ADS-Lead: Lifelong Anomaly Detection in Autonomous Driving Systems

Xingshuo Han, Yuan Zhou, *Member, IEEE*, Kangjie Chen, Han Qiu, Meikang Qiu, *Senior Member, IEEE*, Yang Liu, *Senior Member, IEEE*, and Tianwei Zhang, *Member, IEEE*

Abstract—Autonomous Vehicles (AVs) are closely connected in the Cooperative Intelligent Transportation System (C-ITS). They are equipped with various sensors and controlled by Autonomous Driving Systems (ADSs) to provide high-level autonomy. The vehicles exchange different types of real-time data with each other, which can help reduce traffic accidents and congestion, and improve the efficiency of transportation systems. However, when interacting with the environment, AVs suffer from a broad attack surface, and the sensory data are susceptible to anomalies caused by faults, sensor malfunctions, or attacks, which may jeopardize traffic safety and result in serious accidents. In this paper, we propose `ADS-Lead`, an efficient collaborative anomaly detection methodology to protect the lane-following mechanism of ADSs. `ADS-Lead` is equipped with a novel transformer-based one-class classification model to identify time series anomalies (GPS spoofing threat) and adversarial image examples (traffic sign and lane recognition attacks). Besides, AVs inside the C-ITS form a cognitive network, enabling us to apply the federated learning technology to our anomaly detection method, where the vehicles in the C-ITS jointly update the detection model with higher model generalization and data privacy. Experiments on Baidu *Apollo* and two public data sets (GTSRB and Tumsimple) indicate that our method can not only detect sensor anomalies effectively and efficiently but also outperform state-of-the-art anomaly detection methods.

Index Terms—Federated Learning, Autonomous Driving Systems, Intelligent Transportation System (ITS), Cognitive Networking

1 Introduction

Over the past years, Autonomous Vehicles (AVs) are experiencing rapid development. Benefiting from the advances in the technologies of computing, mechanics and deep learning [1], modern vehicles become more automated and intelligent. Many IT and motor companies are attracted to devote themselves to this promising domain e.g., Baidu *Apollo*[1], Google *Waymo*[2]. Hence, in the near future, we expect to see various types of AVs will be fully commercialized to significantly impact different aspects of our life.

The essential component of an AV is the Autonomous Driving System (ADS). It receives information from the external environment and then makes driving decisions. A standard ADS has a pipeline consisting of multiple modules for different functionalities, e.g., perception, planning, control. They cooperate to achieve end-to-end automation. Unfortunately, the high complexity of the ADS inevitably brings a broad attack surface. For example, an adversary can launch GPS spoofing attacks to mislead AVs to navigate to a dangerous position [2]. The attack cost is only $200 for a low-end "GPS spoofing" device. By adding malicious patches [3], [4] on the road or traffic signs, an adversary can make ADSs perceive the environment mistakenly and make wrong decisions. Attacks on Lidar can make ADSs ignore the surrounding obstacles, resulting in collisions [5].

It is important to guarantee the robustness of the ADS against those cyber attacks and faults. A practical solution is anomaly detection, which monitors the runtime behaviors and states of the ADS, as well as the received environmental information, to identify any suspicious events. The emergence of the Cooperative Intelligent Transport System (C-ITS) provides new opportunities for reliable and effective anomaly detection. In a C-ITS, vehicles are connected with each other, the infrastructures, passengers, and cloud. They naturally form a cognitive network, and frequently exchange runtime data for better traffic and mobility management [6]–[8]. As a result, it is also promising that vehicles in the C-ITS can perform anomaly detection collaboratively to mitigate any attacks against the ADS. This can increase the detection efficiency and accuracy.

Motivated by this feature, this paper proposes `ADS-Lead`, a novel methodology for protecting Autonomous Driving Systems with Lifelong anomaly detection. We consider the lane following mechanism, which is the most common and fundamental scenario in not only ADSs but also state-of-the-art Advanced Driver-Assistance Systems (ADASs) and Lane Keeping Assist Systems (LKASs). Different types of security threats have been disclosed in the lane following scenario, i.e., localization attacks, lane detection attacks, and traffic sign recognition attacks. They can lead to severe consequences and damages, such as car crashes, human injuries or even deaths. Hence it is important for vehicles to be immune to them for secure and safe driving. Although prior works

- *X. Han, Y. Zhou, K. Chen, Y. Liu and T. Zhang are with the School of Computer Science and Engineering, Nanyang Technological University, 639798, Singapore. Email: xingshuo001@e.ntu.edu.sg, y.zhou@ntu.edu.sg, kangjie001@e.ntu.edu.sg, yangliu@ntu.edu.sg, tianwei.zhang@ntu.edu.sg*
- *H. Qiu (corresponding author) is with Institute for Network Sciences and Cyberspace, BNRist, Tsinghua University, Beijing 100084, China. Email: qiuhan@tsinghua.edu.cn*
- *M. Qiu is with Texas A&M University Commerce, TX, 75428, USA. Email: meikang.qiu@tamuc.edu*

1. https://github.com/lgsvl/apollo-5.0
2. https://waymo.com/

proposed some solutions to defeat sensor attacks for AVs [5], [9], [10], they only focus on one specific kind of threats. It is challenging to design a unified and comprehensive method to cover different attack vectors, as they have distinct behaviors and techniques.

`ADS-Lead` introduces two contributions to achieve efficient and unified protection. The first one is a novel one-class classification model, dubbed T-GP (Transformer with Gradient Penalty). It is capable of analyzing and identifying time series anomalies (localization attacks) and adversarial images (i.e., lane detection attacks and traffic sign recognition attacks) in the lane following scenario. This model needs to be trained offline only from normal data, and then deployed in the ADS as an online detector to inspect different sources of sensory data, and discover the suspicious input. T-GP is built from an one-layer transformer encoder. It introduces a novel loss function, which combines the Negative Log Likelihood (NLL) with the Gradient Penalty (GP). The integration of these techniques gives very high accuracy for anomaly detection of various attacks.

The second contribution is the adoption of federated learning and lifelong learning for anomaly detection in the C-ITS. Each vehicle in our system not only performs the online monitoring and detection, but also continuously collects live data to update the one-class model. They train the model locally, and then send the model gradient to a parameter server hosted in the cloud. This parameter server aggregates the gradients from different vehicles at different zones of the C-ITS, and releases the final model back to them for update. The collaboration for anomaly detection based on federated learning can significantly improve the model generalization and performance while preserving the vehicle's privacy.

We implement a prototype of our methodology in a federated learning system. We apply our proposed model on the datasets from the real world, and collected from simulations to comprehensively evaluate its effectiveness. Specifically, for localization attacks, since there are no public datasets available, we collect the Inertial Measurement Unit (IMU) data from Baidu *Apollo*, running on the San Francisco map with the *LGSVL* simulator[3]. We follow [2] to implement GPS attacks, which can cause severe fluctuation of the IMU data generated by the Multi-Sensor Fusion (MSF) component in *Apollo*. For lane attacks, we adopt the Tumsimple datatset, and implement the attack method in [11] to generate fixed and variable adversarial patches. For traffic sign attacks, we use the GTSRB dataset. We reproduce the boundary attacks [12] and poster attacks [4] to generate adversarial data. We compare T-GP with existing one-class classification methods. Evaluation results show that T-GP outperforms other methods in detection of these attacks.

In summary, the main contributions of our work are:

- We propose `ADS-Lead`, a novel collaborative anomaly detection approach to protect the lane following scenario of the ADS efficiently and comprehensively.
- We introduce T-GP, a novel one-class classification model based on the transformer for anomaly detection. It can effectively detect both time series anomalies and adversarial images.
- We are the first to adopt federated learning and lifelong learning to realize collaborative anomaly detection on AVs, which can enhance the model generalization and performance without compromising vehicles' privacy.
- We conduct extensive evaluations of our method over simulation and real-world datasets. We demonstrate T-GP outperforms existing state-of-the-art models on the detection of localization, traffic sign and lane recognition attacks. And `ADS-Lead` with T-GP can be made effectiveness and practical for AVs anomaly detection.

The rest of the paper is organized as follows. Section 2 gives the literature review of the detection of GPS spoofing attacks and adversarial examples. Section 3 illustrates the preliminaries and the problem to be addressed in this work. Section 4 describes the system overview and details of the detection model. Section 5 presents our solutions for model evolution. Section 6 conducts comprehensive experiments to evaluate the effectiveness and efficiency of the proposed method. Conclusion is given in Section 7.

2 Related Works

2.1 Detection of GPS Spoofing Attacks against AVs.

Although prior works made some attempts to detect GPS attacks against AVs [13]–[17], how to effectively mitigate such threat is still a long-standing problem. The MSF algorithms were regarded as the most effective defense method in ADSs [18]. Unfortunately, Shen *et al.* [2] found a vulnerability in the design of MSF-based localization and successfully implemented a sophisticated attack to invalidate the protection. Researchers also studied spoofing detection by cross-checking GPS readings and IMUs data [19]. However, IMU data suffer from the accumulation of drift errors such that they provide reliable protection against spoofing attacks if an adversary causes gradual deviation of the victim vehicles from their actual positions [20]. Compared with these prior works, we only use the instantaneous changes of the IMU data to detect whether the vehicle is being attacked, which achieves very high detection accuracy.

2.2 Detection of Adversarial Examples.

Some works introduced methods to detect adversarial examples, especially in the CV domain. Qiu *et al.* [21] illustrated adversarial attacks against network intrusion detection in IoT systems. Xu *et al.* [22] proposed a method called feature freezing to detect adversarial examples by reducing color bit depth and spatial smoothing. They set a threshold to judge whether the original input data is benign or malicious. Lee *et al.* [23] designed a method using Gaussian discriminant analysis to obtain the confidence score based on the Mahalanobis distance in the feature space of DNN models. Li *et al.* [24] proposed to detect localized adversarial examples by removing and analyzing critical regions controlled by the adversary. Meng and Chen [25] used detector networks to identify adversarial examples by approximating the manifold of normal examples. Feinman *et al.* [26] investigated the Bayesian uncertainty estimates in dropout neural networks, and conducted density estimation in the subspace of deep features to distinguish normal and adversarial examples. Ma *et al.* [27] used the estimation of

3. https://github.com/lgsvl/simulator

Local Intrinsic Dimensionality (LID) to quantify the distance between the target sample and normal samples. Katzir and Elovici [28] explored the sample behaviors in the activation space of different network layers for adversarial example detection. Li and Qiu *et al.* [29] proposed a novel method for intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning. Wang *et al.* [30] randomly mutated the model and perturbs the decision boundary, which can possibly alter the prediction of adversarial examples, while maintaining the prediction of normal samples. Tian *et al.* [31] utilized input transformations to process the input samples, to which the adversarial examples are very sensitive.

In the context of autonomous driving, some works designed solutions to detect adversarial images captured by the vehicles. Sun *et.al* [32] developed a supervised defense method based on adversarial training with a novel and stereo-based regularizer to enhance the 3D object detection model. Safavi *et al.* [33] adopted two distinct and efficient DNN architectures to detect, isolate and predict sensor faults. One-class models (e.g. Deep-SVDD [34], HRN [35]) were designed for anomaly detection of adversarial examples, and evaluated on the stop sign attacks. For lane attacks, Sato *et al.* [3] proposed an attack method based on image segmentation and deployed a bounded patch to simulate the road dirt to fool the lane detection algorithms. Following this work, Xu *et al.* [11] designed a CNN-based model with prior knowledge of abnormal data to achieve attack detection. However, these works need prior knowledge of the adversarial samples, or can only be applied to specific attacks but fail to be extended to others. In contrast, our solution proposed in this paper is unified to cover various types of attacks with different formats of sensory data in the lane following scenario.

3 Background and Problem Statement

3.1 Overview of ADSs

The main responsibility of an ADS is to recognize the surrounding environment and generate proper motion commands to the vehicle [36] [37]. Hence, a typical ADS usually consists of the following modules: localization, perception, planning and control. The localization module uses the information from different sensors (e.g., GPS, IMU, Lidar) to localize the AV on the map based on the Real Time Kinematic (RTK) method and Multi-Sensor Fusion (MSF) algorithms. The perception module is an AI-based subsystem, which receives input data of different formats (e.g., image, point cloud) from various sensors and leverages Deep Learning models to identify the surrounding traffic conditions (e.g., the status of traffic light, stop sign and speed limit sign) and obstacles (e.g., object types, the speeds of other vehicles on the road). The planning module performs offline path planning to generate a feasible path from the initial position to the destination based on the map information. It also conducts real-time trajectory planning, which utilizes the results from the localization module and perception module to generate a collision-free trajectory in a short time duration. The control module finally generates low-level commands, such as steering, throttle and brake, to the chassis to track the generated collision-free trajectory.

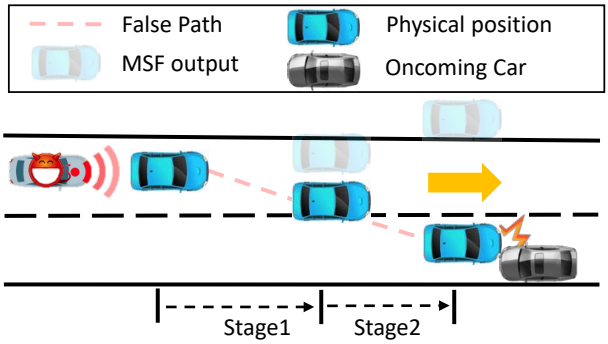


Fig. 1. Illustration of GPS-based localization attacks. Stage 1: Vulnerability profiling; Stage 2: Aggressive spoofing.

3.2 Security Threats in the Lane Following Scenario

Lane following is the most common scenario during the AV operations, where the vehicle moves along the central lines of lanes. In this scenario, the execution of an ADS highly depends on the accuracy of localization, lane boundary detection and traffic signs. Hence, the following three kinds of attacks were proposed to compromise the execution of ADSs in lane following.

Localization Attack. This attack uses counterfeit GPS signals to inference with the legitimate ones. Then the ADS cannot localize the AV correctly, resulting in positioning errors. Consequently, the ADS will mislead the vehicle to deviate from the expected lane and even cause serious accidents. Although the MSF algorithms in ADSs are designed to mitigate GSP spoofing, researchers find that they are still vulnerable to the take-over attack [2] where the spoofed GPS signals can dominate the inputs of the MSF process and fool MSF to ignore other inputs. Figure 1 illustrates the mechanism of such an attack. A victim vehicle (blue) is moving along the straight lane. The attacker vehicle, following the victim, launches a two-stage GPS spoofing attack. The first stage is vulnerability profiling: the attacker collects and analyzes the behaviors of the victim vehicle and determines the time duration to perform GPS attacks. The second stage is aggressive spoofing: the attacker sends wrong GPS signals to the victim vehicle, whose MSF algorithms compute wrong localization of the AV (the shaded blue one). To make the vehicle stay in the center of the lane, the ADS asks the vehicles to move right, which actually makes it cross the lane and collide with the oncoming vehicle.

Lane Detection Attack. In lane following, an ADS should also need to detect the boundaries of a lane to localize the central line of the lane. Currently, DNNs are the most popular method for lane detection in ADSs. Hence, due to the inherent vulnerability of DNNs, the adversary can also fool the DNN model to cause wrong recognition of lane boundaries, resulting in wrong motion controls to drive along the center of the lane. For example, the adversary can add visual perturbations on the real-world road to make the vehicle deviate the central line and hit a surrounding object [3]. Figure 2 shows an attack example [11]: the adversary carefully identifies the optimal location for the patch and then manipulates the subset of pixels of the input images to achieve the goal, i.e., making the lane detection system recognize a wrong lane boundary around the patch.

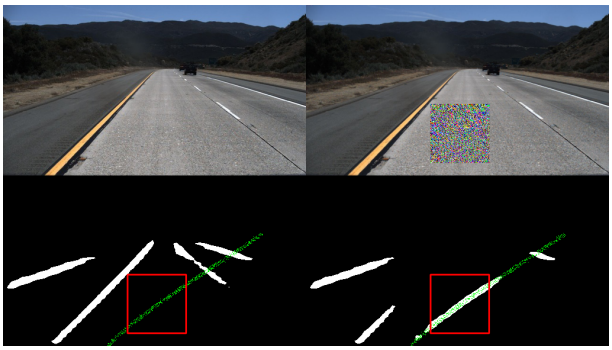


Fig. 2. Lane detection attack. First row: the original input image (left) and the adversarial image with a fixed patch. Second row: the corresponding lane segmentation results from the ADS. Red boxes show the patch localization; induced lanes are marked with green.

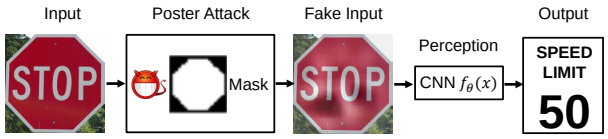


Fig. 3. Poster attacks on the traffic sign.

Traffic Sign Recognition Attack. Recognition of traffic signs can also affect the lane following as an AV must obey the traffic rules described by those signs. Since the ADS leverages CNN models to detect and classify traffic signs, an adversary can leverage the adversarial attack techniques to compromise the model so the ADS will miss or misclassify the traffic signs, and generate wrong motion decisions. Figure 3 shows a poster attack on a stop sign [4]. The adversary adopts the Robust Physical Perturbations (RP_2) algorithm [4] to generate visual adversarial perturbations and attach them to the stop sign. Then the perception module in the ADS will identify it as a speed limit sign. Alternatively, the adversary can also adopt generative adversarial networks to craft malicious patches to compromise the traffic sign recognition model [38].

3.3 Problem Statement

We aim to address the following problem: *How to develop and design a unified and efficient method to detect anomalies of the ADS caused by different kinds of attacks at real time?* We want to have an attack-agnostic approach, i.e., the detector is built from normal data and conditions, but general and effective for various known and unknown threats.

Without loss of generality, we consider the following six attacks from three categories when designing our approach. They represent state-of-the-art security threats against the lane-following mechanism in modern ADSs.

Localization Attacks. We consider the GPS spoofing attack [2] in our method design and evaluation. We assume a malicious vehicle follows the victim AV and interfere with its GPS signals. The faked signals can fool the MSF algorithms based on the take-over vulnerability. We further assume there are no other obstacles on the road, so the motion change of the victim AV only depends on the localization. We focus on two specific attack goals: (1) an off-road attack tries to lead the victim to hit the curb; (2) a wrong-way attack tries to deviate the victim AV to the opposite pavement.

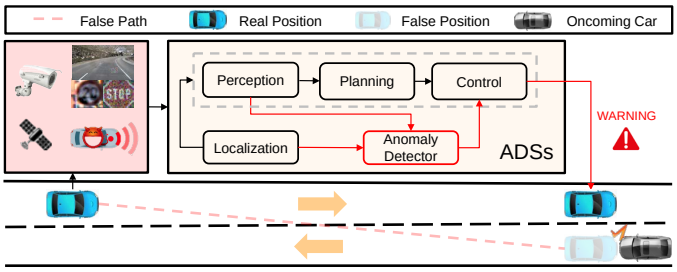


Fig. 4. Overview of our anomaly detection methodology.

Traffic Sign Recognition Attacks. We consider two types of attacks in this category: (3) a boundary attack is a decision-based adversarial attack [12]. The adversary does not need any information about the target model in the ADS. He generates the adversarial perturbations on the traffic sign only from the prediction results of the model corresponding to given input images. (4) In a poster attack, the adversary generates malicious posters for traffic signs using a novel Robust Physical Perturbations algorithm [4]. In these two attacks, the adversary is able to physically alter the traffic signs (e.g., adding posters or patches) without changing their visual semantics.

Lane Detection Attacks. We assume the adversary is able to add carefully-crafted patches on the road to deceive the lane detection model in the target ADS. We adopt the Projected Gradient Descent to generate two types of adversarial patches [11]: (5) a fixed-size patch with the size of 100×100 is injected to the images of 512×288; (6) a varied-size patch has the size scaled based on the distance from the camera to the destination lane segments.

4 ADS-Lead

In this section, we describe `ADS-Lead`, our anomaly detection system for the lane-following scenario.

4.1 System Overview

Figure 4 shows the overview of our `ADS-Lead` system. The essential component is a powerful anomaly detector deployed in an ADS for attack detection. The workflow contains two stages, as described below.

The first stage is *offline training*. We train a one-class model to describe the normal behaviors of the ADS. Since we aim to have an attack-agnostic approach, we cannot include any attack-specific data samples when training the detection model, which are hard to obtain and not general for other unknown attacks. Instead, we just collect normal data during the vehicle operations. Note that the normal data can be collected by making the AV run automatically without launching any attack. The collection is not related to any specific road or road condition. Then this model is able to predict whether the incoming data samples belong to the same distribution as the training data (labeled as benign), or deviate a lot from the normal ones (labeled as malicious).

Since our model is designed to be general for different attacks, it should be able to handle different formats of

sensory data in the ADS. Specifically, we consider two types of sensory data that are vulnerable to be manipulated by the adversary to compromise the vehicle operation. The first one is IMU messages, which are time series data. The second one is images captured from the cameras. These are used to mitigate attacks against the lane detection and traffic sign recognition. We introduce approaches to preprocess those types of data before feeding them into the model for training and inference.

The second stage is *online prediction*. The model is implemented as a module in the ADS to monitor the outputs of the perception and localization modules during the AV operation. When the AV receives malicious sensory data crafted by the adversary (e.g., traffic sign with the adversarial patch, spoofed GPS signals), the anomaly detector is able to identify such suspicious events from these two monitor modules, and then send notifications to the control module. The control module will perform some mitigation actions, e.g., stopping the vehicle, warning and asking the driver in the vehicle to take control of it.

4.2 T-GP: One-class Model for Anomaly Detection

We design T-GP, a novel one-class classification model based on the transformer structure, for each vehicle to achieve anomaly detection. A transformer [39] is a deep neural network structure using the self-attention mechanism. It replaces the Recurrent Neural Network (RNN) structure with an encoder and decoder. It can significantly improve the model accuracy for Natural Language Processing (NLP) tasks. Besides, it is also highly interpretable and supports fully parallel computing. Recently, researchers extended the transformer structure to the domain of Computer Vision (CV) [40], which also demonstrates remarkable performance for image classification.

Inspired by the successful applications of the transformer in the NLP and CV domains, we aim to apply it to build a one-class model for anomaly detection. Figure 5 shows the network structure of our proposed model, T-GP. It adopts an input embedding component (i.e., Input Embedding Matrix) to map each original input data into a vector with a fixed length and an encoder (i.e., Transformer Encoder) as the feature extractor to learn the hidden patterns of normal data and detect abnormal data (i.e., malicious sensory input in ADSs).

Specifically, the input $X = (x_1^T, ..., x_t^T)^T \in \mathbb{R}^{t \times P}$ of the model is a two-dimensional matrix, where t is the length of the input sequence, P is the dimension of each input data x_i, i.e., $x_i \in \mathbb{R}^{1 \times P}$, for $i = 1, 2, \ldots, t$, and $(\cdot)^T$ denotes the transpose operator. Note that our model is unified and can accept both the image data and IMU time series data. Each image is reshaped into a sequence of flattened 2D patches by dividing the original image into t patches [40]. For the IMU data, each single sample x_i is recorded at a time instant. Since the transformer encoder requires a constant latent vector size, each input sequence is first mapped to a fixed-length sequence of patch embeddings using a learnable embedding vector x_{class}, a trainable linear projection E, and a standard learnable 1D position embeddings $\mathbf{E}_{pos}$ [40], as given in Equation 1:

$$z_0 = (x_{class}^T, \mathbf{E}^T X^T)^T + \mathbf{E}_{pos} \quad (1)$$

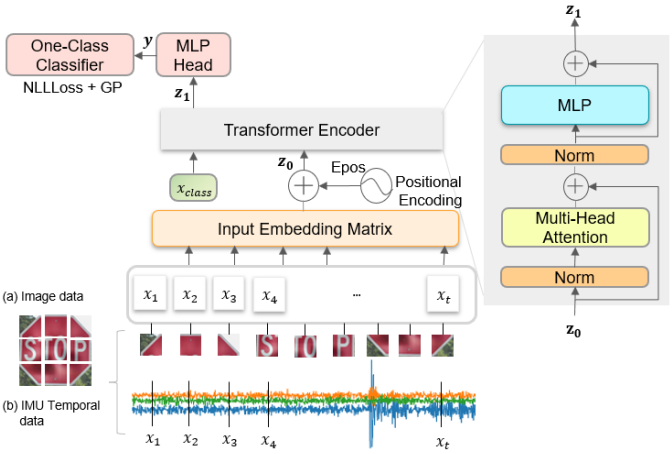


Fig. 5. T-GP model structure.

where $x_{class} \in \mathbb{R}^{1 \times D}$ and its output can be used for classification, $\mathbf{E} \in \mathbb{R}^{P \times D}$ is a fully connected layer, and $\mathbf{E}_{pos} \in \mathbb{R}^{(t+1) \times D}$ is introduced to add the positional information of the input sequence to the patch embeddings.

The patch embeddings z_0 are sent to the transformer encoder, which is used to extract the feature representation of the input data and consists of a Multi-headed Self-Attention (MSA) network and a two-layer Perceptron (MLP) with GELU. Note that the inputs of MSA and MLP are first normalized via layer normalization (LN) [41]. Hence, the operation of the transformer encoder can be formulated as:

$$z_1' = MSA(LN(z_0)) + z_0 \quad (2)$$

$$z_1 = MLP(LN(z_1')) + z_1' \quad (3)$$

We design a novel loss function in T-GP to achieve one-class classification. Negative Log Likelihood Loss (NLLLoss) is widely used in multi-class classification tasks. However, in our one-class model, the output has only one class, so we use the sigmoid function in NLLloss to calculate the probability that an input x belongs to the class. It generally requires regularizaion due to the sigmoid saturation and feature bias in NLLLoss [35]. It means that an unimportant feature with a larger value may have larger effects on the computation of the probability. Hence, inspired by [42], which adds 1-Lipschitz constraints to the discriminator of WGAN by gradient penalty (GP), we apply the gradient penalty in T-GP to mitigate such biases and obtain the following loss function:

$$\begin{aligned} loss = {} & E_{x \sim P_x}[-log(Sigmoid(f(x)))] \\ & + \lambda E_{x \sim P_x}[(\|\nabla_x f(x)\|_2 - 1)^2] \end{aligned} \quad (4)$$

The first term is NLLLoss and the second one is gradient penalty. P_x denotes the data distribution of the given positive class, and λ is a hyper-parameter to balance the penalty. $Sigmoid(f(x)) \in (0, 1)$ is the probability that x belongs to the positive class. The advantage of the gradient penalty will be demonstrated in our evaluations by comparing with the H-regularization [35].

5 Model Evolution

It is possible that the AV behaviors can drift over a long period of time, possibly caused by the varied environment and

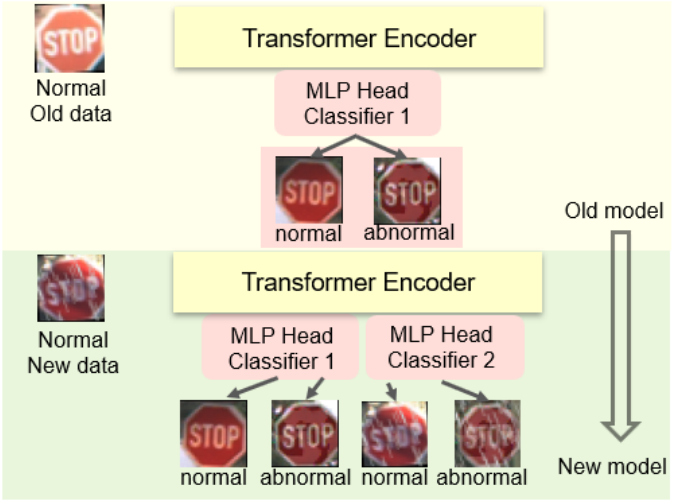


Fig. 6. Lifelong learning for one-class model update.

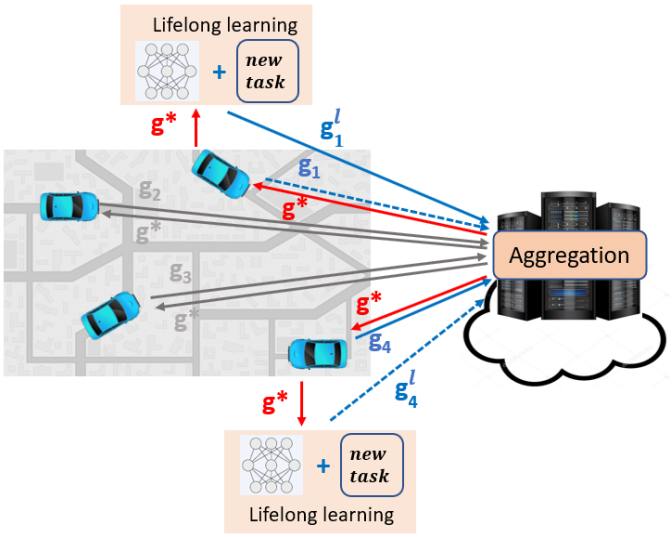


Fig. 7. Model training and update with federated learning in ADS-Lead.

road conditions. Hence, frequent model update is necessary to maintain the high anomaly detection performance. The vehicle can periodically collect the runtime data when it is at the normal state. Then it fine-tunes the detection model based on such data. This process is lightweight compared with model training from scratch, and thus computationally feasible using the on-board computer. We further leverage two technologies to enhance the efficiency of model update.

5.1 Lifelong Learning

ADS-Lead applies lifelong learning for model evolution with runtime data. Lifelong learning is defined as an adaptive machine learning algorithm, which is able to progressively learn from a continuous stream of information over a long time span [43]. A good lifelong learning algorithm can produce a machine learning model with great accommodation of the new information. lifelong learning has become more important in autonomous agents and systems, which need to interact with the dynamic real world.

A variety of strategies have been designed to achieve lifelong learning. In ADS-Lead, we use the method proposed in [35], which is a continuous learning process and shows better performance than other strategies. Its idea is to train a new sub-classifier for each new task, and the prediction is done by selecting the result of one of the existing sub-classifiers. The detailed process of lifelong learning in ADS-Lead is shown in Figure 6. As described in Section 4.2, the T-GP model consists of two main parts: a transformer encoder, which extracts feature vectors of different formats of sensor data, and an MLP head, which classifies the extracted features to make decisions. During lifelong learning, the transformer encoder is fixed. Every time a new dataset is collected, ADS-Lead trains a new MLP head to memorize the new data distributions. Then new MLP heads will be integrated with the old ones in the updated model for decision making. Note that to guarantee the scale of the model, we may set the maximal number of MLP head classifiers and forget the very old classifiers. In this way, we can guarantee to not only remember the historic knowledge but also learn new information from the newly acquired data.

5.2 Model Update with Federated Learning

Since there may be multiple AVs on the roads, and different vehicles have different private data, they can collaborate to train and update a more robust model. Hence, we further propose to use federated learning [44], [45] to optimize the model training process, which enables different vehicles to collaborate on the model training without releasing their data. Hence, the data privacy of the vehicles (e.g., location) is preserved compared to the case where the data are offloaded to the cloud for model training. Figure 7 shows the process of the detection model update with federated learning. A centralized Parameter Server (PS) is introduced in the remote cloud. Each AV in the C-ITS is able to talk with the PS via the V2C communication technology [46]. During the training process, each vehicle trains the model gradient g_i from its local collected data, and uploads the results to the PS. Then the PS will receive multiple gradients from different vehicles. It aggregates them into one gradient vector g^* by calculating the average value, and releases the new model to each vehicle in the C-ITS. So each vehicle can use the latest model for online anomaly detection with better generalization and performance.

It is worth noting that we adopt the asynchronous training instead of synchronous training. The PS does not need to wait for the gradients from all the vehicles in the network, since some vehicles may not participate in the model update process. It performs the gradient aggregation and model release at a fixed frequency, to guarantee the model update service is always available. It is possible that some vehicles are malicious or compromised, trying to send the PS wrong updates to compromise the detection model. We can adopt some sophisticated aggregation rules [47] to filter out such malicious gradients. Besides, we can also follow the works [48] to further mitigate the indirect leakage from the gradients. Implementation of these advanced solutions into ADS-Lead is our future work.

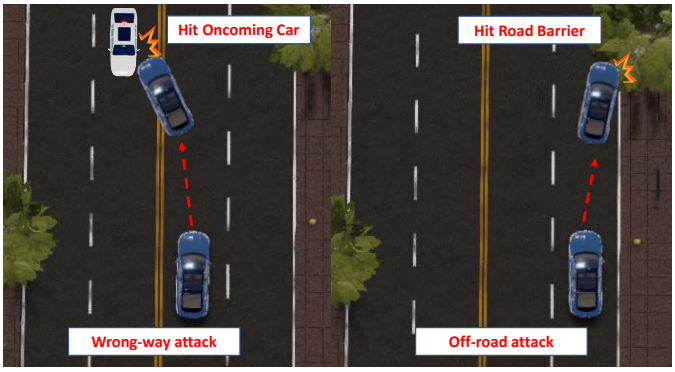


Fig. 8. GPS spoofing attacks in LGSVL simulator.

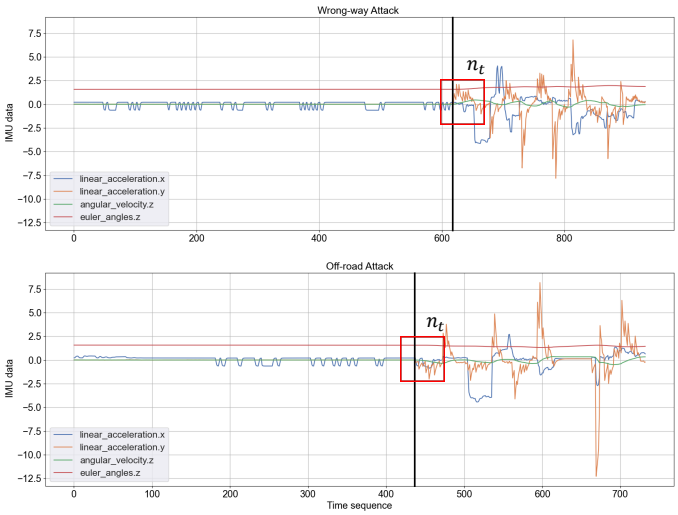


Fig. 9. Data sequences of ax, ay, avz, and γ when the AV is under the off-road and wrong-way attacks, respectively. The black line represents the moment the spoofing attack starts. The red box is the sliding window with the length of $n = 10$ data samples. n_t represents that the attack is detected after n_{th} samples of the attack occurrence.

6 Evaluations

In this section, we evaluate the effectiveness and robustness of the proposed `ADS-Lead` system and T-GP model against the three kinds of attacks described in Section 3.3.

6.1 Evaluation of T-GP

We first evaluate the performance of T-GP on GPS attacks, traffic sign attacks, and lane detection attacks.

6.1.1 Defeating Localization Attacks

Data Sets. Since there are no public datasets for GPS spoofing attacks, we deploy the attacks in Baidu *Apollo* 5.0 running with the *LGSVL* simulator on the San Francisco map, and collect data for normal and malicious cases. We consider the attack scenario where an adversarial vehicle tailgates the victim AV while launching GPS spoofing. Following the attack settings in [2], we consider two concrete adversarial goals as shown in Figure 8: off-road attack aims to deviate the AV to hit the curb; wrong-way attack aims to deviate the AV to the opposite lane and hit the oncoming vehicle.

GPS spoofing will cause a sudden change of the AV's localization computation, resulting in the change of AV's motion. Hence, we monitor the IMU messages, whose channel name is */apollo/sensor/gnss/corrected_imu* in the *Apollo* ADS. There are three kinds of motion data in the IMU messages and each one is a 3D vector: linear acceleration (ax, ay, az), augular velocity (avx, avy, avz), and Euler angles (α, β, γ). Since the current HD map for *Apollo* does not contain the altitude information, only the linear accelerations ax and ay, angular velocity avz, and Euler angle γ are affected by the motion of the AV. Moreover, based on our observation of the real-time IMU data, these four values exhibit distinct behaviors when the AV deviates from the predetermined path, compared to the scenarios of lane change or turn. Hence, at each time instant, we collect these four types of data as the model features. Figure 9 shows two data sequences of the four selected data types during the AV motion under GPS spoofing attacks, where the message sampling frequency is around 85 FPS (Frame-Per-Second) in our experiments.

Since our task is one-class anomaly detection, only benign data are available for model training. The road in the map of LGSVL simulator is flat and we set random NPCs (vehicles and passengers) in the map. We randomly set the destination for the vehicle and collect the four types of IMU data from *Apollo* when the vehicle is in normal and secure states. A total of 32,115 raw data samples are generated for model training. The testing set should contain both the normal and attack samples. We run *Apollo* ten times under the two types of GPS spoofing attacks, and collect the related IMU data. We label the data before the attack occurrence as "normal". We also assign the "abnormal" label to the data collected in a short period right after the GPS spoofing is launched (around 20 new IMU messages). Table 1 summarizes the ten testing data sequences.

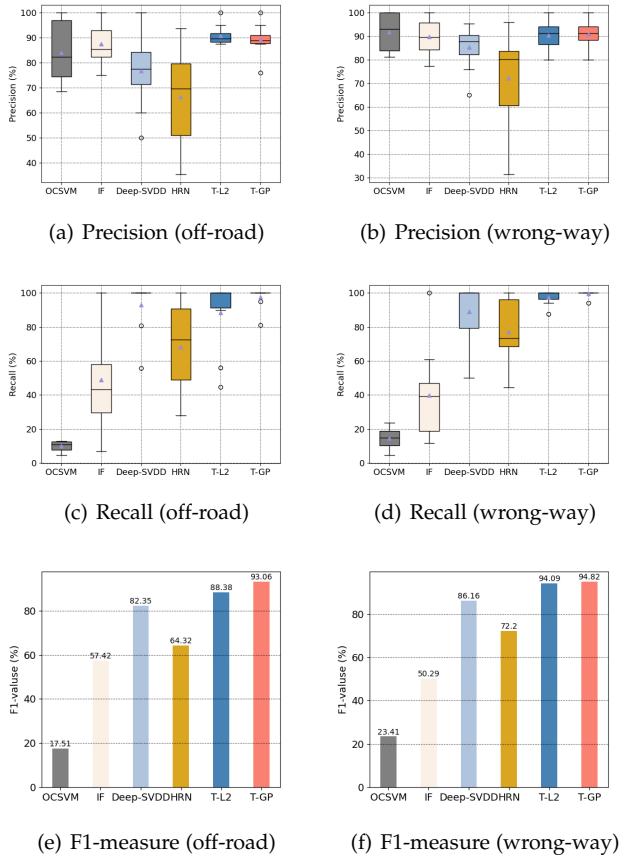
TABLE 1
Number of data samples in each testing sequence

Sequence		#0	#1	#2	#3	#4	#5	#6	#7	#8	#9
off-road attack	normal	420	423	237	294	571	494	210	461	363	535
	abnormal	20	17	23	16	19	16	20	19	17	25
wrong-way attack	normal	245	616	418	325	550	274	271	338	204	396
	abnormal	25	14	22	26	20	16	19	22	16	24

Once we obtain the training and testing data sequences, we generate the corresponding training and testing datasets by dividing each data sequence into a set of sub-sequences with the length of 10. We use the sliding window method with a stride of 1 to generate the sub-sequences. Hence, a sequence with n samples can generate $(n - 9)$ sub-sequences. Note that we employ the same data preprocessing method to all the models for fair comparison.

Model Configurations. According to the format of the generated data samples, the input dimension of T-GP is set as 10×4, i.e., each input sequence has 10 consecutive data samples and each sample is a 4D vector. In terms of the model hyper-parameters, we use an embedding dimension of 4 units, 4 transformer heads, and 128 units in the hidden layer of the output MLP head. We use the AdamW optimizer with a learning rate of 1e-4. λ is set as 0.1.

Baseline Methods We compare our T-GP model with the following baselines.



(a) Precision (off-road) (b) Precision (wrong-way)

(c) Recall (off-road) (d) Recall (wrong-way)

(e) F1-measure (off-road) (f) F1-measure (wrong-way)

Fig. 10. Results of Precision, Recall and F1-measure on the two GPS spoofing attack datasets.

- *OC-SVM* [49]: this is a traditional one-class classifier based on kernel SVM. In our implementation, the RBF kernel is applied and the hyper-parameter is selected from a set of discretized values in the interval $[0, 1]$.
- *iForest* [50]: this is another popular one-class classifier. It isolates anomaly points by building decision trees. We use the default values of the hyper-parameters.
- *Deep-SVDD* [34]: this is a deep one-class model. It classifies anomaly data by penalizing the distance between the extracted feature vector, from the network and the center of the initial hypersphere. Since it only supports non-trivial high-dimensional images, we use the transformer encoder in T-GP to extract features for Deep-SVDD.
- *HRN* [35]: this is a state-of-the-art one-class models based on holistic regularization. We use the default structure with a three-layer perception, whose input, hidden and output dimensions are 40, 100, and 1, respectively.
- *T-L2*: this is a variant of our T-GP model. We replace the gradient penalty-based regularization with L2-regularization.

Evaluation Results We use the standard metrics (precision, recall and F1-measure) to quantify and compare the performance of our model with others baselines. Figure 10 shows the results on the testing datasets of off-road and wrong-way attacks. Note that in anomaly detection tasks, anomaly data are considered as positive. From Figures 10(a) and 10(b), we can find that for both kinds of attacks, the transformer-based models (i.e., T-L2 and T-GP) have higher average precision and lower variance than other models. Hence, the adoption of the transformer exhibits better robustness. They can detect anomalies more precisely with fewer false alarms. As shown in Figures 10(c) and 10(d), the two transformer-based models also have higher average recall than others, indicating that they have smaller false negative rates, i.e., missing fewer anomaly data. Moreover, compared to T-L2, T-GP can provide more fine-grained control over the penalty function and a higher recall with smaller fluctuations. The F1-measure results are shown in Figures 10(e) and 10(f). We also find that T-GP has the highest F1-measure. It means T-GP not only has high precision and recall values, but also can balance these two measures. Hence, we conclude that the proposed T-GP outperforms other one-class models on the 20 testing sequences.

TABLE 2
Levene's test and t-test on F1-value between our T-GP and each of other models. A higher value indicates the model is more similar as T-GP in detection performance.

Baselines		OC-SVM	IF	DSVDD	HRN	T-GP
Off-road attack	Levene's test	0.3908	0.0025	0.1346	0.0060	0.2606
	T-test	4e-11	0.0012	0.0026	0.0007	0.1337
Wrong-way attack	Levene's test	0.0180	0.0023	0.0482	0.0003	0.5477
	T-test	2e-10	0.0003	0.0489	0.0017	0.2549

To analyze the statistical significance of these models, we perform Levene's test and two-sample t-test for equal variance testing and equal mean testing in terms of the F1-measure. The results are shown in Table 2. We can observe that given the 95% confidence interval, our T-GP has significant differences for the mean of F1-measure, from other non-transformer models. Hence, T-GP demonstrates higher performance statistically. Moreover, we can find that there are no significant differences between T-GP and T-L2, indicating the two loss functions in T-GP and T-L2 have similar performance in balancing the precision and recall.

During the online anomaly detection, another important requirement is to detect attacks promptly so that we can prevent accidents as soon as possible. Hence, we also compute the detection time of different models in *Apollo*. We find that T-GP can detect an attack within 6 data samples after launching the attack ($\sim 0.07s$), while other models need more time to identify anomalous events, which is relatively less practical in reality.

In conclusion, our transformer-based model can accurately disclose the underlying dependency in the time series data during the AV's motion, whilst other models cannot describe such temporal relations, even using the sliding window technique. Moreover, the results also show that the transformer with GP is better than with L2 regularization.

6.1.2 *Defeating Traffic Sign Recognition Attacks*

We examine the effectiveness of our model on detecting adversarial traffic signs.

Datasets. We conduct our experiments on the *GTSRB* (German Traffic Sign Recognition Benchmark) dataset, which only contains clean traffic sign images. We select four representative categories of traffic signs, i.e., stop, speed limit, keep right and traffic signals, from this dataset for training.

TABLE 3
Number of images in each dataset

Attack	Traffic Sign	Training	Test	
		Normal	Normal	Abnormal
Boundary	Stop	780	270	20
Poster	Stop	780	270	270
Poster	Speed limit 30	2220	720	720
Poster	Keep right	2070	690	690
Poster	Traffic signals	600	180	180

The numbers of these categories are 780, 2220, 2070, and 600, respectively. For testing, we adopt the boundary attack [12] and poster attack [4] to generate adversarial example from the normal testing images. Specifically, we perform the boundary attack on the stop sign category to generate 20 adversarial samples, and the poster attack on the four categories to generate the same numbers of adversarial images as the testing samples. Figure 11 visualizes the adversarial samples of different attacks and traffic signs.

(a) (b)

Fig. 11. Clean (first row) and adversarial (second row) traffic signs. (a) Boundary attack (b) Poster attack.

.

Table 3 gives the details of the training and testing datasets. We remove 10% border of each category and resize the images to 32×32 as presented in [34]. In addition, global contrast normalization using L1-norm is applied.

Model Configurations. For T-GP, we use the same structure described in Section 4.2, where each input image is divided into 64 patches with an equal size of 4×4. According to the scale of the datasets, λ is set as around 1.5 (similar results for $[0.1, 3]$) and the initial learning rate is 3e-4.

Baseline Methods. We compare our model with Deep-SVDD and HRN in detecting adversarial traffic signs. Specifically, for Deep-SVDD, we apply a CNN structure with three filters of sizes $32 \times (5 \times 5 \times 3)$, $64 \times (5 \times 5 \times 3)$ and $128 \times (5 \times 5 \times 3)$, followed by a fully connected layer with 128 units. We get the maximum accuracy with the AdamW optimizer whose learning rate is set as $1e-3$. For HRN, a three-layer MLP is adopted with the size of 3×[1024-300]-[900-300]-[300-1]. The first layer contains three sub-modules (each one has a size of [1024-300]) to deal with 3 channels, and the outputs are concatenated as the input of the second layer; the second and third layers have the sizes of [900-300] and [300-1], respectively. The optimizer is set as SGD with momentum and the learning rate is $5e-4$.

Evaluation Results. Table 4 shows the AUC (Area Under the ROC) values of different models for detecting the boundary and poster attacks on different traffic signs. The results show that our model outperforms Deep-SVDD and HRN for both kinds of attacks.

TABLE 4
Average AUCs for different models in detecting different attacks

Attack	Traffic Sign	Deep-SVDD	HRN	T-GP
Boundary	Stop	80.78%	95.5%	**98.2%**
Poster	Stop	72.46%	72.83%	**93.26**%
Poster	Speed limit 30	51.96%	64.18 %	**65.56**%
Poster	Keep right	62.64%	83.54%	**84.03**%
Poster	Traffic signals	76.46%	85.68%	77.83%

TABLE 5
Average AUCs for different transformers and loss functions in detecting poster attacks

Solution	Stop	Speed limit 30	Keep right	Traffic signals
T-NLL	88.81%	59.07%	82.01%	77.59%
T-L2	60.25%	63.88%	81.85%	86.98%
T-GP	93.26%	65.56%	84.03%	77.83%

We also compare the performance of the transformer-based one-class model with three kinds of loss functions: NLLLoss, L2 penalty and GP (gradient penalty). Table 5 shows the detection results of the loss functions on the poster attack. We can observe that the model with gradient penalty introduced from WGAN has higher AUC values than the other two loss functions. A possible reason is that, since the output has just one class, we use sigmoid(·) function in NLLloss to calculate the probability of input x labeling y(x). When minimizing the NLLloss function, we need to include a penalty function to reduce the possibility of feature bias. Such feature bias exists as we do not have any other classes to compare, and do not know which feature is essential for class differentiation. Some features and their related parameters with high values may not be important, thus leading a low accuracy.

6.1.3 *Defeating Lane Detection Attacks*

Datasets To evaluate the effectiveness of our method on detecting lane attacks, we adopt the widely-used *Tusimple* traffic lane dataset. This dataset consists of 6,408 annotated images, which are the latest frames from video clips recorded by a high-resolution (720×1280) forward-view camera under various traffic and weather conditions on highways of United States in the daytime. It is spilt into a training set (3268), a validation set (358), and a testing set (2782). We generate two types of adversarial examples from the validation set following the Patch Attack [11], including fixed-size patch and varied-size patch (Figure 2). The size of the former patch is 100×100, and the later patch is scaled according to the lane width and lane marker height. After adding the adversarial patches, all the images are scaled to the size of 320×320. For each type of patches, we obtain 3268 normal images used for training, 358 normal images and 358 abnormal images for testing. Figure 12 shows two adversarial samples under the fixed- and varied-size patch attacks, respectively.

Model Configurations Different with the configurations in adversarial traffic sign detection, we add a split layer before the model input, thus the images are spilt into fixed-size patches first in order to capture the anomalies more carefully. Specifically, we split each image of 320×320×3 to 100 patches of 32×32×3. This gives us 3268×100 training samples, 358×100 normal testing samples and 358×100 abnormal

(a) (b)

Fig. 12. Samples of fixed-size patch and varied-size patch.

TABLE 6
Average AUCs of different models in detecting the patch attacks

Patch Attack	Deep-SVDD	HRN	T-GP
Fixed-size	68.19%	52.79%	**92.25**%
Varied-size	60.60%	51.54%	**67.86**%

testing samples. During testing, if any one of the 100 patches is flagged as abnormal, then the entire image is regarded as anomaly. We use the same preprocessing method for all the models to achieve fair comparison.

Baseline Methods. We compare our transformer-based method with Deep-SVDD and HRN. The two models follow the same settings in Section 6.1.2.

Evaluation Results. Table 6 shows the average AUC values for different models. We observe that T-GP shows better performance than the other two baseline models. Particularly, all these models have relatively low accuracy in detecting the varied patch attacks. One possible reason is that some patches are too small to be recognized as adversarial samples, causing higher false negative rates. But T-GP still outperforms prior solutions. We will explore new models to further enhance the detection accuracy as future work.

6.2 Evaluation of `ADS-Lead`

We evaluate the effectiveness of `ADS-Lead` with lifelong and federated learning on the attack detection. As we discovered in the GPS spoofing detection experiments, the pattern of IMU data shows no divergence in different scenarios when the AV is running in normal and secure states. Hence, the redundant IMU samples from different vehicles cannot further improve the performance of the proposed detector. Therefore, we mainly focus on the detection of traffic sign attacks and lane attacks in this section.

6.2.1 Datasets

In federated learning, each vehicle participates in gradient update during the training process. Therefore, assigning sufficient training data for each vehicle is crucial for the convergence of the model. To amend this, data argumentation is performed over the training datasets on each vehicle.

For the traffic sign data sets, we first rotate the images clockwise and counterclockwise by 5, 10 and 15 degrees, respectively; second, we divide the data into two subsets to represent tasks at two different time instants. Considering the impact of environmental factors (e.g., light, whether and camera resolution), we randomly synthesize the latter subset with the effects of rain by adding controlled random noise. The statistics of the traffic sign datasets are reported in Table 7. Note that for in each testing set, the abnormal samples are generated by the poster attack.

TABLE 7
Number of images in each traffic sign datasets. Note the abnormal data are generated by the poster attack

Datasets	Traffic Sign	Training	Test	
		Normal	Normal	Abnormal
Task 1 (Non-Rainy)	Stop	3855	270	50
	Speed limit 30	10970	720	100
	Keep right	10230	690	100
	Traffic signals	2965	180	20
Task 2 (Rainy)	Stop	1605	270	50
	Speed limit 30	4570	720	100
	Keep right	4260	690	100
	Traffic signals	2965	180	20

TABLE 8
Number of images in lane detection datasets. Note the abnormal data in Task 1 and Task 2 include both varied and fixed patch attacks

Fixed_Varied	Training	Test	
	Normal	Normal	Abnormal
Task 1 (Non-Rainy)	1634	358	358
Task 2 (Rainy)	1634	358	358

For lane detection attacks, we expand the data set by adding rain effects to the original images. Specifically, we first divide the original training data set equally into two subsets: Task 1 for the first phase of training and Task 2 for model update; moreover, we synthesize the images in Task 2 with the same rainy effects as in the traffic sign data set. Second, for either testing data set of Task 1 or Task 2, we apply both the fixed- and varied-size patch attacks to generate adversarial samples, and the testing data set in Task 2 is also added with the rain effects. Table 8 shows the statistics of the two data sets. Figure 13 shows some samples with rainy effects in traffic sign and lane detection data sets.

6.2.2 Baseline Models and Model Configurations

We compare the following algorithms for model update:

BaseModel: This is for federated learning only. In our experiments, we consider a system of 5 vehicles, partition the training data sets equally into 5 sets, and assign each to one vehicle. For each round, we randomly select 4 vehicles to update the gradients for aggregation, to simulate the asynchronous mechanism. We set the batch size as 32 and run 50 epochs with the same hyper-parameters of T-GP as shown in Section 4.

Fed-Finetune: In addition to training the model with federated learning, we further finetune the aggregated model using the dataset from Task 2.

`ADS-Lead`: This is our solution in `ADS-Lead`. In addition to training the model with federated learning, we further perform lifelong learning on Task 2 to obtain the updated model. We adopt the same federated learning settings, i.e., batch size of 32 and 50 running epochs.

6.2.3 Evaluation Results

Figure 14 presents the AUC values of the three algorithms for the two tasks of traffic sign attack detection, respectively. We can find that **BaseModel** performs well on Task 1 (e.g., 91.48% for stop sign) but not well on Task 2 (e.g., 77.17% for stop sign), as the model is trained only from Task 1.

Fig. 13. The synthesized images with rain.

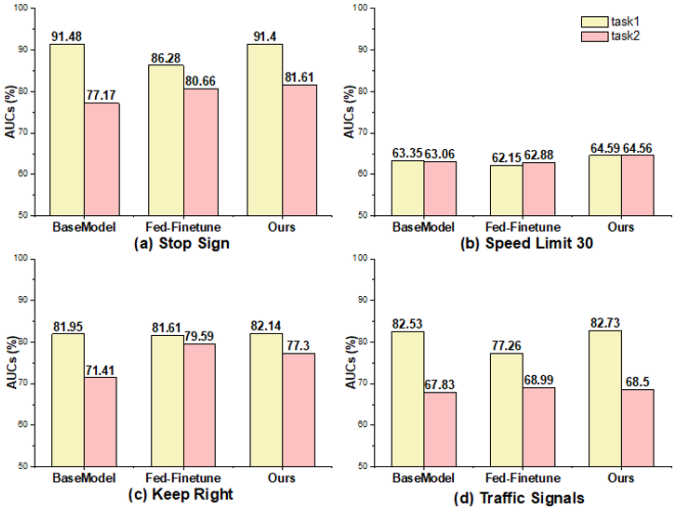


Fig. 14. Evaluation results on traffic sign dataset. **BaseModel**: the federated learning model is trained on Task 1, and tested on Task 1 and Task 2. **Fed-Finetune**: the federated learning model trained on Task 1, and finetuned on Task 2. **Our** `ADS-Lead`: the model is trained on Task 1 and lifelone learned on Task 2.

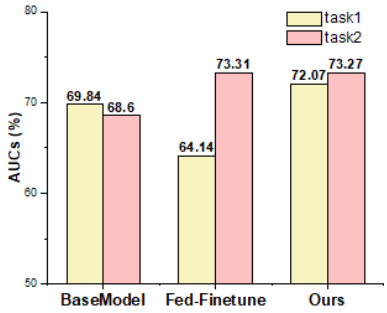


Fig. 15. Evaluation results on the lane detection dataset.

Fed-Finetune improves the performance over **BaseModel** on Task 2 (e.g., from 77.17% to 80.66% for stop sign) due to the fine-tuning operation with the dataset of Task 2. However, its performance on Task 1 is degraded (e.g., from 91.48% to 86.28% for stop sign). This indicates that simply finetuning the model can make it learn new knowledge but forget some prior knowledge. Our `ADS-Lead` model can balance the performance on both Task 1 and Task 2. In detail, we observe that the model performance on Task 1 and Task 2 is similar as **BaseModel** and **Fed-Finetune**, respectively. Hence, with lifelong learning, our model can not only learn new knowledge of new tasks (e.g., Task 2) but also remember the learned knowledge from previous tasks (e.g., Task 1).

Similarly, Figure 15 demonstrates the effectiveness of `ADS-Lead` on lane attack detection. For **Fed-Finetune**, after model fine-tuning on Task 2, the prediction accuracy of Task 2 rises from 68.60% to 73.32%, whereas the accuracy of Task 1 drops from 68.60% to 64.14%. Fortunately, with lifelong learning, our `ADS-Lead` balance the model performance on Task 1 and Task 2 significantly.

Even though the results are encouraging, the improvement brought by lifelong learning for Speed Limit 30 and Keep Right signs is limited. This is because the data in Task 2 are synthesized by only adding normal noise to simulate the rainy effects, and the pattern difference between Task 1 and Task 2 is not very significant. Despite that, the experimental results still show that our proposed `ADS-Lead` is practical for anomaly detection in ADSs, achieved by the globally-trained high-quality model with lifelong learning.

6.3 Discussion on the Robustness of `ADS-Lead`

Finally, we discuss the robustness of our system against possible adaptive attacks. Even though the adversary knows the defense mechanism, it is hard for him to attack our detector. On one hand, federated learning can mitigate the attacks on a single vehicle, as the server will aggregate the local models to generate a global one. On the other hand, with lifelong learning, the server will update the model over time, so each vehicle will update its model such that the adversary cannot use the previous knowledge on the model to launch attacks. We also point out that it is possible for the adversary to launch attacks during two successive update time instants. However, these attacks can be mitigated by setting specific update frequency such that there is no enough time for the adversary to retrieve the model information and then launch proper attacks. How to design more advanced attacks as well as enhancing the system will be our future work.

7 Conclusion

In this paper, we propose `ADS-Lead`, a novel system based on federated learning and lifelong learning to detect anomalies in the lane following scenario of ADSs. We introduce T-GP, a novel one-class classification model with a transformer encoder for feature extraction and new loss function with gradient penalty. It is able to detect GPS spoofing, traffic sign recognition and lane detection attacks with high accuracy. We extensively evaluate our model on the mainstream Baidu *Apollo* ADS with the LGSVL simulator, and two public traffic datasets: GTSRB and Tusimple. The results show that T-GP significantly outperforms existing state-of-the-art one-class models. We also show the practicality and effectiveness of attack detection with advanced model evolution solutions. In the future, we aim to incorporate our system into real-world AVs and study the anomaly detection of other sensor attacks (e.g., Lidar attacks) and scenarios (e.g., lane changing and overtaking).

ACKNOWLEDGMENTS

This work was supported in part by Singapore Ministry of Education (MOE) AcRF Tier 1 RG108/19 (S), NTU-Desay Research Program 2018-0980, Singapore MOE Academic Research Fund Tier 2 grant (MOE-T2EP20120-0004), Singapore National Research Foundation (NRF) under its National Cybersecurity R&D Program (NRF2018NCR-NCR005-0001 and NRF2018NCR-NSOE003-0001), and NRF Investigatorship (NRF-NRFI06-2020-0001).

REFERENCES

[1] H. Su, M. Qiu, and H. Wang, "Secure wireless communication system for smart grid with rechargeable electric vehicles," *IEEE Communications Magazine*, vol. 50, no. 8, pp. 62–68, 2012.

[2] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, "Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under GPS spoofing," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020, pp. 931–948.

[3] T. Sato, J. Shen, N. Wang, Y. J. Jia, X. Lin, and Q. A. Chen, "Hold tight and never let go: Security of deep learning based automated lane centering under physical-world attack," *arXiv preprint arXiv:2009.06701*, 2020.

[4] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[5] 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.

[6] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[7] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[8] F. Zhu, Y. Lv, Y. Chen, X. Wang, G. Xiong, and F.-Y. Wang, "Parallel transportation systems: toward iot-enabled smart urban traffic control and management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4063–4071, 2019.

[9] A. Purwar, D. Joshi, and V. K. Chaubey, "GPS signal jamming and anti-jamming strategy - A theoretical analysis," in *2016 IEEE Annual India Conference (INDICON)*. IEEE, 2016, pp. 1–6.

[10] K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip, and R. Gerdes, "Intelligent transportation system security: Impact-oriented risk assessment of in-vehicle networks," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 2, pp. 91–104, 2021.

[11] H. Xu, A. Ju, and D. Wagner, "Model-agnostic defense for lane detection against adversarial attack," in *Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, vol. 2021, 2021, pp. 1–5.

[12] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *International Conference on Learning Representations*, 2018.

[13] M. L. Psiaki, B. W. O'Hanlon, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "GPS spoofing detection via dual-receiver correlation of military signals," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2250–2267, 2013.

[14] J. Magiera and R. Katulski, "Detection and mitigation of GPS spoofing based on antenna array processing," *Journal of Applied Research and Technology*, vol. 13, no. 1, pp. 45–57, 2015.

[15] A. Boudhir, M. Benahmed, A. Ghadi, and M. Bouhorma, "Vehicular navigation spoofing detection based on V2I calibration," in *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. IEEE, 2016, pp. 847–849.

[16] M. Qiu, Z. Ming, J. Li, J. Liu, G. Quan, and Y. Zhu, "Informer homed routing fault tolerance mechanism for wireless sensor networks," *Journal of Systems Architecture*, vol. 59, no. 4-5, pp. 260–270, 2013.

[17] J. Liu and J. Park, ""seeing is not always believing": Detecting perception error attacks against autonomous vehicles," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[18] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your GPS are belong to us: Towards stealthy manipulation of road navigation systems," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1527–1544.

[19] P. F. Swaszek, S. A. Pratz, B. N. Arocho, K. C. Seals, and R. J. Hartnett, "GNSS spoof detection using shipboard IMU measurements," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, 2014, pp. 745–758.

[20] Y. Wu, H.-B. Zhu, Q.-X. Du, and S.-M. Tang, "A survey of the research status of pedestrian dead reckoning systems based on inertial sensors," *International Journal of Automation and Computing*, vol. 16, no. 1, pp. 65–83, 2019.

[21] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in IoT systems," *IEEE Int. of Things J.*, pp. 1–9, 2021.

[22] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," in *25th Annual Network and Distributed System Security Symposium (NDSS)*, 2018.

[23] K. Lee, K. Lee, H. Lee, and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in *NeurIPS*, 2018, pp. 1–11.

[24] F. Li, X. Liu, X. Zhang, Q. Li, K. Sun, and K. Li, "Detecting localized adversarial examples: A generic approach using critical region analysis," *arXiv preprint arXiv:2102.05241*, 2021.

[25] 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.

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

[27] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, D. Song, M. E. Houle, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," *arXiv preprint arXiv:1801.02613*, 2018.

[28] Z. Katzir and Y. Elovici, "Detecting adversarial perturbations through spatial behavior in activation spaces," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–9.

[29] Y. Li, Y. Song, L. Jia, S. Gao, Q. Li, and M. Qiu, "Intelligent fault diagnosis by fusing domain adversarial training and maximum mean discrepancy via ensemble learning," *IEEE Trans. on Indu. Info.*, vol. 17, no. 4, pp. 2831–2842, 2021.

[30] J. Wang, G. Dong, J. Sun, X. Wang, and P. Zhang, "Adversarial sample detection for deep neural network through model mutation testing," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*. IEEE, 2019, pp. 1245–1256.

[31] S. Tian, G. Yang, and Y. Cai, "Detecting adversarial examples through image transformation," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[32] Q. Sun, A. A. Rao, X. Yao, B. Yu, and S. Hu, "Counteracting adversarial attacks in autonomous driving," in *2020 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*. IEEE, 2020, pp. 1–7.

[33] S. Safavi, M. A. Safavi, H. Hamid, and S. Fallah, "Multi-sensor fault detection, identification, isolation and health forecasting for autonomous vehicles," *Sensors*, vol. 21, no. 7, p. 2547, 2021.

[34] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4393–4402.

[35] W. Hu, M. Wang, Q. Qin, J. Ma, and B. Liu, "HRN: A holistic approach to one class learning," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[36] M. Zhu, X. Liu, F. Tang, M. Qiu, R. Shen, W. Shu, and M. Wu, "Public vehicles for future urban transportation," *IEEE Trans. on Intell. Trans. Sys.*, vol. 17, no. 12, pp. 3344–3353, 2016.

[37] H. Qiu, Q. Zheng, M. Msahli, G. Memmi, M. Qiu, and J. Lu, "Topological graph convolutional network-based urban traffic flow and density prediction," *IEEE Trans. on Intell. Trans. Sys.*, 2020.

[38] A. Liu, X. Liu, J. Fan, Y. Ma, A. Zhang, H. Xie, and D. Tao, "Perceptual-sensitive gan for generating adversarial patches," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, 2019, pp. 1028–1035.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, vol. 30, 2017, pp. 1–11.

[40] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[41] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[42] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *31st International Conference on Neural Information Processing Systems*, 2017, pp. 5769–5779.

[43] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Networks*, vol. 113, pp. 54–71, 2019.

[44] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[45] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[46] S. Rangarajan, M. Verma, A. Kannan, A. Sharma, and I. Schoen, "V2c: A secure vehicle to cloud framework for virtualized and on-demand service provisioning," in *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, 2012, pp. 148–154.

[47] S. Guo, T. Zhang, X. Xie, L. Ma, T. Xiang, and Y. Liu, "Towards byzantine-resilient learning in decentralized systems," *arXiv preprint arXiv:2002.08569*, 2020.

[48] W. Gao, S. Guo, T. Zhang, H. Qiu, Y. Wen, and Y. Liu, "Privacy-preserving collaborative learning with automatic transformation search," in *Conference on Computer Vision and Pattern Recognition*, 2021.

[49] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class SVM for learning in image retrieval," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 1. IEEE, 2001, pp. 34–37.

[50] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 8th IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

Xingshuo Han is a Ph.D. student at School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include safety and privacy of deep learning, safety and security of autonomous vehicles, and intelligent transportation system.

Yuan Zhou received his M.S. degree in computational mathematics from Zhejiang Sci-Tech University, Hangzhou, China, in March 2015 and received his Ph.D. degree in computer science from Nanyang Technological University, Singapore, in June 2019. He is currently a Research Fellow in School of Computer Science and Engineering at Nanyang Technological University, Singapore. His research interests include motion planning for multi-robot systems, Security of autonomous vehicles, and self-adaptive systems.

Kangjie Chen received the B.E. degree from University of Electronic Science and Technology of China in 2015 and the M.E. degree from Tianjin University in 2019. He is now a second-year Ph.D. student at School of Computer Science and Engineering, Nanyang Technological University. His research interests include safety and privacy of deep learning, reinforcement learning, and adversarial machine learning.

Han Qiu received the B.E. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2011, the M.S. degree from Telecom-ParisTech (Institute Eurecom), Biot, France, in 2013, and the Ph.D. degree in computer science from the Department of Networks and Computer Science, Telecom-ParisTech, Paris, France, in 2017. He worked as a postdoc and a research engineer with Telecom Paris and LINCS Lab from 2017 to 2020. Currently, he is an assistant professor at Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing, China. His research interests include AI security, big data security, and the security of intelligent transportation systems.

Meikang Qiu received the BE and ME degrees from Shanghai Jiao Tong University and received Ph.D. degree of Computer Science from University of Texas at Dallas. He is the Department Head and tenured full professor of Texas A&M University Commerce. He is ACM Distinguished Member and IEEE Senior Member. He is the Chair of IEEE Smart Computing Technical Committee. He has published 20+ books, 600+ peer-reviewed journal and conference papers, including 100+ IEEE/ACM Transactions papers. His research interests include Cyber Security, Big Data Analysis, Cloud Computing, Smarting Computing, Intelligent Data, Embedded systems, etc. He is an Associate Editor of 10+ international journals, including IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Transactions on Big Data, and IEEE Transactions on SMC. He is ACM Distinguished Member (2019), Highly Cited Researcher (2020, Web of Science), and IEEE Distinguished Visitor (2021-2023).

Yang Liu received the B.Comp. degree (Hons.) from the National University of Singapore (NUS) in 2005 and the Ph.D. degree from NUS and MIT, in 2010. He started his postdoctoral work in NUS and MIT. He is currently a Full Professor and the Director of the Cybersecurity Laboratory, NTU. He specializes in software verification, security, and software engineering. His research has bridged the gap between the theory and practical usage of formal methods and program analysis to evaluate the design and implementation of software for high assurance and security. By now, he has more than 270 publications in top tier conferences and journals. He received a number of prestigious awards, including the MSRA Fellowship, the TRF Fellowship, the Nanyang Assistant Professor, the Tan Chin Tuan Fellowship, the Nanyang Research Award, and eight best paper awards in top conferences, such as ASE, FSE, and ICSE.

Tianwei Zhang is an assistant professor at School of Computer Science and Engineering, Nanyang Technological University. His research focuses on computer system security. He is particularly interested in security threats and defenses in machine learning systems, autonomous systems, computer architecture and distributed systems. He received his Bachelor's degree at Peking University in 2011, and the Ph.D degree in at Princeton University in 2017.