{r}_{t-PCA}\|_2$ and $\|\boldsymbol{r}_{t-RDA}\|_2$ of each anomaly detection model is found for the new data point $\boldsymbol{x}_t$ independently. The anomaly score is expected to be higher in the case of adversarial attack. Since the procedure is the same for all three models, we explain the remaining anomaly statistic extraction algorithm for the GEM model as an example. For a new data point $\boldsymbol{x}_t$, once d_{t-GEM} summary score is computed using (5), tail probability of p_t would be computed with respect to baseline set $\boldsymbol{D}_{GEM}$ as follow:

$$p_t = \frac{1}{N_2} \sum_{\boldsymbol{x}_j = \mathcal{S}_2} \mathbb{1}\{d_j > d_{t-GEM}\} \tag{7}$$

which shows the fraction of the baseline summary statistics $\boldsymbol{D}_{GEM}$ greater than d_t. Given the significance level α, we can get a real valued statistical score in log scale with

$$s_{GEM} = \log(\frac{\alpha}{p_t}), \tag{8}$$

if the tail probability $p_t < \alpha$, we can consider $\boldsymbol{x}_t$ as an outlier. We follow the same approach in equations (7) and (8) to calculate s_{PCA} and s_{RDA} scores. Since the three scores are independent from each other, they can be calculated in parallel. For extracting the final anomaly score, we sanitized the three anomaly scores using a simple averaging as follows:

$$s_t = \frac{1}{3} \sum (s_{GEM}, s_{PCA}, s_{RDA}) \tag{9}$$

Note that the anomaly scores s_t can be positive or negative values with respect to the existence of anomalies. Instead of sample-by-sample anomaly declaration we propose to use a model-free CUSUM-like anomaly detection approach [44]:

$$\begin{aligned} g_t &\leftarrow \max\{0, g_{t-1} + s_t\}, \ g_0 = 0 \\ \mathcal{T} &= \inf\{t : \max\{0, g_t \geq h\} \end{aligned} \tag{10}$$

where g_t refers to the decision statistic. The anomaly is declared if enough sequential anomaly evidence is accumulated. The detection threshold h is chosen to strike a balance between minimum detection delay and lower false alarm rate. While a lower detection threshold h results in lower detection delay, it enables higher false alarm rates. The summary of the proposed anomaly detection technique is shown in Algorithm 1. The proposed sequential anomaly detector is also robust against system misbehaviour due to the nature of the cumulative anomaly detection model.

Algorithm 1 Proposed Nonparametric Anomaly Detection

Offline Phase

1: Partition the training set $\mathcal{X}_N$ into two subsets $\mathcal{S}_1$ and $\mathcal{S}_2$ with sizes N_1 and N_2.
2: Compute GEM baseline set $\boldsymbol{D}_{GEM} = \{d_j : \boldsymbol{x}_j \in \mathcal{S}_2\}$
3: Compute PCA baseline set $\boldsymbol{D}_{PCA} = \{\|\boldsymbol{r}_{j-PCA}\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$
4: Compute RDA baseline set $\boldsymbol{D}_{RDA} = \{\|\boldsymbol{r}_{j-RDA}\|_2 : \boldsymbol{x}_j \in \mathcal{S}_2\}$

Online Detection Phase

1: Initialization: $t \leftarrow 0$, $g_0 \leftarrow 0$.
2: **while** $g_t < h$ **do**
3: $t \leftarrow t + 1$.
4: Obtain the new data point $\boldsymbol{x}_t$.
5: Compute statistic s_{GEM}, s_{PCA} and s_{RDA}
6: Form ensemble statistic s_t with averaging as in (9)
7: $g_t \leftarrow \max\{0, g_{t-1} + \hat{s}_t\}$.
8: **end while**
9: Declare an anomaly and stop the procedure.

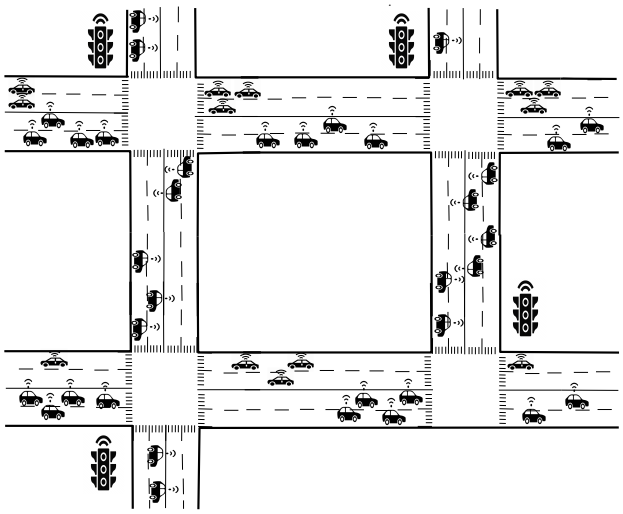


Fig. 2: Traffic scenario for multi-agent multi-intersection TSCs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

VI. Adversarial Attack Performance

In this section, we evaluated the impact of adversarial attacks on DRL-based TSCs using SUMO [45] real-time vehicular traffic simulator, with Tensorflow Python API[1] for DRL-based controller and CleverHans Python API for adversarial input generation built upon Tensorflow [46]. We simulated both a single-intersection and a multi-intersection environment with DQN and A2C DRL-based TSCs [1], [39]. For the single-intersection scenario, we observed similar results for DQN and A2C DRL-based TSCs. For the rest of the paper, we only present results for DQN for the single-intersection case. Value-based DQN approaches do not perform well for large environments. Therefore, we only simulated the multi-agent A2C (MA2C) model based RL controllers for multiple intersections traffic scenario organized in 2x2 grid topology.

Experimental settings for DRL-TSCs: A single intersection DQN-DRL agent has 4 incoming roads with 500 meters long. MA2C-DRL model coordinates multiple agents at 4 connected intersections as shown in Fig 2 with individual DRL agents. One traffic intersection has only 3 incoming roads while the other three intersections have 4 incoming roads. The roads connecting the different intersections are 1000 meters long, while the roads on the edges are 500 meters long. The traffic for both single intersection and multi-intersection is generated with the arrival rate of one vehicle per second spanning 1-hour simulation time. The DRL agent selects among four possible green phases as described in Section III-B. For each arrival, travel route is assigned with random origin and destination selection. We trained both DQN and MA2C agents on the same parameters with 2000 experience replay buffer size, $\gamma = 0.95$ discount factor, 0.00001 learning rate for DQN and actor network and 0.000005 learning rate for critic network, respectively. We applied the same DRL configurations for all attack experiments.

We implemented FGSM and JSMA adversarial attacks for both white-box and black-box attacks. One technical challenge we faced is the lack of computational resources to launch these adversarial attacks continuously (for more than 5 episodes), as it requires high memory footprints due to the batch gradient of the NNs[2]. All our experiments compare the performance of DRL TSCs with three baselines. One of the baselines is standard fixed time TSC where traffic lights are allocated to different phases with pre-defined durations. We also compared our method with two adaptive controller methods: queue-based actuated TSC, and max-pressure-based TSC [8]. Maximum phase duration for both actuated controller and max-pressure controller is set to be 45 seconds. All the attacks experimented in this paper starts after 15 episodes and the attack continues for 5 episodes, where every episode spans one hour of traffic simulation. After the attack terminates, we observed the performance of the learning agent for an additional 20 episodes. In the absence of attack, DQN achieves the second lowest total waiting time (only slightly inferior to Maxpressure) for the single-intersection case while multi-agent A2C model achieves the lowest total waiting time for multiple-intersection scenario.

[1]Allows to create and train ML models without loss of speed or performance.

[2]We employed transfer learning while simulating adversarial attacks on both single-intersection DQN and multi-intersection A2C scenarios. We saved the NN model weights after training the agents, and launched the attack using the latest NN weights. We repeated this for each attack episode.

A. White-box Insider Attack

Regardless of the DNN structure, learning models are vulnerable to white-box adversarial attacks, even with a very slight perturbation on input data. White-box adversarial attacks assume that the attacker has access to the target model of learning policy.

1) Attack Model: Using the target model, launching an adversarial attack with FGSM and JSMA models on the DNN of RL agents is possible.The adversary launches the attacks on DRL-based TSCs by injecting anomaly to the original input state. Since DNN is the policy of a learning agent, selecting correct action of the DRL agent will be affected by the white-box attack.

For FGSM attack, an attacker will perturb the input state with very small changes that are invisible by the controller. As pointed out in the original FGSM paper [4], minimal perturbation leads to the DNN to classify output to a wrong class. We used the same attack magnitude $\epsilon = 0.007$ as in the original FGSM attack [4] for DQN and A2C TSC simulations.

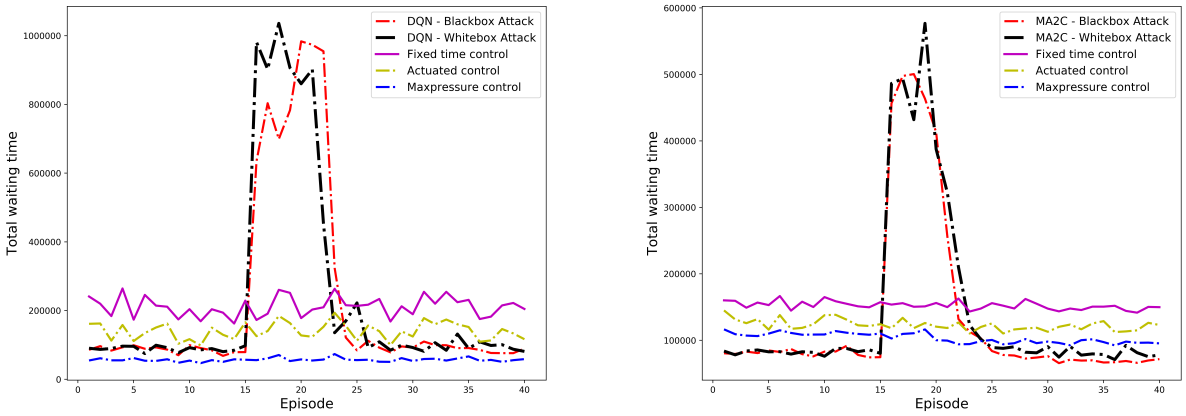
For JSMA attack, the attacker constructs the saliency map of given input state with respect to randomly selected action using the forward gradient of the DNN. In this attack model, we found that the attacker needs to perturb at least 40% of the feature dimensions to mislead the DRL agent, hence, we selected $\gamma = 0.4$ as an input parameter for our experiments.

2) Results: Fig. 3 and Fig. 4 show the results from FGSM and JSMA, respectively. After the attack is launched, both DQN and A2C TSCs perform poorly during the attack duration with FGSM and JSMA attacks. Although DRL settings are different, single-intersection TSC (Fig. 3(a) and 4(a)) and multi-agent multi-intersection TSC (Fig. 3(b) and 4(b)) are both affected, and the total waiting time in the traffic exceeds even the fixed-time controller. While the total waiting time increases almost 10x for single-intersection, it increases almost 6x for multi-intersection immediately after the white-box FGSM and JSMA attacks are launched. DRL agents cannot respond to these attack models and the attack continuously effects the learning agents as long as the DRL agent is targeted because the DQN and A2C agents do not recognize the attacks. For FGSM attack, the total waiting time decreases to pre-attack levels in 5 episodes after the attack ends in both the single-intersection DQN and the multi-intersection A2C cases. On the other hand, for JSMA attack, the total waiting time decreases to the pre-attack levels immediately right after the attack ends.

B. Black-box External Attack

In the black-box attack scenario, the attacker does not have a precise knowledge about the model. Here, we investigate the vulnerability of the DNN policies for DRL-based TSCs when the attacker does not have access to the actual target model.

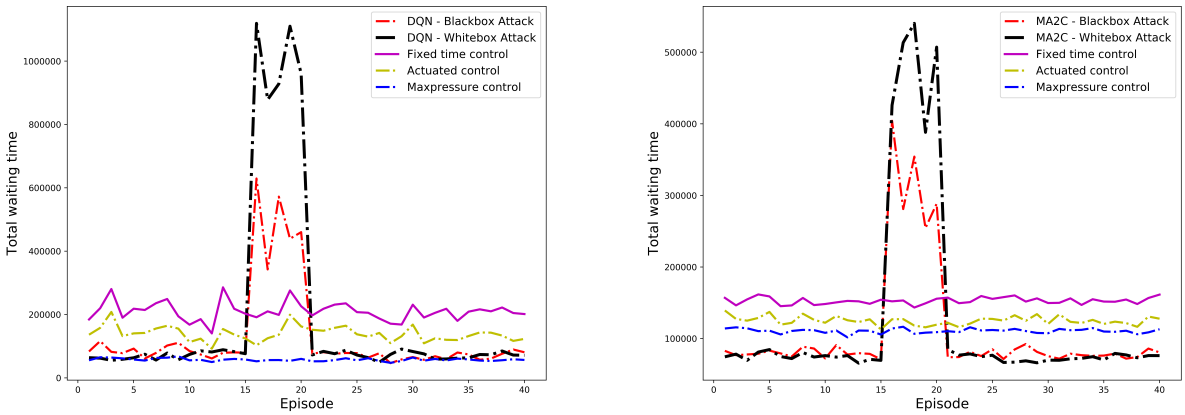
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems



(a) FGSM attack for single-intersection DQN model.

(b) FGSM attack for multiple intersections Multi-agent A2C model

Fig. 3: FGSM White-box and black-box attack results for DQN and multi-agent A2C with 0.007 attack magnitude using FGSM attack model. Attack continues 5 episodes from 15 to 20. Both white-box and black-box attacks continuously effects the performance of DRL agent while attack continues.



(a) JSMA attack for single-intersection DQN model

(b) JSMA attack for multiple intersection multi-agent A2C model

Fig. 4: JSMA attack continues a) with 10% of data perturbation for single agent DQN model and b) with 40% of data perturbation for multi-agent A2C model. The attack injects falsified data by selecting specific lanes of the intersection. The attack starts at episode 15 and lasts for 5 episode and ends in episode 20.

1) Attack Model: The transferability of trained DNNs allows attacker to train a separate learning model and use it to generate adversarial perturbation. Both FGSM and JSMA adversarial attacks require knowledge of DNNs for calculating gradients regarding to the DNN policy. Practically it is not hard to train a separate policy for TSCs using real traffic maps on traffic simulators, and an attacker can do this training at a very low cost. In this work, we are proposing a practical black-box attack strategy where the attacker uses the same number of layers for training a different DNN policy as the original learning agent. Also, for training a separate DNN, the attacker considers linear activation functions instead of the ReLU and Random Uniform DNN initialization technique instead of Glorot initialization [47]. We assumed that the attacker is not able to predict true travel demand on the simulator. Therefore, we trained our adversarial policy with slightly different traffic demands. Since we simulate the same adversarial attacks with black-box attack settings, to have a precise comparison, we kept the same attack magnitudes as $\epsilon = 0.007$ for FGSM attack and $\gamma = 0.4$ for JSMA attack similar to white-box attacks.

2) Results: The results of black-box adversarial attacks on DRL TSCs have similar patterns with the white-box attacks for FGSM attack model. However, the impact of the JSMA attack decreases to the half compared to white-box JSMA attacks in terms of the total waiting time. The results for the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

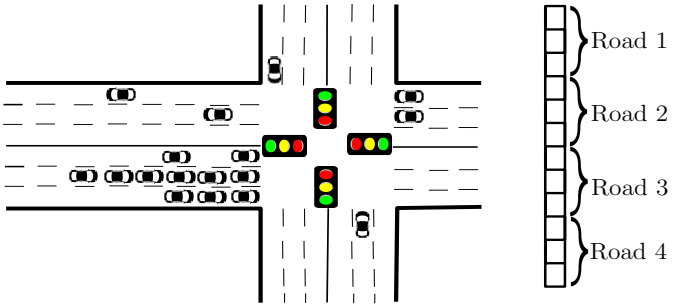


Fig. 5: Feature-based vectorized state representation.

three baseline TSCs are almost identical across two adversarial attack models. Red dashed lines in Fig. 3 and Fig. 4 shows the adversarial attack results for DQN and multi-agent A2C under black-box attacks. Similar to the white-box settings, the DRL agent is severely impacted by the attack resulting in average 9x and 6x increase in total waiting time in single and multi-intersection scenarios respectively during the FGSM attack. The black-box JSMA attack increases the total waiting time 5x and 3x for single intersection and multi-intersection scenarios likewise FGSM attack. The impact of the attack continues throughout 5 attack episodes by performing worse than the other three control methods in both attack cases. Similar to the white-box attack case, while the recovery period of DRL agent under FGSM attack is about 4 episodes after the episode 24th, DRL agent recovers itself immediately after the attack terminates for JSMA attack.

C. Robustness Against Noise

1) Noise Injection Model: After assessing the vulnerability of DRL-TSCs against specific adversarial attacks, it is also essential to test the performance of TSCs in the presence of intrinsic noise. The impact of the noise on the learning agents is discussed in terms of the action selection [48] and state observation [49]. Here, we evaluate the performance of the DRL-TSC controller when additive noise is injected into the state observation as measurement noise. For each experiment, noise is injected with fixed zero mean and varying standard-deviations between 0.05 and 0.6. We followed the same attack procedure as we did for FGSM and JSMA attacks and injected noise to the DRL agent after training with 15 episodes. Also, zero mean noise might result in some state features to go below zero. To prevent this, negative values are filtered with floor function. There are two scenarios with noisy state observation. The observation might be noisy in training or in the implementation phase due to unexpected conditions of the environment.

2) Results: We first trained our DRL agent with noisy data. It is seen that both DQN and MA2C agents are robust to measurement noise up to a certain standard deviation: 0.4 for the DQN agent and 0.2 for the MA2C agent. The DRL performance fluctuates in the first few episodes during training. Noise larger than 0.4 for DQN and 0.2 for MA2C results in a higher total waiting time and hence increased congestion. The DRL agent learns how to behave in a noisy environment with a lower noise magnitude and reaches optimal performance after enough training. Regarding the implementation phase, noise above the 0.2 standard deviation affects the single intersection DQN agent. On the other hand, the noise deteriorates the performance of the MA2C agent after 0.1 standard deviation. Besides, the additive noise impacts the DRL agent for DQN and MA2C continuously during the noise injection phase. After the noisy episode ends, the DRL agents behave as the second-best controller following the max-pressure for single intersection DQN and the best controller for multi-intersection MA2C. The higher noise has a twofold impact on DRL-TSC. First, noisy data causes more congestion during the attack period, and it takes time for the DRL agent to return to normal behavior when the noise disappears.

VII. Adversarial Detection Performance

After showing the vulnerability of DRL-TSCs against adversarial attacks, we evaluate the proposed statistical anomaly detection model on DRL-TSCs. The nature of adversarial attack for white-box and black-box attack settings are almost the same in terms of the data perturbation. Therefore, in this section, we only evaluated the detection performance on white-box FGSM and JSMA adversarial attacks on single intersection DQN-TSC and multi-intersection MA2C-TSC (see Fig. 2). We also use the same attack magnitudes and DRL settings as described in Section VI for evaluating the performance of statistical detectors.

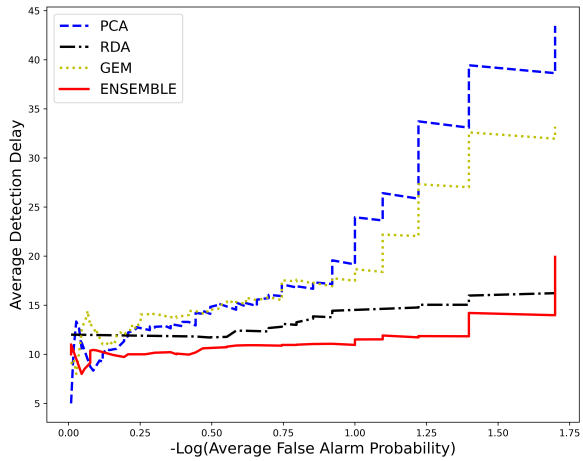
For evaluating the ensemble statistical detection performance, we compare the proposed algorithm with individual adversarial detectors PCA, RDA and GEM models. We use the same CUSUM-like detection structure on each model. Note that each anomaly detection algorithm is most effective in recognizing different anomaly types. While noise injections on all input vectors such as FGSM attacks can be detected by PCA anomaly detection model easily, selective perturbation-based anomalies such as JSMA can be detected with RDA and GEM models effectively.

We quantified the detection performance in terms of three metrics. Quick and accurate detection performance is presented with average detection delay vs false alarm rates, which is our first result representation. Later, we present the performance of sequential detectors on ROC (Receiver Operating Characteristics) curve and AUC (Area Under The Curve) scores which are the two leading performance metrics for classification tasks. While ROC is the probability curve for true positive rate vs false positive rate, AUC score quantifies how much the model is capable of distinguishing between classes.

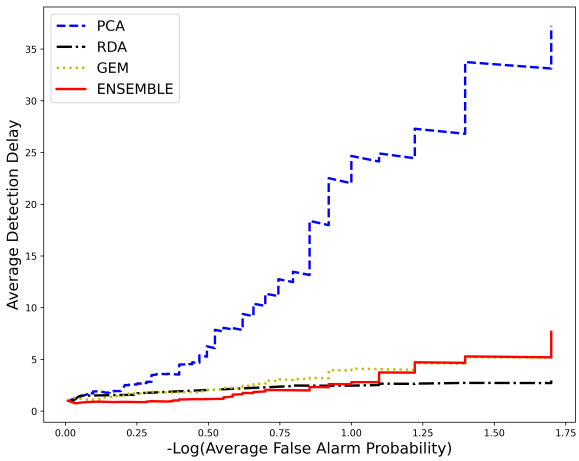
A. Sequential Detector Setup

To generate training and test sets for sequential detectors, we collect anomaly-free training states and test sets that include anomalies from the DRL-TSCs. For single intersection TSC model, the DRL setup has relatively low dimensional state format since each lane corresponds to two dimensions in state

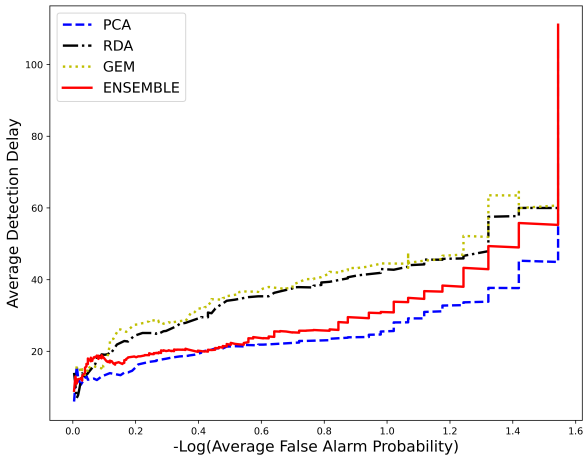
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems



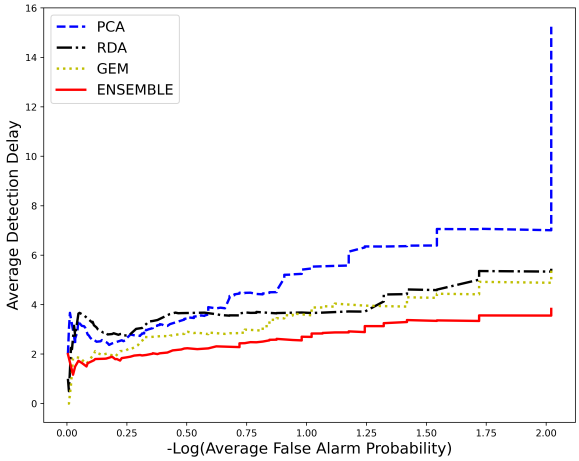
(a) FGSM on DQN-based single intersection DRL-TSC



(b) JSMA on DQN-based single intersection DRL-TSC



(c) FGSM on MA2C-based multiple intersection DRL-TSC



(d) JSMA on MA2C-based multiple intersection DRL-TSC

Fig. 6: Comparison of sequential detection performances in terms of average detection delay vs false alarm period.

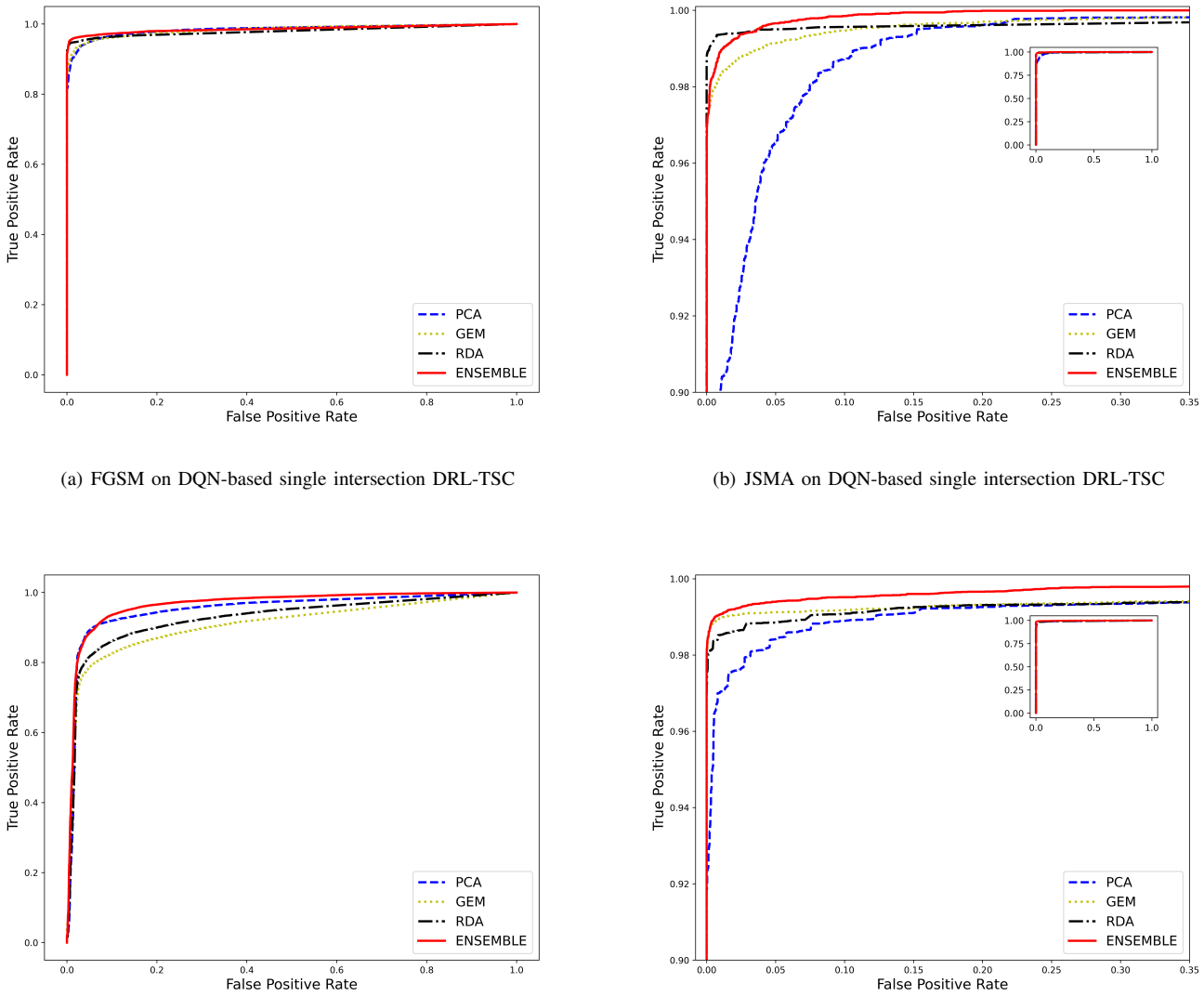
vector which are the number of vehicle and average speed. This form of state known as feature-based vectorized state representation (Se Fig. 5). The number of vehicle and average speed per lane states are concatenated to form final state representation. For example, the state definition for our single intersection DQN-TSC , which has 4 incoming roads with a 4 lane single intersection, is 32 units column vector.

Regarding the single-intersection DQN, sequential detectors are trained with 1 episode of anomaly-free traffic flow. Then, the detectors are trained on FGSM and JSMA adversarial attacks using 50 test episodes where adversarial attack starts after 200 state samples. Regarding the multi-intersection MA2C, we followed similar data collection procedure with slight changes. In our MA2C model, every intersection has a different number of approaching lanes, therefore, the state dimensions varies in MA2C model. We have 3 groups of state representation for 4 intersections as 82, 86, 92. After collecting neighborhood information, two of four intersections have the same size of state dimensions. Due to having different state dimensions, each agent of MA2C model is trained and tested separately, then, test results are concatenated. Adversarial attack for 1 episode is highly time consuming. Hence, the number of test samples are relatively low which is 35 MA2C episodes. In total, we have 105 test trials for MA2C-TSC model.

B. Results

Fig. 6 shows average detection delay vs false alarm probability results for the proposed ensemble model compared with the other statistical anomaly detectors. We observe that the proposed ensemble model has the lowest detection delay vs lower false alarm probability on FGSM attack to the single

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems



(a) FGSM on DQN-based single intersection DRL-TSC

(b) JSMA on DQN-based single intersection DRL-TSC

(c) FGSM on MA2C-based multiple intersection DRL-TSC

(d) JSMA on MA2C-based multiple intersection DRL-TSC

Fig. 7: ROC curves for different attacks and TSC settings with the proposed anomaly detection model.

intersection DQN model (Fig 6(a)) and JSMA attack to multi-intersection MA2C model (Fig 6(d)). The ensemble model also performs closer to the other statistical detectors for JSMA to single intersection DQN (Fig 6(b)) and FGSM to multi-intersection MA2C (Fig. 6(c)). Due to invisible nature of FGSM attack, all detectors have higher detection delays. The proposed ensemble model is the second best detector among all. Except for FGSM attacks on MA2C, the ensemble model detects the adversarial samples within less than 10 samples. This means that the ensemble detector informs the DRL agent within 10 adversarial samples, which is small enough for taking an action against adversarial attack. The ensemble model is able to handle multiple adversarial attack types on different controller settings. The results can be extended to a broader range of adversarial attacks that may target the DRL-TSCs. One proposed mitigation strategy on top of detecting the anomalies is switching to another TSC model such as max-pressure TSC after attacks are detected.

Next, we analyzed the overall detection performance with ROC curve and AUC scores. Since anomaly detectors are simple binary classifiers, evaluating the accuracy of anomaly detectors with the ROC classifier curve is important where the curve does not assume any distribution on data for producing classification performance. As depicted on Fig 7 and supported by the AUC scores in Table I, the proposed ensemble model outperforms all the other statistical detectors. While the statistics in bold shows the best detection performance, statistics in green tells the second best detection performance in Table I. It is clear from the statistics in green that different statistical anomaly detectors performs differently on different threat models, however, the proposed ensemble model has a clear advantage over the other detectors with almost perfect

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2021.3118972, IEEE Open Journal of Intelligent Transportation Systems

detection performance.

TABLE I: AUC scores for different baselines for all configurations

TSC-Attack	PCA	RDA	GEM	Ensemble
DQN-FGSM	0.9844	0.9749	0.9839	**0.9895**
DQN-Jacob	0.9950	0.9980	0.9979	**0.9994**
MA2C-FGSM	0.9549	0.9258	0.9028	**0.9637**
MA2C-Jacob	0.9942	0.9951	0.9954	**0.9978**

VIII. Conclusions

We have demonstrated the impact of adversarial attacks on DRL-based TSCs for a single-intersection and multiple intersection cases using different threat models. First, we evaluated the adverse impact of two adversarial attack models: FGSM and JSMA using white-box, and practical black-box settings. The results show that the performance of a DRL agent decreases sharply after the attack starts in all attack models and total waiting time increases become worse than the standard TSC methods. While, white-box FGSM and JSMA attacks affects the learning performance with similar impact. black-box FGSM attacks has severer impact compared to black-box JSMA attacks. Second, we presented a non-parametric online anomaly detection model which detects different anomalies sequentially by combining three existing anomaly detection models with a CUSUM-like algorithm. Through realistic SUMO traffic simulations, we evaluate the online detection performance of various anomaly detection approaches in the presence of adversarial attacks. The results show that the proposed ensemble model achieves superior performance in detecting anomalies in all threat models compared to other existing anomaly detectors.

The proposed study provides a security mechanism for known attack models. However, there are still some limitations that need to be addressed with further studies. While there are many different attack models, the vulnerability of DRL-TSCs should be evaluated with more threat models. This paper provides a novel anomaly detection model for DRL-TSCs but practical mitigation strategies and internal system robustness mechanisms should also be investigated. For future work, we plan to investigate other types of adversarial attacks and provide an integration mechanism with the proposed anomaly detection model and internal robustness mechanisms.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[2] A. Haydari and Y. Yilmaz, "Deep reinforcement learning for intelligent transportation systems: A survey," *arXiv preprint arXiv:2005.00935*, 2020.

[3] T. Chen, J. Liu, Y. Xiang, W. Niu, E. Tong, and Z. Han, "Adversarial attack and defense in reinforcement learning-from ai security view," *Cybersecurity*, vol. 2, no. 1, p. 11, 2019.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[5] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.

[6] S. E. Huang, W. Wong, Y. Feng, Q. A. Chen, Z. M. Mao, and H. X. Liu, "Impact evaluation of falsified data attacks on connected vehicle based traffic signal control," *arXiv preprint arXiv:2010.04753*, 2020.

[7] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[8] H. Wei, G. Zheng, V. V. Gayah, and Z. Li, "A survey on traffic signal control methods," *CoRR*, vol. abs/1904.08117, 2019. [Online]. Available: http://arxiv.org/abs/1904.08117

[9] X. Wang, J. Li, X. Kuang, Y.-a. Tan, and J. Li, "The security of machine learning in an adversarial setting: A survey," *Journal of Parallel and Distributed Computing*, vol. 130, pp. 12–23, 2019.

[10] K. Pandit, D. Ghosal, H. M. Zhang, and C.-N. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459–1471, 2013.

[11] Z. Li, D. Jin, C. Hannon, M. Shahidehpour, and J. Wang, "Assessing and mitigating cybersecurity risks of traffic light systems in smart cities," *IET Cyber-Physical Systems: Theory & Applications*, vol. 1, no. 1, pp. 60–69, 2016.

[12] C.-C. Yen, D. Ghosal, M. Zhang, C.-N. Chuah, and H. Chen, "Falsified data attack on backpressure-based traffic signal control algorithms," in *2018 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2018, pp. 1–8.

[13] C.-C. Yen, D. Ghosal, M. Zhang, and C.-N. Chuah, "Security vulnerabilities and protection algorithms for backpressure-based traffic signal control at an isolated intersection," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[14] A. Qayyum, M. Usama, J. Qadir, and A. I. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020.

[15] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2267–2281.

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[17] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 ieee symposium on security and privacy (sp)*. IEEE, 2017, pp. 39–57.

[18] J. Kos and D. Song, "Delving into adversarial attacks on deep policies," *arXiv preprint arXiv:1705.06452*, 2017.

[19] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 2017, pp. 506–519.

[20] V. Behzadan and A. Munir, "Vulnerability of deep reinforcement learning to policy induction attacks," in *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2017, pp. 262–275.

[21] S. H. Huang, N. Papernot, I. J. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017. [Online]. Available: https://arXiv:1802.06430

[22] E. Tretschk, S. J. Oh, and M. Fritz, "Sequential attacks on agents for long-term adversarial goals," *arXiv preprint arXiv:1805.12487*, 2018.

[23] S. Baluja and I. Fischer, "Learning to attack: Adversarial transformation networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[24] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," *arXiv preprint arXiv:1703.06748*, 2017.

[25] V. Behzadan and W. Hsu, "Adversarial exploitation of policy imitation," *arXiv preprint arXiv:1906.01121*, 2019.

[26] A. Gleave, M. Dennis, C. Wild, N. Kant, S. Levine, and S. Russell, "Adversarial policies: Attacking deep reinforcement learning," *arXiv preprint arXiv:1905.10615*, 2019.

[27] T. Chen, W. Niu, Y. Xiang, X. Bai, J. Liu, Z. Han, and G. Li, "Gradient band-based adversarial training for generalized attack immunity of a3c path finding," *arXiv preprint arXiv:1807.06752*, 2018.

[28] A. Pattanaik, Z. Tang, S. Liu, G. Bommannan, and G. Chowdhary, "Robust deep reinforcement learning with adversarial attacks," *arXiv preprint arXiv:1712.03632*, 2017.

[29] Y. Han, B. I. Rubinstein, T. Abraham, T. Alpcan, O. De Vel, S. Erfani, D. Hubczenko, C. Leckie, and P. Montague, "Reinforcement learning for autonomous defence in software-defined networking," in *International Conference on Decision and Game Theory for Security*. Springer, 2018, pp. 145–165.

[30] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, pp. 582–597.

[31] Y.-C. Lin, M.-Y. Liu, M. Sun, and J.-B. Huang, "Detecting adversarial attacks on neural network policies with visual foresight," *arXiv preprint arXiv:1710.00814*, 2017.

[32] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, 2017, pp. 135–147.

[33] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 665–674.

[34] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (statistical) detection of adversarial examples," *arXiv preprint arXiv:1702.06280*, 2017.

[35] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," *arXiv preprint arXiv:1703.00410*, 2017.

[36] A. N. Bhagoji, D. Cullina, C. Sitawarin, and P. Mittal, "Enhancing robustness of machine learning systems via data transformations," in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2018, pp. 1–5.

[37] A. J. Havens, Z. Jiang, and S. Sarkar, "Online robust policy learning in the presence of unknown adversaries," *arXiv preprint arXiv:1807.06064*, 2018.

[38] V. Gallego, R. Naveiro, and D. R. Insua, "Reinforcement learning under threats," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 9939–9940.

[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[40] A. O. Hero, "Geometric entropy minimization (gem) for anomaly detection and localization," in *Advances in Neural Information Processing Systems*, 2007, pp. 585–592.

[41] K. Sricharan and A. Hero, "Efficient anomaly detection using bipartite k-nn graphs," *Advances in Neural Information Processing Systems*, vol. 24, pp. 478–486, 2011.

[42] M. N. Kurt, Y. Yilmaz, and X. Wang, "Real-time nonparametric anomaly detection in high-dimensional settings," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[43] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.

[44] M. Basseville, I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. Prentice hall Englewood Cliffs, 1993, vol. 104.

[45] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. WieBner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[46] N. Papernot, F. Faghri, N. Carlini, I. Goodfellow, R. Feinman, A. Kurakin, C. Xie, Y. Sharma, T. Brown, A. Roy, A. Matyasko, V. Behzadan, K. Hambardzumyan, Z. Zhang, Y.-L. Juang, Z. Li, R. Sheatsley, A. Garg, J. Uesato, W. Gierke, Y. Dong, D. Berthelot, P. Hendricks, J. Rauber, and R. Long, "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.

[47] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[48] S. Thrun and A. Schwartz, "Issues in using function approximation for reinforcement learning," in *Proceedings of the Fourth Connectionist Models Summer School*. Hillsdale, NJ, 1993, pp. 255–263.

[49] H. Zhang, H. Chen, C. Xiao, B. Li, M. Liu, D. Boning, and C.-J. Hsieh, "Robust deep reinforcement learning against adversarial perturbations on state observations," *arXiv preprint arXiv:2003.08938*, 2020.

Ammar Haydari received the B.Sc. degree in Electronic Engineering from Uludag University, Bursa, Turkey, in 2014 and M.S. degree in Electrical Engineering from University of south Florida, Tampa, FL, in 2019. He is currently a Ph.D. candidate at the Department of Electrical and Computer Engineering at the University of California, Davis. His research interests include intelligent transportation systems, cyber-security, privacy and machine learning.

Micheal Zhang is currently a professor in the Civil and Environmental Engineering Department at University of California Davis. His research is in traffic operations and control, transportation network analysis and intelligent transportation systems. Professor Zhang received his BS degree in Civil Engineering from Tongji University, and MS and PhD degrees in Engineering from University of California Irvine. He is an Area Editor of the journal Network and Spatial Economics, and an Associate Editor of Transportation Research, Part B: Methodological.

Chen-Nee Chuah is currently the Child Family Professor in Engineering in the Electrical and Computer Engineering Department, University of California, Davis. She received her Ph.D. degrees in Electrical Engineering and Computer Sciences from the University of California, Berkeley. Her research interests include Internet measurements, cybersecurity, and applying data science and intelligent learning techniques to societal-scale networked systems and applications, including smart health and intelligent transportation systems. She is a Fellow of the IEEE and an ACM Distinguished Scientist.