2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom) | 978-1-6654-3574-1/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00119

A Unified Anomaly Detection Methodology for Lane-Following of Autonomous Driving Systems

Xingshuo Han*, Kangjie Chen*, Yuan Zhou*‡, Meikang Qiu†, Chun Fan§, Yang Liu*, Tianwei Zhang*
*Nanyang Technological University, Singapore 639798
Email: {xingshuo001,kangjie001}@e.ntu.edu.sg, {y.zhou, yangliu, tianwei.zhang}@ntu.edu.sg
†Texas A&M University Commerce, TX, USA 75428
Email: meikang.qiu@tamuc.edu
§Peng Cheng Laboratory & Peking University, China
Email: fanchun@pku.edu.cn
‡Corresponding author

***Abstract*—Autonomous Vehicles (AVs) are equipped with various sensors and controlled by Autonomous Driving Systems (ADSs) to provide high-level autonomy. 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. Most of the current works focus on anomaly detection of specific attacks, such as GPS spoofing or traffic sign attacks. There are no works on scenario-aware anomaly detection for ADSs. In this paper, focusing on the lane-following scenario, we introduce a novel transformer-based one-class classification model to identify time series anomalies and adversarial image examples. It can detect GPS spoofing, traffic sign recognition and lane detection attacks with high efficiency and accuracy. We further design a Swin-transformer model to enhance the detection performance. Experiments on Baidu *Apollo* and two public data sets (GTSRB and Tusimple) show that compared with the state-of-the-art methods, our method, on average, improves the detection performance by 9.7%, 14.7% and 15.7% for GPS spoofing, traffic sign recognition and lane detection attacks, respectively.**

***Index Terms*—One-Class Classification, Autonomous Driving Systems, Transformer, Multi-source Anomaly Detection**

I. Introduction

Autonomous Vehicles (AVs) will play an essential role in modern intelligent transportation systems to reduce traffic accidents and congestion [1], [2]. Recent advances in the technologies of computing, automation and artificial intelligence inspire many companies to devote themselves to this promising domain and accelerate the commercialization of autonomous driving, e.g., Baidu *Apollo* [3], Google *Waymo* [4].

To guarantee high-level automation, Autonomous Driving Systems (ADSs) serve as the brain of AVs, which communicate with the external environment and internal vehicle components, and make driving decisions. Due to the complex environment and requirements, most of the current ADSs are scenario-sensitive, i.e., they have different tasks to complete under different scenarios (lane following, lane changing, overtaking, and intersections, etc.) based on the information from different sensors. For example, in the lane following scenario, an AV is required to move along the central lines of lanes. So the preliminary task for an ADS is to recognize the lane boundaries and locate the central lines. Cameras and GPS are required to achieve this function. In the overtaking scenario, an ADS needs to recognize surrounding obstacles and determine whether it is safe to perform overtaking. The decision is made from the data in Lidar and GPS.

The high complexity of ADSs inevitably brings a broad attack surface [5]. For example, an adversary can launch GPS spoofing attacks to mislead AVs to navigate to a dangerous position [6]. The attack cost is only $200 for a low-end "GPS spoofing" device. By adding malicious patches [7], paint [8] or stickers [9] on the road or traffic signs, an adversary can make ADSs perceive the environment mistakenly and make wrong decisions [10], [11]. Attacks on Lidar can deceive ADSs into ignoring the surrounding obstacles, resulting in collisions [12], [13]. Different attacks may cause different damages under different scenarios. For instance, adversarial attacks against Lidars target obstacle avoidance rather than lane following, which mainly depends on AV's localization and lane detection; GPS spoofing focuses on the lane following and change scenarios.

In this paper, we consider the security protection of 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). We aim to introduce a unified methodology to detect any anomalies during lane following, and mitigate different types of security vulnerabilities, i.e., localization attacks, lane detection attacks, and traffic sign recognition attacks. They have significant impacts on the functionality of ADSs, and it is important for vehicles to be immune to them for secure and safe driving. Although prior studies proposed some solutions to defeat sensor attacks for AVs [13]–[16], 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.

We develop a novel detection methodology, called T-GP (Transformer with Gradient Penalty), to analyze and identify time series anomalies (localization attacks) and adversarial images (i.e., lane detection attacks and traffic sign recognition

978-1-6654-3574-1/21/$31.00 ©2021 IEEE

DOI 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00119

attacks) in the lane following scenario of AVs. T-GP is a one-class classification model, which needs to be trained offline only from normal data. Then it is implemented in ADSs as an online detector to inspect different sources of sensory data and discover the suspicious input. T-GP is built from a one-layer transformer encoder. It introduces a novel loss function, which combines the Negative Log Likelihood (NLL) with the Gradient Penalty (GP). We also design S-GP, a Swin transformer based model, for effective feature extraction of images. The integration of these techniques gives very high accuracy for anomaly detection of various attacks.

We apply our proposed model on datasets from the real world, and collected from simulations to comprehensively evaluate its effectiveness. 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 [17]. We follow [6] to implement GPS attacks in LGSVL, 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 Tusimple datatset, and adopt the attack method in [18] to generate fixed and variable adversarial patches. For traffic sign attacks, we use the GTSRB dataset. We implement the boundary attacks [19] and poster attacks [9] 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 T-GP, a novel one-class classification model based on the transformer for anomaly detection in the lane following scenario. It can be integrated into ADSs to detect both time series anomalies and adversarial images.
- With the T-GP model, we utilize the instantaneous changes of IMU data to detect attacks on localization. Evaluations show that our model can identify malicious GPS input in 0.07s in the mainstream Baidu *Apollo* ADS.
- T-GP shows extraordinary performance over state-of-the-art models on the detection of both traffic sign and lane attacks. We further introduce an advanced S-GP model to improve the performance of anomaly detection on traffic sign attacks.

II. Background and Threat Model

A. Autonomous Driving System

An ADS needs to recognize the external environment and promptly produce the correct motion commands to the vehicle. Hence, a typical ADS usually consists of the following modules. They collaborate closely to achieve the above functions.

- *Localization*: this 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.
- *Perception*: this 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., traffic lights, stop signs and speed limits) and obstacles (e.g., object types, the speeds of other vehicles on the road).
- *Planning*: this module performs offline path planning to generate a feasible path from the initial position to the destination based on the map information. It conducts real-time trajectory planning, which utilize the results from the localization and perception modules to generate a collision-free trajectory in a short time duration.
- *Control*: this module finally generates low-level commands, such as steering, throttle and brake, to the chassis to track the generated collision-free trajectory.

B. Security Threats

An ADS may face different types of scenarios based on different map topology [20]–[22]. In this paper, we focus on lane following, the most common scenario during AV operations. In this scenario, the vehicles are required to move along the central lines of lanes. The execution of an ADS highly depends on the accuracy of localization, lane boundary detection and traffic signs. Past studies have proposed different attacks to compromise the execution of ADSs in lane following. The goal of this paper is to design a method, which can detect these attacks in an efficient and unified way.

1) 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 [6] where the spoofed GPS signals can dominate the inputs of the MSF process and fool MSF to ignore other inputs. Specifically, when the victim vehicle is moving along the straight lane, the attacker vehicle follows the victim vehicle and 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. To make the vehicle stay in the center of the lane, the ADS asks the vehicle to move, which actually makes it cross the lane.

The attacks can have two specific goals, as shown in Figure 1: an *off-road attack* tries to lead the victim to hit the curb; a *wrong-way attack* tries to deviate the victim AV to the opposite pavement.

2) Lane Detection Attack: An ADS needs 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. 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 control 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

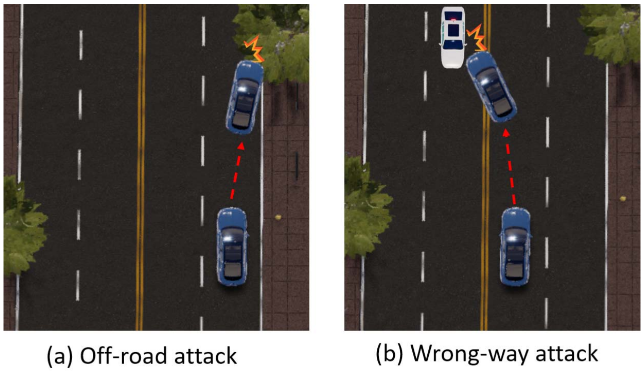
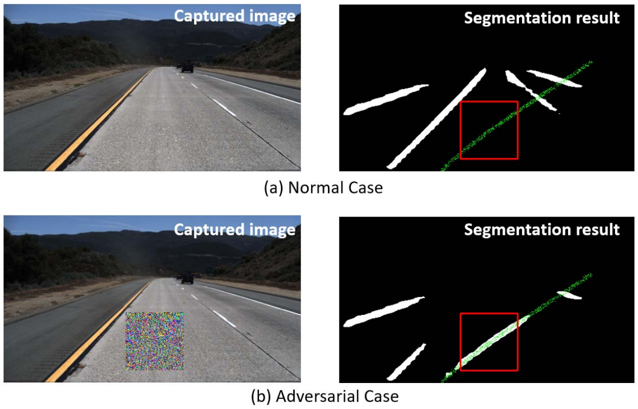


Fig. 1. GPS spoofing attack.



(a) Normal Case

(b) Adversarial Case

Fig. 2. Lane detection attack.

surrounding object [8]. This is also verified in [7], where the injection of three small patches on the road can compromise the recognition system of Tesla *Autopilot*. Figure 2 shows an attack example [18]. The first row shows the clean road image with the corresponding lane segmentation results analyzed the ADS. The ADS is able to correctly identify the lane boundaries (white). For the second row, the adversary adopts the Projected Gradient Descent [23] to carefully craft an adversarial patch and inject it to the road. Then based on the segmentation results, the ADS will recognize a wrong lane boundary around the patch. The red boxes show the patch localization, which is computed according to the virtual induced lanes (green).

In our paper, we consider two types of patch attacks [18]: (1) fixed-size patches, whose size is 100×100 and is injected to the images of 512×288; (2) varied-size patches, whose sizes are scaled based on the distances from the camera to the destination lane segments.

3) Traffic Sign Recognition Attack: Recognition of traffic signs can also affect the lane following since an AV must obey the traffic rules described by those signs. As the ADS leverages CNN models to detect and classify traffic signs, an adversary can leverage the adversarial attack techniques to compromise the model, and the ADS will miss or misclassify the traffic signs and generate wrong motion decisions. This requires the adversary to physically alter the traffic signs (e.g., adding posters or patches) without changing their visual semantics.

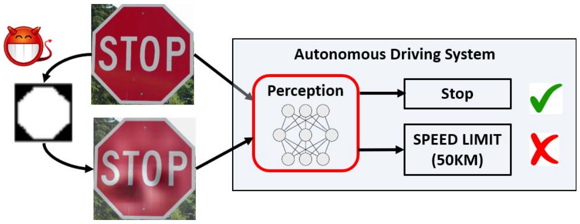


Fig. 3. Traffic sign recognition attack.

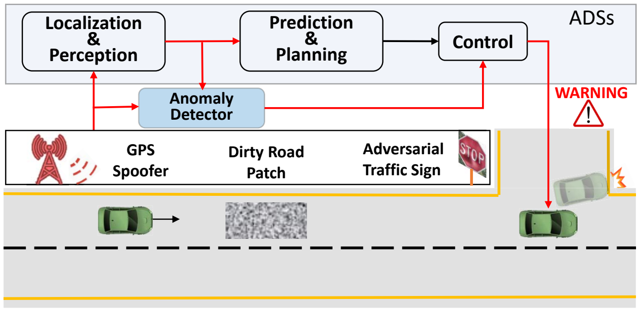


Fig. 4. Overview of our anomaly detection methodology.

Typically there are two types of attacks in this category: (1) In a *poster attack*, the adversary generates malicious posters for traffic signs using a novel Robust Physical Perturbations algorithm [9], and then attach them to the traffic sign. Then the perception module in the ADS will identify it as a different sign. Figure 3 shows such a poster attack on a stop sign [9]. Alternatively, the adversary can also adopt generative adversarial networks to craft malicious patches to compromise the traffic sign recognition model [24]. (2) A *boundary attack* is a decision-based adversarial attack [19]. 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.

III. Methodology

In this section, we describe our novel methodology to detect the above attacks in the lane-following scenario. Figure 4 shows the methodology overview. We introduce a powerful anomaly detector, deployed in an ADS to monitor the outputs of the perception and localization modules. 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.

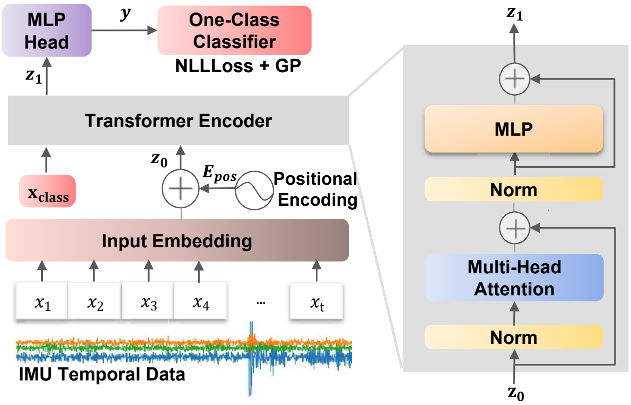


Fig. 5. T-GP model structure.

The essential component of the anomaly detector is a one-class classification model for inspecting various types of sensory data. Below is the detailed description of this model.

A. Transformer with Gradient Penalty (T-GP)

Our novel one-class model, T-GP, is based on the transformer structure for anomaly detection in the lane following scenario. A transformer [25] is a deep neural network using the self-attention mechanism. It replaces the Recurrent Neural Network (RNN) structure with an encoder and decoder. Figure 5 shows the structure of T-GP. It adopts a 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). Note that main computation cost of T-GP is the computation of the Transformer Encoder, whose computational complexity can be found in [25].

Input Preprocessing. 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 [26]. For the IMU data, each single sample x_i is recorded at a time instant. The input sequence X is first mapped to patch embeddings z_0 using a learnable embedding vector x_{class}, a trainable linear projection E, and a standard learnable 1D position embeddings $\mathbf{E}_{pos}$ [26], as given in Equation 1:

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

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.

Classification. The patch embeddings z_0 is sent to the transformer encoder, which 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) [27]. Hence, the operation of the transformer encoder can be formulated as:

$$z_1' = MSA(LN(z_0)) + z_0, \tag{2}$$

$$z_1 = MLP(LN(z_1')) + z_1'. \tag{3}$$

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. It generally requires regularizaion due to the sigmoid saturation and feature bias in NLLLoss [28]. Inspired by [29], which adds 1-Lipschitz constraints to the discriminator of WGAN by gradient penalty (GP), we also consider gradient penalty in T-GP to obtain the following loss function:

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

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 [28].

B. Swin-Transformer with Gradient Penalty (S-GP)

We propose an enhanced transformer-based one-class classification model, S-GP, to further improve the accuracy and robustness of our anomaly detector for image data.

S-GP is based on the Swin-Transformer, which is the state-of-the-art backbone network widely applied to different CV tasks [30]. S-GP follows the structure of the original Swin-Transformer: it consists of four stages with each stage composed of two consecutive Swin Transformer blocks, as shown in Figure 6. To adapt to our task, we replace the standard MSA module with a window-based MSA in each Swin transformer block: the MSA in the first Swin transformer block applies a regular windowing configuration (W-MSA), while the MSA in the second block utilizes a shifted windowing configuration (SW-MSA). It adopts the same loss function in Equation 4. Similarly, in S-GP, the main computation cost lies in the computation of the (S)W-MSA, whose computation complexity is given in [30].

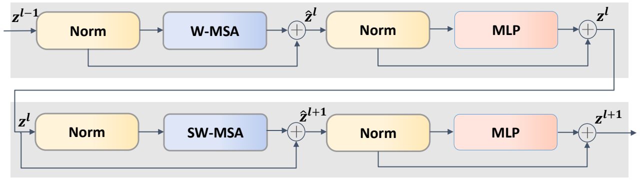


Fig. 6. A stage with two consecutive blocks in Swin-GP.

Given an image with a size of $H \times W \times 3$, S-GP first uses patch partition to divide it into a set of non-overlapping patches with the size of 4×4, resulting in $H/4\times W/4$ patches. In stage 1, the patch feature is transformed into a vector with a dimension of C through a linear embedding and then sent to the Swin-Transformer Block. Stages 2-4 share the similar operations but through a patch merging operator, where the input is merged according to 2×2 adjacent patches. Hence, the number of patch blocks and the feature dimension of each patch after Stage i $(i = 2, 3, 4)$ become $H/2^{i+1} \times W/2^{i+1}$ and $2(i-1)C$, respectively.

IV. Implementation

A. Datasets

GPS spoofing attacks. Since there are no public datasets, 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. Following the attack settings in [6], we consider two concrete adversarial goals as shown in Figure 1: wrong-way attack aims to deviate the AV to the opposite lane and hit the oncoming vehicle; off-road attack aims to deviate the AV to hit the curb, and left attacks. 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), angular 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 normal lane change or turn. Hence, at each time instant, we collect these four types of data as the model features. Figure 7 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. We first 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 are generated for model training. The testing set should contain both normal and attack samples. We run *Apollo* ten times under either type 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 I summarizes the ten testing data sequences for either GPS spoofing attack.

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 a 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.

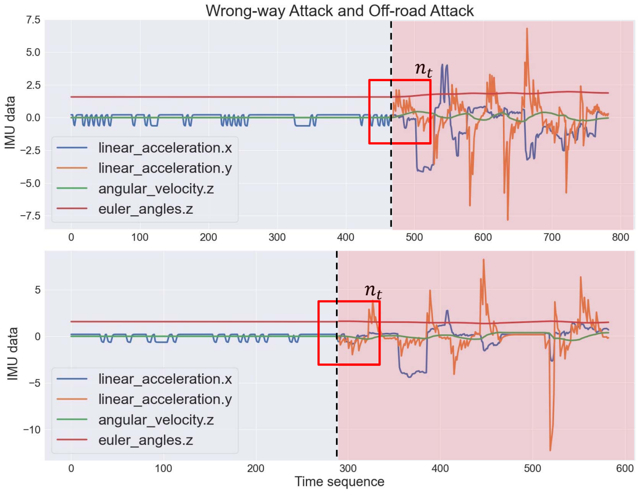


Fig. 7. Data sequences of ax, ay, avz, and γ when the AV is under the wrong-way and off-road attacks, respectively. The black line represents the moment the spoofing attack starts. The red area shows the vehicle is in abnormal status. The red box is the sliding window with a length of $n = 10$ data samples. n_t represents that the attack is detected after n_t occurrences of the attack.

TABLE I
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

Traffic sign recognition attacks. 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 20, keep right, and traffic signals, from this dataset for training. The numbers of these categories are 780, 210, 2070, and 600, respectively. For testing, we adopt the boundary attack [19] and poster attack [9] to generate adversarial examples 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. Table II gives the details of the datasets. We remove 10% border of each category and resize the images to 32×32 as presented in [31]. In addition, global contrast normalization using L1-norm is applied.

Lane detection 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 data), a validation set (358 data), and a testing set (2782 data). We generate two types of adversarial examples from the validation

TABLE II
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 20	210	60	60
Poster	Keep right	2070	690	690
Poster	Traffic signals	600	180	180

set following the Patch Attack [18], 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 get 3268 normal images for training, 358 normal images and 358 abnormal images for testing.

B. Model configurations

To detect GPS spoofing attacks, the input dimension of T-GP is set as 10×4 according to the format of the generated data samples, 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. The λ in Equation 4 is set as 0.1.

To detect traffic sign recognition attacks, we use the same structure described in III-A, where each input image is divided into 64 patches with an equal size of 4×4. λ is set around as 1.5 (similar results for $[0.1, 3]$) and the initial learning rate is 3e-4. For the S-GP model, we utilize the same Swin-Transformer hyper-parameters for all the data sets, where the hidden dimension is 96, the layers (resp., heads and downscaling factors) of the four stages are 2 (resp., 3 and 4), 2 (resp., 6 and 2), 6 (resp., 12 and 2), and 2 (resp., 24 and 2), respectively, and the number of output classes is 1. Since the input image size is 32×32, the window size is set as 1. We apply the AdamW optimizer whose learning rate is around 5e-6 for defeating the boundary attack and 5e-5 for the poster attack. λ is set as 2.5.

To detect lane recognition attacks, different from the 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 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.

V. EVALUATIONS

In this section, we evaluate the effectiveness and robustness of the proposed anomaly detection model against the three kinds of attacks described in Section II-B.

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

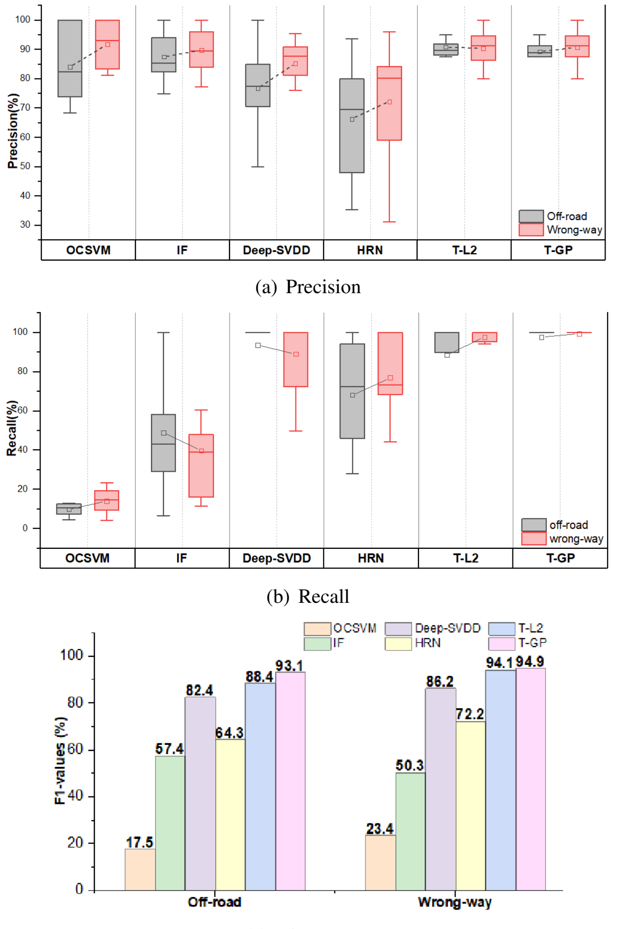
- *OC-SVM* [32]: it is a traditional one-class classifier based on kernel SVM. In our implementation, the RBF kernel is applied and the hyper-parameter is selected from a set of discretized values in the interval $[0, 1]$.
- *iForest* [33]: it is another popular one-class classifier. It isolates anomaly points by building decision trees. We use the default values of the hyper-parameters.
- *Deep-SVDD* [31]: it 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* [28]: it 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*: it is a variant of our T-GP model. We replace the gradient penalty-based regularization with L2-regularization.

We also try the GAN-based models [34]. However, these methods require a large amount of training data, and the training processes cannot converge on our small-scale datasets. Hence, they are not applicable in our task, and will not be considered as the baselines.

A. Effectiveness against Localization Attacks

We use the standard metrics (precision, recall and F1-measure) to compare the performance of our model with others baselines. Figure 8 shows the results on the testing datasets of off-road and wrong-way attacks. In anomaly detection tasks, the anomaly data is considered as positive. From Figures 8(a), 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 Figure 8(b), 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 provide a higher recall with smaller fluctuations. The F1-measure results are shown in Figure 8(c). We can also find that the T-GP model 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 can conclude that T-GP outperforms other one-class models on the 20 testing sequences.

To analyze the statistical significance of these models, we perform Levene's test and two-sample t-test [35] for equal variance testing and equal mean testing, respectively, in terms of the F1-measure. The results are shown in Table III. 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



(a) Precision

(b) Recall

(c) F1-measure

Fig. 8. Results of Precision, Recall and F1-measure on the two GPS spoofing attack datasets.

TABLE III
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-L2
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

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.

Another important requirement for online anomaly detection 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.

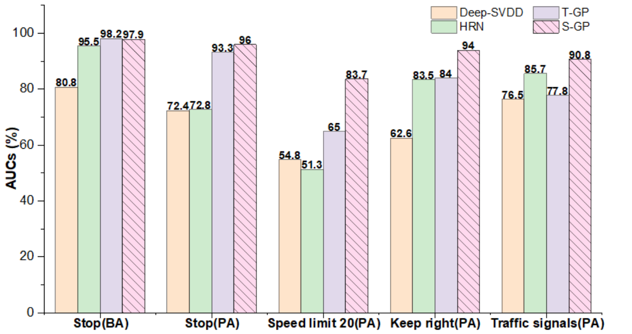


Fig. 9. Average AUCs for different models in detecting boundary attacks (BA) and poster attacks (PA) against traffic signs.

In conclusion, our transformer-based models 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.

B. Effectiveness against Traffic Sign Recognition Attacks

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

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 $32\times(5\times5\times3)$, $64\times(5\times5\times3)$ and $128\times(5\times5\times3)$, 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 are with the size of [900-300] and [300-1], respectively. The optimizer is set as SGD with momentum and the learning rate is $5e-4$.

Figure 9 shows the AUC (Area Under the ROC curve) values of different models for detecting the boundary attack (BA) and poster attack (PA) on different traffic signs. The results show that our two models outperform Deep-SVDD and HRN for both kinds of attacks, and S-GP shows performance improvement over T-GP.

We also compare the performance of the two transformer-based models with three kinds of loss functions: NLLLoss, L2 penalty and Gradient Penalty (GP). Figure 10 shows the detection performance of the two models on the two kinds of poster attacks. We can observe that the model with gradient penalty has higher AUC values than the other two loss functions. S-GP model gives the best results over other solutions for all the attacks.

C. Effectiveness against Lane Detection Attacks

We compare our transformer-based methods, i.e., T-GP and S-GP, with Deep-SVDD and HRN. The settings of these two models are the same as the ones described in Section V-B.

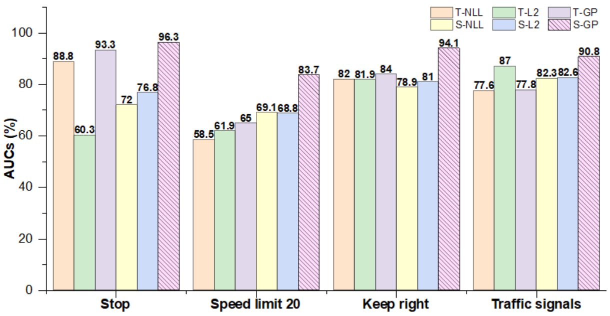


Fig. 10. Average AUCs for different transformers and loss functions in detecting poster attacks against traffic signs.

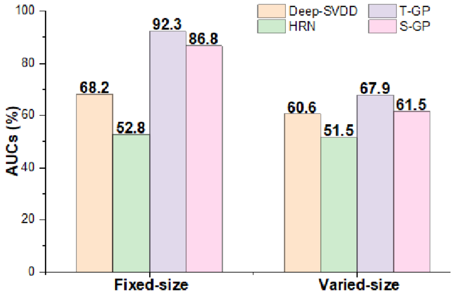


Fig. 11. Average AUCs of different models in detecting the patch attacks against lane detection.

Figure 11 presents the average AUC values for different models. We can observe that the two transformer-based models show better performance than other two baseline models. T-GP shows better detection performance than S-GP. Particularly, all these models have relatively low accuracy in detecting 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 the transformer-based models still outperform prior solutions. We will explore new models and algorithms to further enhance the detection accuracy as future work.

VI. Related Works

A. Detecting Localization Attacks

Although prior works made some attempts to detect GPS attacks against AVs [36]–[38], 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 [39], [40]. Unfortunately, Shen *et al.* [6] found a vulnerability in the design of MSF-based localization and implemented a sophisticated attack to invalidate the protection. Researchers also studied spoofing detection by cross-checking GPS readings and IMU data [41]. However, IMU data suffers 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 [42]. Compared with these studies, we only use the instantaneous changes of the IMU data to detect whether the vehicle is being attacked and achieve high detection accuracy.

B. Detecting Adversarial Images

Some works introduced methods to detect adversarial examples, especially in the CV domain. One popular direction is to build classifiers to differentiate adversarial examples from normal samples, based on their hidden unique features. Xu *et al.* [43] 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.* [44] designed a method using Gaussian discriminant analysis to obtain the confidence score based on the Mahalanobis distance in the feature space of DNN models. However, these methods need prior knowledge of the adversarial samples, which is hard to be satisfied in the autonomous driving scenario. Other works, e.g. Deep-SVDD [31], OCNN [45], HRN [28], introduced one-class models for anomaly detection of adversarial examples. They are only evaluated on the stop sign detection. For lane attacks, Sato *et al.* [8] 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.* [18] designed a CNN-based model with prior knowledge of abnormal data to achieve attack detection. These works can only be applied to specific attacks, but fail to be extended to others. In contrast, our proposed solution is unified to cover various types of attacks with different formats of sensory data in the lane following scenario.

VII. Conclusion

In this paper, we proposed to leverage the transformer to build anomaly detection models for the lane following scenario of ADSs. We introduced T-GP, a novel one-class classification model based on a transformer encoder for feature extraction and new loss function with gradient penalty. It can detect GPS spoofing, traffic sign recognition and lane detection attacks with high accuracy. We further designed S-GP, an enhanced model over T-GP to improve the detection accuracy of adversarial image samples. We extensively evaluated our models on the mainstream Baidu *Apollo* ADS with the LGSVL simulator, and two public traffic datasets: GTSRB and Tusimple. The results showed that our models significantly outperform existing state-of-the-art one-class models. In the future, we aim to incorporate our models into real-world AVs and study the anomaly detection of other sensor attacks.

Acknowledgments

This work was supported in part by Key-Area Research and Development Program of Guangdong Province (No.2019B121204008), 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] 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.

[2] 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.

[3] "Baidu Apollo," https://github.com/lgsvl/apollo-5.0.

[4] "Google Waymo," https://waymo.com/.

[5] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2014.

[6] 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.

[7] "Experimental security research of tesla autopilot," https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Research_of_Tesla_Autopilot.pdf, 2019, Tencent Keen Security Lab.

[8] 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.

[9] 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 *IEEE Conf. on Computer Vision and Pattern Recognition*, 2018, pp. 1625–1634.

[10] 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 Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2831–2842, 2021.

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

[12] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *29th USENIX Security Symp. (USENIX Security)*, 2020, pp. 877–894.

[13] 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 *ACM SIGSAC Conf. on Computer and Comm. Security*, 2019, pp. 2267–2281.

[14] 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.

[15] 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.

[16] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig *et al.*, "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.

[17] G. Rong, B. H. Shin, H. Tabatabaee, Q. Lu, S. Lemke, M. Možeiko, E. Boise, G. Uhm, M. Gerow, S. Mehta *et al.*, "LGSVL simulator: A high fidelity simulator for autonomous driving," in *IEEE 23rd Int'l Conf. on Intell. Trans. Systems (ITSC)*, 2020, pp. 1–6.

[18] 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.

[19] 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.

[20] Y. Tang, Y. Zhou, Y. Liu, J. Sun, and G. Wang, "Collision avoidance testing for autonomous driving systems on complete maps," in *2021 IEEE Intelligent Vehicles Symposium (IV21)*, 2021.

[21] Y. Tang, Y. Zhou, F. Wu, Y. Liu, J. Sun, W. Huang, and G. Wang, "Route coverage testing for autonomous vehicles via map modeling," in *IEEE Int. Conf. Robot.d Autom.*, 2021.

[22] Y. Tang, Y. Zhou, T. Zhang, F. Wu, Y. Liu, and G. Wang, "Systematic testing of autonomous driving systems using map topology-based scenario classification," in *The 36th IEEE/ACM Int.l Conf. Automat. Softw. Eng.: New Ideas and Emerging Results (NIER) track*, 2021.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018, pp. 1–23.

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

[30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *arXiv preprint arXiv:2103.14030*, 2021.

[31] 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*, 2018, pp. 4393–4402.

[32] Y. Chen, X. S. Zhou, and T. S. Huang, "One-class SVM for learning in image retrieval," in *IEEE Int. Conf. Image Process.*, vol. 1, 2001, pp. 34–37.

[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *8th IEEE Int'l Conf. on Data Mining*, 2008, pp. 413–422.

[34] P. Perera, R. Nallapati, and B. Xiang, "OCGAN: One-class novelty detection using GANs with constrained latent representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2898–2906.

[35] N. A. Heckert and J. J. Filliben, "NIST/SEMATECH e-Handbook of Statistical Methods; Chapter 1: Exploratory Data Analysis," 2003.

[36] 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.

[37] 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.

[38] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, "Prediction-based GNSS spoofing attack detection for autonomous vehicles," *arXiv preprint arXiv:2010.11722*, 2020.

[39] 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.

[40] S. M. Albrektsen, T. H. Bryne, and T. A. Johansen, "Robust and secure UAV navigation using GNSS, phased-array radio system and inertial sensor fusion," in *2018 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2018, pp. 1338–1345.

[41] P. Guo, H. Kim, N. Virani, J. Xu, M. Zhu, and P. Liu, "RoboADS: Anomaly detection against sensor and actuator misbehaviors in mobile robots," in *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2018, pp. 574–585.

[42] 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," *Int'l Jou. of Auto. and Computing*, vol. 16, no. 1, pp. 65–83, 2019.

[43] 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.

[44] 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.

[45] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.