VehiGAN: Generative Adversarial Networks for Adversarially Robust V2X Misbehavior Detection Systems

MD HASAN SHAHRIAR, Virginia Tech, USA
MOHAMMAD RAASHID ANSARI, Self, USA
JEAN-PHILIPPE MONTEUUIS, Qualcomm Technologies, Inc., USA
MD SHAHEDUL HAQUE, Virginia Tech, USA
CONG CHEN and JONATHAN PETIT, Qualcomm Technologies, Inc., USA
Y. THOMAS HOU and WENJING LOU, Virginia Tech, USA

Vehicle-to-Everything (V2X) communication enables vehicles to communicate with other vehicles and roadside infrastructure, enhancing traffic management and improving road safety. However, the open and decentralized nature of V2X networks exposes them to various security threats, especially misbehaviors, necessitating a robust misbehavior detection system (MBDS). While machine learning (ML) has proved effective in different anomaly detection applications, the existing ML-based MBDSs have shown limitations in generalizing due to the dynamic nature of V2X and insufficient and imbalanced training data. Moreover, they are known to be vulnerable to adversarial ML attacks. On the other hand, generative adversarial networks (GAN) possess the potential to mitigate the aforementioned issues and improve detection performance by synthesizing unseen samples of minority classes and utilizing them during their model training. Therefore, we propose the first application of GAN to design an MBDS that detects any misbehavior and ensures robustness against adversarial perturbation.

In this paper, we present several key contributions. First, we propose an advanced threat model for stealthy V2X misbehavior where the attacker can transmit malicious data and mask it using adversarial attacks to avoid detection by ML-based MBDS. We formulate two categories of adversarial attacks against the anomaly-based MBDS. Later, in the pursuit of a generalized and robust GAN-based MBDS, we train and evaluate a diverse set of Wasserstein GAN (WGAN) models and present Vehicular GAN (VehiGAN), an ensemble of multiple top-performing WGANs, which transcends the limitations of individual models and improves detection performance. We present a physics-guided data preprocessing technique that generates effective features for ML-based MBDS. In the evaluation, we leverage the state-of-the-art V2X attack simulation tool VASP to create a comprehensive dataset of V2X messages with diverse misbehaviors. Evaluation results show that in 20 out of 35 misbehaviors, VehiGAN outperforms the baseline and exhibits comparable detection performance in other scenarios. Particularly, VehiGAN excels in detecting advanced misbehaviors that manipulate multiple fields in V2X messages simultaneously, replicating unique maneuvers. Moreover, VehiGAN provides approximately 92% improvement in false positive rate under powerful adaptive adversarial attacks, and possesses intrinsic robustness against other adversarial attacks that target the false negative rate. Finally, we make the data and code available for reproducibility and future benchmarking, available at https://github.com/shahriar0651/VehiGAN.

Authors' Contact Information: Md Hasan Shahriar, Virginia Tech, Arlington, Virginia, USA, hshahriar@vt.edu; Mohammad Raashid Ansari, Self, Massachusetts, USA, m.raashid.ansari@gmail.com; Jean-Philippe Monteuuis, Qualcomm Technologies, Inc., Boxborough, Massachusetts, USA, jmonteuu@qti.qualcomm.com; Md Shahedul Haque, Virginia Tech, Arlington, Virginia, USA, mdshahedul@vt.edu; Cong Chen, congchen@qti.qualcomm.com; Jonathan Petit, Qualcomm Technologies, Inc., Boxborough, Massachusetts, USA, petit@qti.qualcomm.com; Y. Thomas Hou, thou@vt.edu; Wenjing Lou, Virginia Tech, Arlington, Virginia, USA, wjlou@vt.edu.

ACM 2378-9638/2025/6-ART
https://doi.org/10.1145/3745787

CCS Concepts: • **Networks** → **Mobile and wireless security**; • **Security and privacy** → **Intrusion detection systems**; • **Computing methodologies** → *Neural networks.*

Additional Key Words and Phrases: Vehicular Network, Misbehavior Detection Systems, Generative Adversarial Networks, Deep learning, Adversarial Attacks

1 INTRODUCTION

Road traffic accidents take 1.35 million lives every year around the world, leaving another 50 million non-fatally injured [35]. Approximately 94% of major accidents in conventional transportation systems are caused, at least in part, by human errors [47]. Conversely, a cooperative intelligent traffic system (C-ITS) has the potential to help reduce these human errors and save millions of lives. One of the fundamental enabling technologies of C-ITS is Vehicle-to-Everything (V2X) communication that allows vehicles to communicate with their environment, such as other vehicles (V2V), infrastructure (V2I), networks (V2N), and pedestrians (V2P) [32]. V2X technology provides vehicles with real-time traffic information along with alerts on potential hazards, which help coordinate traffic flow, avoid collisions, and reduce fatalities and injuries on the roads.

Moreover, V2X can also augment safe, efficient, and convenient autonomous driving systems. By V2X communication protocols, connected vehicles transmit Basic Safety Messages (BSMs) (also known as Cooperative Awareness Messages (CAM) in the European Union), as defined in the SAE J2735 standard [9]. A BSM primarily contains a short-term pseudonym for sender identification, current location, speed, acceleration, direction, etc., and is generally transmitted every 100 milliseconds. A security credential management system (SCMS) incorporates a public key infrastructure (PKI) to deliver digital certificates to the vehicles that serve as a signature for the exchanged messages [8]. Such a cryptographic solution secures V2X by thwarting any outsider attackers from sending bogus messages.

While V2X has the potential to boost C-ITS and is secure against an outsider attacker, it still poses several security challenges [23], especially from insider attackers. Insider attackers have valid access credentials but disseminate incorrect information to achieve attack goals [33]. Hence, while digital signatures confirm the origin of the BSMs, they cannot ensure the truthfulness of the content. Such malicious actions by rogue insiders, referred to as "misbehaviors" in V2X, are hard to detect through cryptographic methods and can seriously threaten road safety. On the other hand, a misbehavior detection system (MBDS) continuously checks for such potential misbehavior by analyzing the transmitted messages, serving as an essential defense for the V2X communication system [23].

The MBDS, usually running on an ego vehicle, receives BSMs from another vehicle and checks whether the content has anomalies or is physically implausible [23]. Upon observing a potential anomaly, it reports such an event with its evidence to the misbehavior authority (MA), another component of SCMS, following a misbehavior reporting protocol (MBR) [23]. Such reporting allows the MA to further investigate and penalize the malicious vehicle, if needed, by putting its credentials on the certificates revocation list (CRL) to isolate it from the V2X network [8].

Nevertheless, MBDS confronts a multitude of formidable challenges [7], rendering it a complex and evolving research task. There are different MBDSs proposed in the existing research [7] to detect malicious or erroneous V2X messages. While the state-of-the-art threat landscape has become quite broad [3], mandating a comprehensive solution, most of the existing MBDSs provide a partial defense, focusing on specific types of attacks and features [23]. Although traditional supervised deep learning (DL) models have the capability to learn complex V2X data distribution and detect misbehaviors, they struggle to generalize well due to the lack of sufficient and imbalanced training datasets [36]. On the other hand, generative adversarial networks (GAN) [16], a generative DL model, has already demonstrated its capacity to overcome data imbalance issues by synthesizing unseen benign samples of minority classes (such as rare vehicular states) and incorporating them into the training

process [44]. Consequently, GAN can effectively identify zero-day attacks by capturing subtle deviations from benign behavior. Besides, the traditional DL-based methods—classification or anomaly detection—are proven vulnerable to adversarial ML attacks [24], where noise-like perturbations are added to mislead the model's prediction. Unlike traditional MBDS, GAN's training through implicit density estimation makes it intrinsically robust against adversarial attacks. Thus, by utilizing both the generative and discriminative powers along with a powerful learning technique, GAN possesses the potential to serve as an effective anomaly detection system [21]. To the best of our knowledge, we are the first to explore the adaptability of GAN in designing a generalized and robust MBDS for V2X.

Our contributions are as follows[1]:

- We outlined an enhanced threat model for stealthy V2X misbehavior, in which the attacker induces a malicious BSM field and employs adversarial ML attacks to conceal it, thereby evading detection by ML-based MBDS. To investigate the robustness against adversarial adaptive attackers, we further formulate two categories of attacks targeting the GAN-based MBDS.
- We study the feasibility of using Wasserstein GAN (WGAN), one of the most prominent and stable variants of the GANs [4], to design an unsupervised DL-based MBDS for V2X communication. To overcome the limitations of the individual WGAN, we propose VehiGAN, an ensemble of multiple top-performing WGANs that provides enhanced detection performance across misbehaviors and robustness against adversarial attacks. We introduce two distinct techniques for selecting top-performing WGANs for ensemble and address the unique challenges associated with each approach. We present a physics-guided data preprocessing technique for the V2X dataset that generates effective features from raw BSM fields for any ML-based MBDS.
- Employing the state-of-the-art V2X attack simulation tool VASP [3], we generate an extensive V2X message dataset containing 68 distinct types of misbehaviors, representing a substantial enhancement compared to prior V2X misbehavior datasets [29, 55]. We make them publicly available [42] to advance state-of-the-art MBDS research.
- We evaluate VehiGAN against 35 different types of misbehaviors (as the other 33 misbehaviors do not fit our threat model), and compare the performance with various anomaly detection techniques. The results indicate that VehiGAN achieves the best detection performance in 20 out of 35 misbehaviors, particularly against advanced ones that manipulate multiple fields in V2X messages, replicating unique maneuvers, with a comparable high performance against the rest.
- To investigate the adversarial robustness, we consider a spectrum of adversarial scenarios, ranging from white-box to black-box settings and from single-model to multi-model attacks. VehiGAN shows approximately 92% improvement in false positive rate under one type of powerful adaptive attack and intrinsic robustness against other types of attacks that aim for high false negatives.

The paper is organized as follows: Section 2 provides background information, followed by the problem formulation and threat model in Section 3. In Section 4, we detail the design of our proposed system, VehiGAN. Section 5 describes the experimental setup and implementation, and Section 6 presents our evaluation results. Related work is discussed in Section 7, discussion is in Section 8, and finally, we conclude in Section 9.

2 PRELIMINARIES

This section provides the necessary background on C-ITS, V2X communication, and GANs, which serve as foundational components for the development of VehiGAN.

[1]A preliminary version of this research was presented at the IEEE International Conference on Distributed Computing Systems (ICDCS) 2024 [43]

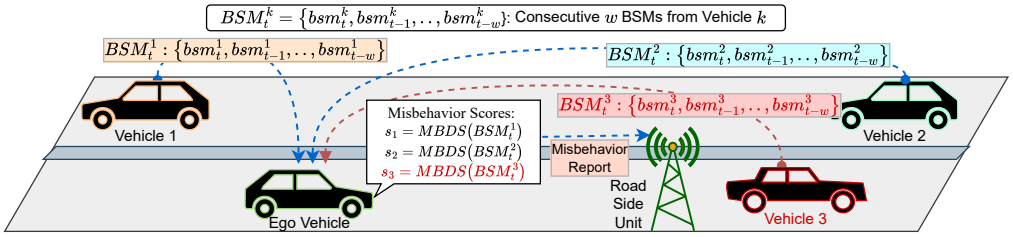


Fig. 1. V2X communication systems where the ego vehicle collects w consecutive BMSs for each vehicle and analyzes them using the MBDS to check for potential misbehavior. MBDS detects the malicious messages transmitted by vehicle 3 (shown in red color) and reports that to MA through the roadside unit.

2.1 Cooperative Intelligent Transport Systems (C-ITS)

Cooperative Intelligent Transport Systems (C-ITS) utilize wireless technology to enable real-time communication between vehicles and infrastructure [7]. This connectivity facilitates enhanced coordination among road users, leading to safer and more efficient traffic flows.

2.1.1 V2X Communication. V2X communication is a fundamental technology of C-ITS that enables vehicles to communicate with other vehicles (V2V), infrastructure (V2I), and road users, such as pedestrians and cyclists (V2P) [7]. By facilitating this exchange of information, V2X enhances situational awareness, supports advanced driver assistance systems, and contributes to safer and more efficient transportation systems. A key component of V2X is the basic safety message (BSM), which is a standardized data packet broadcasted by vehicles to share critical safety information, such as speed, position, acceleration, brake status, etc., with nearby vehicles and infrastructure. V2X safety applications can enhance driver awareness and prevent accidents by providing timely alerts in critical scenarios [53]. For instance, emergency electronic brake lights (EEBL) notify drivers of sudden braking in traffic ahead, allowing them more time to respond to potential hazards. Forward collision warning (FCW) helps avoid rear-end crashes by alerting drivers to immediate threats in front. Similarly, such applications, including intersection movement assist (IMA), left turn assist (LTA), blind spot/lane change warning (BSW/LCW), etc., can improve road safety by giving drivers more time to make informed decisions in high-risk situations.

2.1.2 V2X Vulnerabilities. V2X communication, while enhancing safety and situational awareness, is particularly vulnerable to attacks from insider threats—malicious actors within the V2X network who possess valid credentials [33]. These insiders can exploit V2X protocols and data to disrupt traffic flow and create hazardous conditions. Misbehavior in the context of V2X refers to any action that deviates from expected behavior. *A misbehaving entity – either intentionally or unintentionally – manipulates and transmits data improperly or inappropriately, resulting in unexpected behavior [7]* Misbehavior can stem from malicious intent, such as targeting any of the V2X safety applications of other entities and misleading them to an unsafe driving condition. To counter such threats, an MBDS is essential. *An MBDS monitors V2X data to identify suspicious or anomalous patterns indicative of misbehavior.* By filtering out potentially harmful information from malicious insiders, MBDS strengthens the integrity and reliability of V2X communication, helping to maintain safety across the V2X ecosystem. Fig. 1 illustrates a V2X application scenario with MBDS running on the ego vehicle.

2.2 Generative Adversarial Networks

GAN, introduced by Ian Goodfellow in 2014 [16], is an implicit generative model based on artificial neural networks. It has become a popular technique for generating realistic data (e.g., image, video, audio) that resembles the distribution of the training dataset. GAN consists of two neural networks: a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The generator's role is to transform a random noise vector $\mathbf{z}$ drawn from a simple distribution (P_z) into

fake-but-realistic data samples $\mathbf{x}_{fake} = \mathcal{G}(\mathbf{z})$. The discriminator, on the other hand, is tasked with distinguishing between real sample $\mathbf{x}_{real}$ from the training data distribution (P_r) and generated fake data sample $\mathbf{x}_{fake}$. While $\mathcal{G}$ is trained to deceive $\mathcal{D}$ into accepting the fake data as real, $\mathcal{D}$ is optimized to discriminate both the real and fake samples correctly. Hence, when both networks are sufficiently trained, $\mathcal{D}$ can be used as an anomaly detection model to discriminate between benign and malicious inputs [44].

Out of different variants of GAN, we adopted Wasserstein GAN (WGAN) with gradient penalty, which is the most popular due to its high performance, robustness, and training stability [19]. WGAN solves a min-max optimization problem, where $\mathcal{G}$ is trained to minimize the Wasserstein distance between the real and fake data samples, and $\mathcal{D}$ is trained to maximize such distance. The objective of WGAN is to find the parameters of the generator ($\theta_{\mathcal{G}}$) and the discriminator ($\theta_{\mathcal{D}}$) that satisfy a Nash equilibrium [16]. Mathematically, the optimization problem can be expressed as:

$$\min_{\theta_{\mathcal{G}}} \max_{\theta_{\mathcal{D}}} \left[\mathbb{E}_{x \sim P_r}[\mathcal{D}(\mathbf{x})] - \mathbb{E}_{z \sim P_z}[\mathcal{D}(\mathcal{G}(\mathbf{z}))] \right] \tag{1}$$

Here, $\mathcal{D}(x_{real})$ and $\mathcal{D}(\mathcal{G}(z)) = \mathcal{D}(x_{fake})$ are the outputs of the discriminator on a real and a generated sample, x_{real} and x_{fake}, respectively. Thus, $\mathcal{G}$ and $\mathcal{D}$ learn together in an adversarial training fashion, making them efficient in their individual tasks. From one perspective, with the help of $\mathcal{G}$, $\mathcal{D}$ implicitly learns the complex distribution of the benign (real) data distribution, making it a good candidate for an anomaly-based MBDS [58].

As a GAN model consists of two components—$\mathcal{G}$ and $\mathcal{D}$—its evaluation involves assessing each component individually using distinct metrics. Below, we present different ways for evaluating $\mathcal{G}$ and $\mathcal{D}$.

2.2.1 Evaluation of Generator. As the $\mathcal{G}$ outputs synthetic/fake data samples, its performance is evaluated based on the resemblance of the generated data to the real training data. There are different methods to evaluate the quality of the n fake samples $\mathbf{X}_{fake} = \{\mathbf{x}^1_{fake}, \mathbf{x}^2_{fake}, \mathbf{x}^3_{fake}, \cdots, \mathbf{x}^n_{fake}\}$ by comparing them with the set of n real samples $\mathbf{X}_{real} = \{\mathbf{x}^1_{real}, \mathbf{x}^2_{real}, \mathbf{x}^3_{real}, \ldots, \mathbf{x}^n_{real}\}$.

Distance-based Evaluation. Converting real and synthetic data samples into two multivariate datasets by concatenating them sequentially is a common practice for evaluating generative models. Let the concatenated matrices are expressed as $\mathbf{X}^{concat}_{real} = [\mathbf{x}^1_{real}\ \mathbf{x}^2_{real}\ \cdots\ \mathbf{x}^n_{real}]^T$ and $\mathbf{X}^{concat}_{fake} = [\mathbf{x}^1_{fake}\ \mathbf{x}^2_{fake}\ \cdots\ \mathbf{x}^n_{fake}]^T$. Distance-based evaluation metrics assess the statistical relation between the distributions of $\mathbf{X}^{concat}_{real}$ and $\mathbf{X}^{concat}_{fake}$. By analyzing the distributions, correlations, and summary statistics, such methods estimate how closely the generated samples resemble those of the real data, which eventually indicates the generator's performance. However, they do not consider the temporal characteristics of the time-series data.

Model-based Evaluation. ML models work under the assumption that the training and the testing data come from the same distribution. Such fundamental assumption is used to evaluate the generator's performance by training a model (such as an autoencoder [26]) using one dataset (either $\mathbf{X}_{real}$ or $\mathbf{X}_{fake}$) and then testing it on the other dataset. The discrepancy between the model's performance on training and testing datasets indicates how similar the generated and real data are and, therefore, how well the generator performs. Such methods retain both the temporal and spatial properties of the data and, hence, are more suitable for time-series datasets.

Computer Vision-based Evaluation. Such methods are almost similar to the distance-based methods. However, instead of comparing the individual samples directly, these metrics first use a pre-trained inception model [51] trained on a real-world image dataset (*e.g.*, ImageNet [10]), to extract representative features. Let I be the feature extractor based on the inception model, and such methods use different distance-based similarity metrics to estimate how close the feature distributions $F_{real} = I(\mathbf{X}_{real})$ and $F_{fake} = I(\mathbf{X}_{fake})$ are, indicating the performance of the generator.

2.2.2 Evaluation of Discriminator. Unlike Vanilla GANs [16], the $\mathcal{D}$ in a WGAN is designed to output higher values for benign inputs. In this work, we use the negative of the discriminator's output as the anomaly score. Thus, let $\mathcal{A}$ be a generalized anomaly-based misbehavior detection process so that:

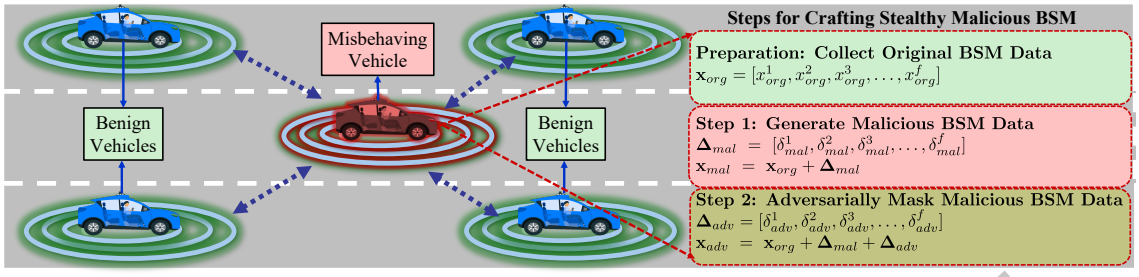


Fig. 2. Process for generating adversarially stealthy malicious Basic Safety Messages (BSMs) by an insider attacker in V2X communication. In Step 1, malicious data is introduced into specific BSM fields, and in Step 2, adversarial perturbations are applied to conceal these malicious modifications.

$$\mathcal{A}(.) = -\mathcal{D}(.) \tag{2}$$

We then apply a thresholding strategy, commonly used in anomaly detection, to classify samples based on their anomaly scores, thereby evaluating the performance of the discriminator. The metrics used to evaluate both the $\mathcal{G}$ and $\mathcal{D}$ in each category are detailed in Section 5.3.

2.3 Adversarial Attacks

Adversarial examples are inputs that are deliberately perturbed to remain indistinguishable to humans but are designed to cause misclassification in an ML model. These perturbations, often in the form of carefully crafted "noise", exploit vulnerabilities in the model's decision-making process. The Fast Gradient Sign Method (FGSM) is a fundamental adversarial attack technique to deceive DL models [17], initially designed for classification models. Mathematically, FGSM perturbs an input data point ($\mathbf{x}$) by adding a small perturbation (ϵ) in the direction of the sign of the gradient of the model's loss ($\mathcal{L}$) with respect to the input. The objective is to maximize the loss, leading to misclassification by the model. The FGSM attack against a classification model can be expressed as:

$$\mathbf{x_{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(\mathcal{M}(\mathbf{x}), y)) \tag{3}$$

Here, $\mathbf{x_{adv}}$ represents the adversarial example, $\mathcal{M}(\mathbf{x})$ is the classification model's prediction on input $\mathbf{x}$, y is the actual label, $\nabla_x \mathcal{L}$ denotes the gradient of the model's loss with respect to the $\mathbf{x}$ and ϵ controls the magnitude of the perturbation.

Anomaly detectors, primarily based on unsupervised DL techniques, assign anomaly scores to data points based on their deviation from normal patterns. Hence, FGSM can be extended to generate adversarial examples for anomaly detectors, focusing on manipulating the anomaly scores output by these models [28]. The FGSM attack against an anomaly detection model can be expressed as:

$$\mathbf{x_{adv}} = \mathbf{x} \pm \epsilon \cdot \text{sign}(\nabla_x \mathcal{A}(\mathbf{x})) \tag{4}$$

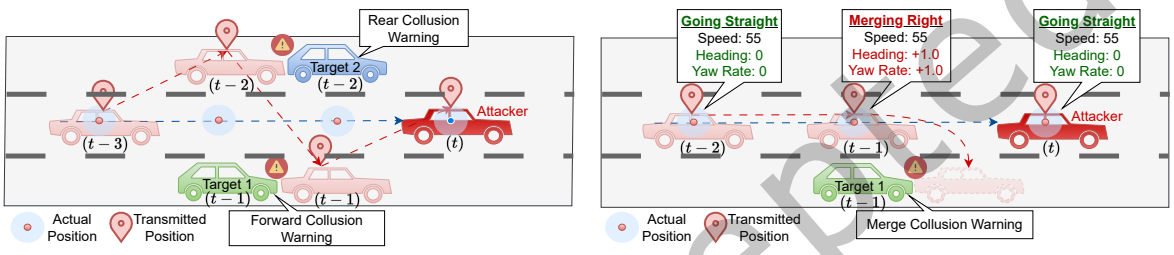
Here $\mathcal{A}(\mathbf{x})$ is the anomaly detection model's prediction on input, which indicates the anomaly score. The goal is to manipulate (increase/decrease) the anomaly score, potentially leading to misclassifications by causing normal instances to be mislabeled as anomalies or vice versa. In Section 3, we extend and elaborate these adversarial attacks to GAN-based MBDS.

3 PROBLEM FORMULATION AND THREAT MODEL

We formulate the design of a robust misbehavior detection problem considering an advanced adversary that can craft malicious BSM fields and fine-tune that malicious data by adding a noise-like perturbation to remain stealthy from being detected. This section further formalizes the generation steps for adversarially stealthy misbehavior, explains the threat model, and finally, the defense objectives.

Table 1. Attack matrix with attack type and targeted fields.

Attack Type	Value(s) of targeted field(s)	Targeted Field(s)					
		Position	Speed	Acceleration	Heading	Yaw Rate	Heading & Yaw Rate
Random	Random value	❶	❺	⓫	⓱	㉔	㉚
Random Offset	Value with random offset	❷	❻	⓬	⓲	㉕	㉛
Constant	Constant value	❸	❼	⓭	⓳	㉖	㉜
Constant Offset	Value with constant offset	❹	❽	⓮	⓴	㉗	㉝
High	Significantly high value		❾	⓯		㉘	㉞
Low	Significantly low value		❿	⓰		㉙	㉟
Opposite	Opposite to the original heading				㉑		
Perpendicular	Perpendicular to the original heading				㉒		
Rotating	Rotating heading over time				㉓		



(a) An illustration of random position attack.

(b) An illustration of high heading & yaw rate attack

Fig. 3. An illustration of two types of misbehaviors with diverse intentions in V2X space. In (a), while going straight, the attacker vehicle transmits fake random positions to influence the decision of nearby benign target vehicles. In (b), the attacker vehicle transmits fake high heading & yaw rate to stage a potential right turn and, thus, a collision scenario with the target.

3.1 Attack Goals: Adversarially Stealthy Misbehavior

Fig. 2 shows the two steps of generating adversarially stealthy malicious BSMs by an insider attacker. The attacker has two primary objectives, which are executed in the following two consecutive steps.

3.1.1 Inducing Misbehavior in BSM Data. The attacker's primary goal is to alter the original BSM data, $\mathbf{x}_{org} = [x^1_{org}, x^2_{org}, x^3_{org}, \ldots, x^f_{org}]$, which includes key f different telematic fields such as location, speed, acceleration, direction, etc. This enables the attacker to mislead other vehicles or infrastructure nodes, potentially creating unsafe conditions. To achieve this, the attacker injects a malicious perturbation $\mathbf{x}_{mal} = [\delta x^1_{mal}, \delta x^2_{mal}, \delta x^3_{mal}, \ldots, \delta x^f_{mal}]$ into the original data, leading to a malicious BMS containing $\mathbf{x}_{mal} = \mathbf{x}_{org} + \mathbf{x}_{mal}$. Here each element (δx^i_{mal}) depends on the type of attacks and represents the degree of malicious modification applied to field i, where $(\delta x^i_{mal} = 0)$ indicates that field i is not targeted by the attacker and remains in its original state.

Table 1 summarizes the overall threat landscape outlining various attack types and possible functions to calculate $\delta\mathbf{x}_{mal}$, targeted field(s), and the description of the value transmitted in the targeted field(s). The circle ❶ indicates the target field(s) of each type of attack, and the number within it denotes the attack index. For example, in the case of a *"Random"* attack, the attacker can transmit random values for either position, speed, acceleration, etc. (as illustrated in Fig. 3(a)). However, in the *Rotating* attack, the attacker only targets the heading as it is the only field that can have meaningful values indicating a rotation. We name each attack based on the attack type and targeted field(s). For example, a *RandomPosition* attack transmits random values in the fields of positional fields; *RotatingHeading* transmits heading data demonstrating that the vehicle is rotating over time.

We assume that to keep attack complexity low, most of the attacks (❶ – ㉙) only compromise a single targeted field, such as position, speed, acceleration, heading, or yaw rate, and do not account for the change on the other correlated non-targeted fields. Furthermore, we consider a set of advanced attacks (㉚ – ㉟) when the attacker compromises both the heading & yaw rate, as illustrated in Fig. 3(b), and modifies these two fields together, coherently, following their inter-dependency. We use this threat matrix in Section 5.1 to generate a misbehavior dataset and evaluate VehiGAN's performance.

3.1.2 Evasion of Detection Mechanisms. The attacker also aims to evade existing ML-based MBDS by masking the malicious changes in the BSM fields by adding further adversarial perturbation to the data calculated in the previous steps. Here, adversaries can generate adversarial input by making subtle adjustments to any/all of the BSM fields, following the patterns outlined by the specific attack algorithms. Let us assume that the adversarial perturbation $x_{adv} = [\delta x^1_{adv}, \delta x^2_{adv}, \delta x^3_{adv}, \ldots, \delta x^f_{adv}]$ with much smaller in magnitude than the malicious one ($\|\Delta x_{adv}\| \ll \|x_{mal}\|$), is designed to mimic the patterns of random noise with negligible strengths but has the potential to change the decision of the ML-based MBDS. Thus, the final attacked telematics data vector transmitted becomes $\mathbf{x}_{adv} = \mathbf{x}_{org} + x_{mal} + x_{adv}$. Here, the low magnitude of $\|\Delta x_{adv}\|$ ensures that $\mathbf{x}_{adv}$ still preserves the malicious properties and the adversarial perturbation is just working as a mask for the attack. While adversarial attack algorithms were initially developed for classification-based models [24], there have been very few efforts to extend these techniques to anomaly detection models [28]. Moreover, to date, there is a lack of research targeting adversarial attacks against the discriminators of Wasserstein GANs (WGANs) or in the context of V2X MBDS. Given that there are eventually two possible outcomes from the discriminators — benign or anomaly — we categorize the adversarial attacks against any MBDS into two types:

Adversarial False Positive (AFP) Attack. An APF attack on MBDS involves manipulating a benign (original) input to deceive the model into outputting an anomaly score high enough to be flagged as a misbehavior (false positive). An AFP attack happens when $\Delta\mathbf{x}_{mal} = 0$ and thus, the adversarial perturbation $\Delta\mathbf{x}^{AFP}_{adv}$, is added directly to $\mathbf{x}_{org}$ and calculated using the gradient that maximizes the anomaly score. Thus, the adversarial input under AFP attacks $\mathbf{x}^{AFP}_{adv} = \mathbf{x}_{org} + \Delta\mathbf{x}^{AFP}_{adv}$ and combining (4) and (2), we get the following:

$$\mathbf{x}^{AFP}_{adv} = \mathbf{x}_{org} + \epsilon \cdot \text{sign}(\nabla_x \mathcal{A}(\mathbf{x}_{org})) \quad \& \quad \mathbf{x}^{AFP}_{adv} = \mathbf{x}_{org} - \epsilon \cdot \text{sign}(\nabla_x \mathcal{D}(\mathbf{x}_{org})) \tag{5}$$

Under the AFP attacks, the attacker crafts an adversarial input $\mathbf{x}^{AFP}_{adv}$ by applying a small, targeted perturbation to the original input $\mathbf{x}_{org}$. The attacker's goal is to maximize the anomaly score $\mathcal{A}(\mathbf{x}_{org})$ or minimize the output of the discriminator $\mathcal{D}(\mathbf{x}_{org})$, thereby increasing the likelihood of it being flagged as an anomaly. The goal of the AFP attacks is to increase the FP rates, thereby flooding the system with false alarms, potentially overwhelming the MBDS and creating a denial-of-service condition.

Adversarial False Negative (AFN) Attack. On the other hand, an AFN attack involves manipulating a misbehavior (positive) input to deceive the model into outputting an anomaly score low enough to be determined as a benign (false negative) one. An AFN attack occurs when $\Delta\mathbf{x}_{mal} \neq 0$, resulting in the adversarial input under AFN attack as $\mathbf{x}^{AFN}_{adv} = \mathbf{x}_{mal} + \Delta\mathbf{x}^{AFN}_{adv}$. This can be expressed as:

$$\mathbf{x}^{AFN}_{adv} = \mathbf{x}_{mal} - \epsilon \cdot \text{sign}(\nabla_x \mathcal{A}(\mathbf{x}_{mal})) \quad \& \quad \mathbf{x}^{AFN}_{adv} = \mathbf{x}_{mal} + \epsilon \cdot \text{sign}(\nabla_x \mathcal{D}(\mathbf{x}_{mal})) \tag{6}$$

The attacker aims to reduce the anomaly score $\mathcal{A}(\mathbf{x}_{mal})$ or maximize the output of the discriminator $\mathcal{D}(\mathbf{x}_{mal})$, making the malicious input appear benign. This subtle modification reduces the likelihood of the input being flagged as an anomaly by shifting it closer to the distribution of normal data. In summary, the primary objective of AFN attacks is to raise the FN rate, thereby masking malicious alterations and enabling the attacker to successfully transmit misbehaving data without detection.

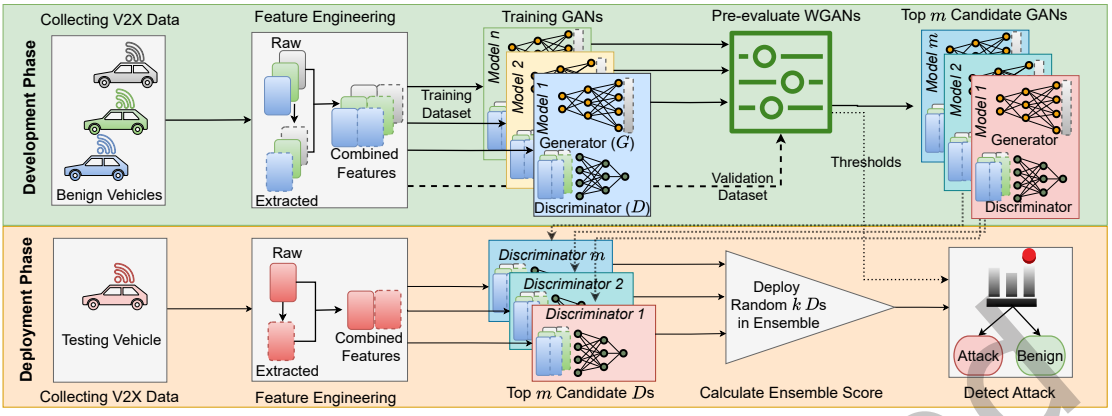


Fig. 4. Workflow of VehiGAN, which has two phases of operation: i) development and ii) deployment phase.

3.2 Attackers Capabilities

We focus on the *insider* and *active* attackers within the V2X network who are authenticated members with legitimate cryptographic credentials and actively engaged in malicious actions, as considered in [3, 33]. To fulfill their objectives, the attacker further possesses the following capabilities:

- *Payload Modification.* They are *local* parties who can capture, modify, and transmit the payload of BSMs by accessing the data from the sensors and changing the targeted fields before transmission.
- *Knowledge of MBDS.* Based on the extent of access/knowledge of the MBDS, we categorize the adversarial attacks in two types, namely, white-box and black-box attacks [57]. In white-box adversarial attacks, the attacker possesses complete knowledge of the detection mechanism, as well as the parameters and gradients of all the WGAN model(s) employed. In black-box attacks, adversaries lack direct access to the model's parameters and gradients. Hence, they employ transfer attacks, generating adversarial samples using a surrogate WGAN model and deploying them against the target model(s).
- *Precise Synchronization and Timing Control.* The attacker can synchronize the application of $\Delta\mathbf{x}_{mal}$ and $\Delta\mathbf{x}_{adv}$ with the BSM data transmission intervals of $100ms$, ensuring that the data alterations remain consistent with the V2X system's timing expectations.

3.3 Defense Goal and Capabilities

The defender has two objectives: i) detect any misbehavior by analyzing BSMs, and ii) remain robust against adversarial perturbations. We assume that the defender has no information on what type of misbehavior and type of adversarial attacks the attacker is utilizing to achieve their goals. Further, the defender has a training dataset $\mathcal{S}^{bsm}_{train} = \{\mathcal{X}^{bsm}_{train}, y^{bsm}_{train}\}$ that contains benign BSMs, i.e., $y^{bsm}_{train} = \{0\}$, from normal driving events, which the defender utilizes to train the MBDS. Moreover, the defender possesses a small validation dataset $\mathcal{S}^{bsm}_{valid} = \{\mathcal{X}^{bsm}_{valid}, y^{bsm}_{valid}\}$, and we consider two types of defenders based on the type of validation dataset. One type of defender possesses $\mathcal{S}^{bsm}_{valid}$ that contains both benign and malicious BMSs , i.e., $y^{bsm}_{valid} = \{0, 1\}$, to pre-evaluate the trained MBDS. Other defenders lack access to malicious data and consider only benign BSMs as $\mathcal{S}^{bsm}_{valid}$, where $y^{bsm}_{valid} = \{0\}$. Alternatively, they may use $\mathcal{S}^{bsm}_{valid}$ as a subset of $\mathcal{S}^{bsm}_{train}$ for preliminary evaluation. In the following section, we show how we design an MBDS that achieves both objectives for each defender type.

4 VEHIGAN: GAN-BASED MBDS

This section first describes an overview of the VehiGAN architecture, followed by the details of each part.

4.1 VehiGAN Overview

Fig. 4 shows the workflow of VehiGAN, its two phases (development and deployment), and its different components. The development phase has five core tasks: i) collecting V2X data, ii) feature engineering, iii) WGAN training, iv) pre-evaluating WGANs, and v) selecting top WGAN candidates. The deployment phase has similar tasks, but instead of WGAN training, it deploys a subset of candidate WGAN models for ensembling, runs the inference on the collected data, and takes action based on the output. The central element of VehiGAN is a software system designed to gather and analyze BSMs from nearby vehicles in near real-time. It can be implemented both in the onboard units (OBU) of the individual ego vehicles for self-defense or in the roadside units (RSU) by local authorities.

Development Phase. The top panel of Fig. 4 shows the development phase of VehiGAN. In the first step, VehiGAN collects BSMs from trusted participating vehicles. Such trusted participant vehicles can be pre-selected by the V2X authority to ensure the reliability of the collected data for future MBDS training. On the other hand, there are traffic simulators, such as Veins [50], that can generate BSMs resembling real-world traffic mobility. Once sufficient data is collected to generalize the traffic behaviors and mobility, VehiGAN initiates the feature engineering tasks. VehiGAN extracts new features from the transmitted raw fields from the BSMs and extends the dataset by combining all the features altogether, effectively creating multi-dimensional time-series telematics data.

There exists an inherent complexity in finding the optimal architecture of any DL model, and the most prevailing approach is to perform a grid search. VehiGAN employs the same strategy and trains different WGAN models with varying architectures and hyperparameters on the combined dataset. After training all the models, VehiGAN starts pre-evaluating the performance of the WGAN models using a validation dataset. If the validation dataset contains only benign traces, the pre-evaluation happens by evaluating the generators on the WGANs; otherwise, the dataset contains some attack traces as well, and then the discriminator is directly evaluated instead. After the pre-evaluation, instead of selecting the single best-performing discriminator for MBDS, VehiGAN shortlists the top-performing m candidate discriminators for the ensemble during the deployment phase. Finally, VehiGAN calculates the anomaly scores threshold for each of the top m discriminators using the validation dataset, which will be used during the deployment phase.

Deployment Phase. The bottom panel of Fig. 4 shows the deployment phase of VehiGAN, which is completely executed locally on the OBU/RSU. Similar to the development phase, VehiGAN keeps collecting raw BSMs from individual testing vehicles, runs the feature engineering task, and creates an effective representation. Later, instead of using all the m top-performing discriminators, VehiGAN randomly selects k discriminators, where $k \leq m$ from the m top candidates and ensembles them for misbehavior detection. We define such detector as VehiGAN^k_m that predicts the misbehavior scores with different random k discriminator every time. If the anomaly score for any vehicle surpasses a predetermined threshold, which is the average threshold of the deployed k discriminator, VehiGAN reports that to MA as potential misbehavior. The following subsections explain the details of each part of VehiGAN in each phase.

4.2 Collecting Raw V2X Data

Throughout both the training and deployment phases, VehiGAN gathers BSMs from nearby vehicles. VehiGAN places particular emphasis on BSM's core features that are important for the V2X applications. VehiGAN categorizes the entire dataset into multiple groups based on the vehicle id, v, where each of these groups contains continuous time series data for a specific vehicle. Whereas VehiGAN keeps all BMSs of individual vehicles in the

Table 2. Feature engineering to extract highly correlated features from the raw features.

Type	Raw Feats	Decomposed Features		Relation		Delta Features	
		X Comp	Y Comp	X Comp	Y Comp	X Comp	Y Comp
Position	x, y	x	y	–	–	$\Delta x = x(t+1) - x(t)$	$\Delta y = y(t+1) - y(t)$
Speed	$\mathbf{v}$	$v_x = \mathbf{v} \times cos(\theta)$	$v_y = \mathbf{v} \times sin(\theta)$	$\Delta x = v_x \times \Delta t$	$\Delta y = v_y \times \Delta t$	$\Delta v_x = v_x(t+1) - v_x(t)$	$\Delta v_y = v_y(t+1) - v_y(t)$
Acceleration	$\mathbf{a}$	$a_x = \mathbf{a} \times cos(\theta)$	$a_y = \mathbf{a} \times sin(\theta)$	$\Delta v_x = a_x \times \Delta t$	$\Delta v_y = a_y \times \Delta t$	–	–
Heading	θ	$\theta_x = \mathbf{1} \times cos(\theta)$	$\theta_y = \mathbf{1} \times sin(\theta)$	–	–	$\Delta\theta_x = \theta_x(t+1) - \theta_x(t)$	$\Delta\theta_y = \theta_y(t+1) - \theta_y(t)$
Yaw Rate	ω	$\omega_x = \omega \times cos(\theta)$	$\omega_y = \omega \times sin(\theta)$	$\Delta\theta_x = \omega_x \times \Delta t$	$\Delta\theta_y = \omega_y \times \Delta t$	–	–

development phase, it keeps only the latest messages that are sufficient to run the inference in the deployment phase. Before engaging in training or running inference with the raw features, VehiGAN performs essential feature engineering, as detailed in the subsequent section.

4.3 Feature Engineering

VehiGAN leverages domain expertise related to classical physics to conduct vector decomposition of raw features to extract new correlated features. For instance, when considering the scalar values of speed and acceleration, there is no direct correlation. However, upon vector decomposition into their respective X and Y components, it becomes evident that changes in speed exhibit a high correlation with acceleration for each component. This feature extraction capability empowers VehiGAN to create consistent, new features from raw data attributes, ultimately facilitating the development of a robust MBDS. Table 2 provides an overview of how VehiGAN performs vector decomposition into X and Y components, represented by subscripts $(_x)$ and $(_y)$, respectively, for various raw features. VehiGAN further computes the changes between them in the consecutive time steps, defined as delta (Δ) features. The table illustrates the interrelationships among the extracted features and delta features. The combined features, that need to be secured, may contain both the raw and extracted features (as illustrated in Fig. 4). However, the current implementation of VehiGAN only considers the extracted features as combined features for the defense, which can be easily extended by adding more raw features.

To train the WGAN models, VehiGAN takes the pre-selected core feature set F as $\{\Delta x, \Delta y, v_x, v_y, \Delta v_x, \Delta v_y, a_x, a_y, \Delta\theta_x, \Delta\theta_y, w_x, w_y\}$ and generates numerous 2D snapshots $\mathbf{x}_v^{bsm} \in \mathbb{R}^{w \times f}$ from the time series data of vehicle v. This is achieved using a moving window of size w, and the length of the selected feature set is f. These snapshots are aggregated to form the training dataset $\mathcal{X}_{train}^{bsm} \in \mathbb{R}^{n \times w \times f}$, where n represents the total number of snapshots across all vehicular groups. Fig. 5 shows such steps of generating $\mathcal{X}_{train}^{bsm}$ from consecutive BSMs. Such snapshots encapsulate both temporal patterns of various vehicles and feature-wise relationships. On the other hand, to create testing data to check for MBDS using the trained and ensembled WGAN models, VehiGAN only keeps a single 2D snapshot $\mathbf{x}_v^{bsm} \in \mathbb{R}^{w \times f}$ of time series data from the most recent w BSMs for every vehicle v. Every time a new message comes from the vehicle v, its corresponding $\mathbf{x}_v^{bsm}$ gets updated.

4.4 Model Training

To find the best-performing WGAN models, VehiGAN explores a wide range of model architectures and hyperparameters. Each configuration is designed to experiment with different hyperparameters and architectural choices for both $\mathcal{G}$ and $\mathcal{D}$ models. For every configuration, VehiGAN initializes the models with slightly different architectures and sets their respective hyperparameters, such as training epochs, to find potential candidates for the best models. To adapt the WGAN model to the multi-dimensional time series data, we use a 2D convolutional neural network (CNN) in both $\mathcal{G}$ and $\mathcal{D}$. While $\mathcal{G}$ converts a 1D noise vector $\mathbf{z} \in \mathbb{R}^d$ into a 2D snapshots $\mathbf{x_{fake}} \in \mathbb{R}^{w \times f}$, $\mathcal{D}$ takes the real or fake snapshots $\mathbf{x_{real}}$ or $\mathbf{x_{fake}} \in \mathbb{R}^{w \times f}$ as inputs and outputs a scalar value that represents the likelihood of the input being real. Upon completing the training on X_{train}^{bsm}, VehiGAN stores model checkpoints and relevant training statistics for further processing.

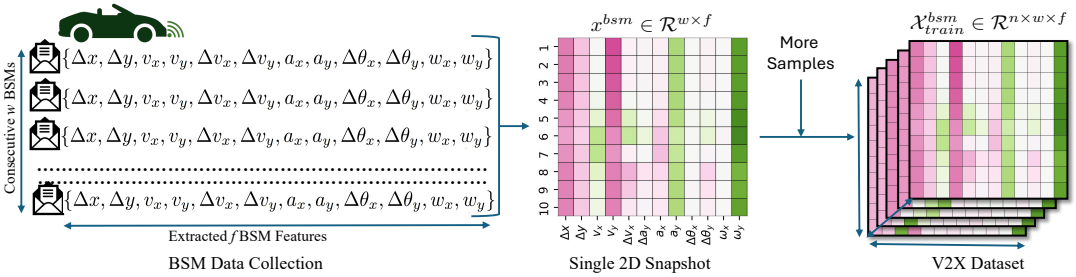


Fig. 5. Steps for creating a dataset for V2X MBDS: Consecutive BSM features are extracted and represented as 2D images, with the MBDS dataset forming as a series of these images over time.

4.5 Model Pre-evaluation and Selection

To determine the top m WGAN models as the candidate for the ensemble, VehiGAN runs the pre-evaluation on a validation dataset $\mathcal{S}^{bsm}_{valid} = \{\mathcal{X}^{bsm}_{valid}, y^{bsm}_{valid}\}$. Based on the attacker's capabilities on the validation dataset $\mathcal{S}^{bsm}_{valid}$, we consider two following approaches:

4.5.1 Generator-based Pre-evaluation and Selection. If $\mathcal{S}^{bsm}_{valid}$ contains only benign traces (i.e., $y^{bsm}_{valid} = 0$ for all samples), where it may even be a subset of the training dataset $\mathcal{S}^{bsm}_{train}$ (i.e., $\mathcal{S}^{bsm}_{valid} \subseteq \mathcal{S}^{bsm}_{train}$), the defender cannot directly evaluate the performance of the discriminators but rather evaluate the generators and select the discriminators that are trained with the m top-performing generators. We define this process as generator-based model selection. As the generator and discriminator are trained together, we hypothesize that there could be a correlation between their performances in their individual tasks. For the pre-evaluation, each $\mathcal{G}$ is used to synthesize representative fake snapshots $\mathbf{X}^{bsm}_{fake} = \{\mathbf{x}^1_{fake}, \mathbf{x}^2_{fake}, \cdots \mathbf{x}^n_{fake}\}$ while the same amount of real snapshots $\mathbf{X}^{bsm}_{real} = \{\mathbf{x}^1_{real}, \mathbf{x}^2_{real}, \cdots, \mathbf{x}^n_{real}\} \subseteq \mathbf{X}^{bsm}_{valid}$ is sampled from the validation dataset. Later, we calculate the discrepancy between $\mathbf{X}^{bsm}_{real}$ and $\mathbf{X}^{bsm}_{fake}$ using any suitable metrics (as defined in Section 2.2 and 5.3), and consider that as that generative score (*GS*) of that $\mathcal{G}$. A lower *GS* indicates less distance between the two distributions and, hence, more realistic fake data. Therefore, the top m WGAN models with the lowest *GS* are selected for the ensemble model, and the ensemble model is defined as G-based VehiGAN^k_m.

4.5.2 Discriminator-based Selection. On the other hand, when $\mathcal{S}^{bsm}_{valid}$ contains both benign and representative attack traces, i.e., $y^{bsm}_{train} = \{0, 1\}$, where 0 represents benign traces and 1 represents attack traces, the defender can directly evaluate the discriminators and select the top m candidate ones. We define discriminative score (*DS*) as the average detection score of $\mathcal{D}$ over all the samples in the validation dataset. The *DS* can be any commonly used metrics used to evaluate a classifier, such as AUROC, AUPRC, etc. (explained in Section 5.3). A higher score indicates that a certain $\mathcal{D}$ is more likely to be effective against any unseen misbehavior in the test data. Consequently, the top m WGAN models with the highest *DS* are selected as the candidates for the ensemble model, and the ensemble model is defined as $\mathcal{D}$-based VehiGAN^k_m.

4.6 Threshold Selection and Attack Detection

As the discriminator of a WGAN is designed to output higher values for benign inputs, we take the negative of that value as an anomaly score to generalize the misbehavior detection process. Hence, the benign anomaly scores of any model on all the snapshots in $\mathbf{X}^{bsm}_{train}$ are calculated as $\mathbf{a}_i = -\mathcal{D}_i(\mathbf{X}^{bsm}_{train})$. The detection threshold τ_i for each of the individual discriminators is calculated based on the p-th percentile of that benign anomaly sore $\mathbf{a}_i$ where p is a system parameter (usually 99 to 99.99). Although there are m candidate discriminators, VehiGAN constructs

the final ensemble detector during inference by randomly selecting k discriminators from the m candidates and averaging their anomaly scores to obtain the final anomaly score. Thus, the ensemble discriminator $\mathcal{D}_{\text{ens}}(.)$ is defined as:

$$\mathcal{D}_{\text{ens}}(.) = \frac{1}{k}\sum_{i \in C} \mathcal{D}_i(.) \quad \text{where } C \subset \{1, \ldots, m\} \text{ and } |C| = k. \tag{7}$$

Here C is a subset of m candidates containing exactly k elements. Similarly, the detection threshold is also dynamically calculated as $\tau_{ens} = \frac{1}{k}\sum_{i \in C} \tau_i$ using the same set C. Thus, during the deployment phase, the anomaly score of the most recent w BSMs transmitted by the target vehicle v is calculated using $a_v = -\mathcal{D}_{ens}(\mathbf{x}_v^{bsm})$, and the detection threshold τ_{ens} is used to check if vehicle v is misbehaving. A value of $\mathbf{a}_v > \tau_{ens}$ indicates the existence of misbehavior, and VehiGAN immediately creates an MBR on vehicle v, including the corresponding BSMs, and sends it to the MA.

The benefits of selecting $k \leq m$ are twofold. It reduces the computational overhead to $(m - k/m)$ times with negligible performance drops. Second, selecting k random $\mathcal{D}$s in every inference increases uncertainty, which hardens the generation of adversarial samples.

5 IMPLEMENTATION

5.1 Dataset

We implement VehiGAN on the V2X misbehavior dataset simulated using VASP [3], an open-source framework. VASP allows the simulation of diverse types of V2X attacks and works as a sub-module for Veins[50], a well-established open-source framework for running vehicular network simulations. Veins further runs on an event-based network simulator OMNeT++[56], and road traffic simulator SUMO [30]. We ran a VASP simulation on the Boston traffic network for 3,000 simulated seconds to collect benign traces without any attacks. Such simulation provided us with 1, 018, 098 benign BSMs from 475 different vehicles. Similarly, we ran another VASP simulation for 1360 simulated seconds to collect malicious traces with 68 distinct attacks, out of which we used 35 of them for our evaluation, resulting in a dataset of 2, 641, 309 BSMs. It is noted that the remaining 33 attacks fall outside the scope of our threat model. Nonetheless, we have published the complete dataset, titled MisbehaviorX [42], to facilitate future research endeavors. We consider 25% of the vehicles to be malicious during each simulation. While running the attack, we selected the attack policy as *persistent*, where the attacker vehicle always transmits attack messages. In the evaluation, we create the 2D snapshots for the evaluation of VehiGAN by considering approximately 1 sec of telematics or 10 consecutive BSMs, hence a sliding window size w as 10, and a number of features f as 12 as listed in Fig 5.

5.2 Model Architecture.

We train a diverse set of WGAN models based on predefined hyperparameters, carefully selecting them to balance model complexity and training efficiency. Specifically, we vary the noise vector dimension z across {8, 16, 32, 48, 64} to explore its impact on the quality and diversity of generated samples, with larger dimensions providing richer latent representations at the cost of increased computational demand. The number of layers in $\mathcal{G}$ and $\mathcal{D}$ networks is varied within {6, 7, 8}, allowing us to assess the trade-off between model expressiveness and overfitting risks. Additionally, we train models for {25, 50, 75, 100} epochs to examine convergence behavior and mitigate under- or overfitting. This approach results in a total of 60 WGAN model checkpoints, enabling a comprehensive evaluation of architectural choices. During training, we set a batch size of 128 and a learning rate of 1×10^{-3}. In $\mathcal{G}$'s 2D up-sampling layers and $\mathcal{D}$'s 2D convolution layers, we use 2x2 filters with the LeakyReLU activation function, enhancing model stability and performance.

5.3 Evaluation Metrics

We consider the following metrics for $\mathcal{G}$ and $\mathcal{D}$ to evaluate the individual WGAN models as well as VehiGAN.

5.3.1 Evaluation Metrics for the Generator. We outlined various types of evaluation metrics for assessing the performance of the $\mathcal{G}$ in Section 2.2.1. Below, we define different metrics used within their respective categories.

Distribution-based Evaluation. We consider the following distribution-based evaluation metrics. Maximum Mean Discrepancy (*MMD*) [6, 18] uses a kernel-based approach and measures the difference between two distributions by comparing their mean embeddings in a high-dimensional space. Fréchet Distance (*FD*) [11] compares two distributions by calculating the distance between their mean and covariance matrices. Kolmogorov-Smirnov (*KS*) Test [15, 48] shows whether two distributions are the same by measuring the maximum difference between their cumulative distribution functions. Anderson-Darling (*AD*) test [2] assesses whether multiple samples come from the same distribution, placing additional weight on the tails of the distributions. On the other hand, the earth mover distance (*EMD*) [38] measures the minimum cost to transform one distribution into another, which captures differences in both shape and location without relying on kernel functions.

Model-based Evaluation. We consider the following model-based evaluation metrics. Train on Synthetic, Test on Real (*TSTR*) [14] evaluates the quality of synthetic data by training a model on synthetic data and then testing it on real data. The model's performance on real data indicates the coverage of the synthetic data compared to the real data. Contrarily, Train on Real, Test on Synthetic (*TRTS*) [14] considers training a model on real data and testing it on synthetic data, where testing performance indicates whether the synthetic data falls within the distribution of the real data. In both cases, we train an autoencoder model and consider the mean absolute reconstruction error as the performance metric.

Computer Vision-based Evaluation. We consider the following computer vision-based evaluation metrics that use Inception V3 [52] to extract the 64-dimensional feature vector. Later, Fréchet Inception Distance (*FID)* [25] uses the Fréchet distance to compare the Gaussian-based means and covariances of feature distributions from real and generated data. On the other hand, Kernel Inception Distance (*KID*) [5] uses MMD with a polynomial kernel to compare feature distributions without assuming Gaussianity.

5.3.2 Evaluation Metrics for the Discriminator. The discriminator can produce four distinct outcomes. True Positive (TP) and True Negative (TN) occur when the model accurately predicts an input as misbehavior and benign behavior, respectively. On the other hand, False Positive (FP) and False Negative (FN) happen when the model incorrectly predicts an input as misbehavior and benign behavior. We evaluate the discriminator's performance based on these outcomes using the following metrics:

- True Positive Rate (TPR) is the proportion of total positive instances correctly identified as positives ($\frac{TP}{TP+FN}$).
- False Positive Rate (FPR) is the proportion of negative instances incorrectly identified as positives ($\frac{FP}{FP+TN}$).
- False Negative Rate (FNR) is the proportion of positive instances incorrectly identified as negatives ($\frac{FN}{TP+FN}$).
- ROC Curve indicates the classifiers performance with varying discrimination threshold [22]. The ROC curve plots TPRs and FPRs for different thresholds. The area under the ROC curve (AUROC) indicates the robustness of the detectors against both benign and misbehavior instances.

5.4 Baseline Models

To compare the performance of VehiGAN with the existing baselines, we consider the following anomaly detection methods:

5.4.1 Linear Models for Outlier Detection. Such models assume that the normal data points in the dataset can be well-described by linear relationships, and outliers are data points that significantly deviate from this linearity.

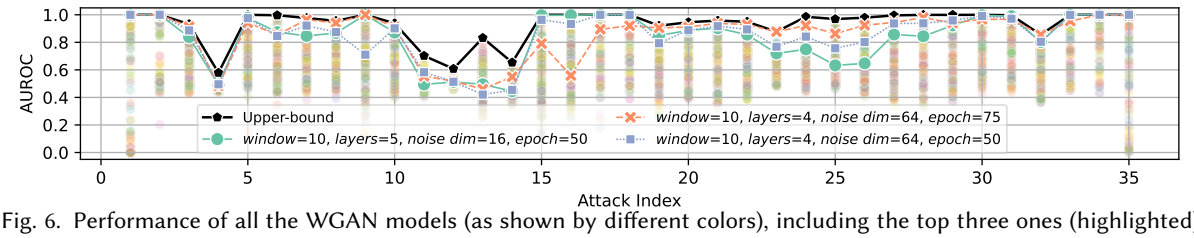


Fig. 6. Performance of all the WGAN models (as shown by different colors), including the top three ones (highlighted), against individual attacks. Different WGAN models perform differently against the same attack, indicating no single WGAN can achieve stable and robust performance.

For instance, *Principal Component Analysis* (PCA) uses the sum of weighted projected distances to the eigenvector hyperplane as the outlier scores [46].

5.4.2 Proximity-based Outlier Detection. Such methods, also known as distance-based outlier detection models, assume outliers are significantly different (far) from the benign data points in the dataset. For instance, *k-Nearest Neighbors* (KNN) assigns each data point an outlier score based on the distance to its k-nearest neighbors [37].

5.4.3 Probabilistic Models for Outlier Detection. Such methods model the data distribution and assess the likelihood of each data point under that distribution with the assumption that outliers are generated from a less probable distribution. For example, *Gaussian Mixture Models* (GMM) is a probabilistic model where outliers have a low probability of being generated by any of the mixtures of several Gaussian distributions [27].

5.4.4 DL Models for Outlier Detection. DL models are well-suited for identifying stealthy and complex anomalies within large datasets by learning intricate data distributions. While our proposed system, VehiGAN, belongs to this category, we also use CNN-based *Autoencoders* (AE) as DL baselines for comparison in this study [26]. AEs are neural network architectures commonly employed for tasks such as anomaly detection, where they learn to reconstruct input data and flag poorly reconstructed data points as outliers. In this study, we train the AE baseline on raw feature data, referring to this model as BaseAE.

Moreover, to show the contribution of the featured engineering of VehiGAN, we also evaluate all the baselines on the VehiGAN extracted features and name them with the prefix Vehi- as mentioned in Table 3.

6 RESULTS

We evaluate the effectiveness of VehiGAN from different perspectives. First, we analyze the misbehavior detection performance of both individual WGAN (VehiGAN_1^1) and ensemble-based VehiGAN_m^k with k deployed models out of m candidate ones against different misbehaviors. For the VehiGAN_m^k model, we start with $\mathcal{D}$-based model selection and later contrast it with the performance with $\mathcal{G}$-based model selection. Later, we conduct an extensive robustness analysis of VehiGAN_m^k against different adversarial attack settings. Finally, we compare the performance of two representative VehiGAN_m^k models with other baseline models, which is followed by a scalability analysis.

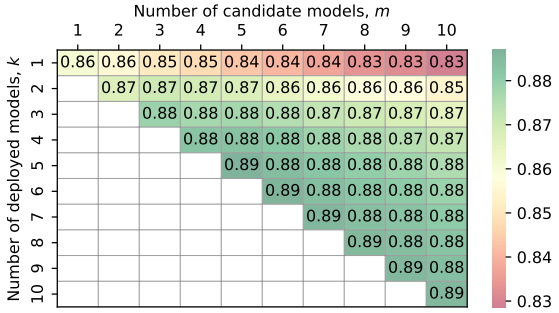


Fig. 7. Average AUROC of VehiGAN_m^k, where $m \geq 5$ with $k \geq \frac{m}{2}$ leads to higher AUROC scores.

6.1 Misbehavior Detection

6.1.1 Performance of Single WGAN-based VehiGAN$_1^1$. Fig. 6 provides a comprehensive assessment of all the 60 trained WGAN instances, particularly the discriminators with respect to the AUROC scores, against all the 35 attacks considered in the evaluation. Here, different color indicates different discriminators for which we skipped the legend as there are 60 of them. However, we highlight the AUROC scores for the top three discriminators that provided the highest average AUROC scores, along with the upper bound, the maximum achievable performance by any individual discriminators, across attacks. According to the figure, different discriminators performed differently against the same attack. Even the top three discriminators, despite having the highest AUROC scores, failed to detect certain attacks effectively. This implies that it is challenging to train a single WGAN model capable of providing a comprehensive solution to all types of misbehaviors.

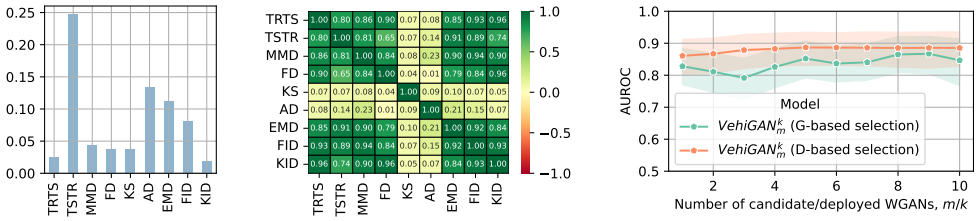
6.1.2 Performance of Ensemble-based VehiGAN$_m^k$.

$\mathcal{D}$-based Model Selection. We now evaluate an ensemble-based VehiGAN$_m^k$ to check if combining the top-performing WGAN models harnesses the strengths of each model while mitigating their weaknesses. Fig. 7 shows the impact of m and k on the average AUROC scores against all the 35 attacks in the ensemble-based VehiGAN$_m^k$. We observe that adding more discriminators (higher m and k) mostly leads to higher AUROC scores. However, the benefits of adding more discriminators for VehiGAN tend to plateau after a certain point ($m \geq 5$), indicating that a small number of discriminators, typically 5 to 6, are enough to provide decent AUROC scores. We also notice that k does not necessarily need to be equal to m; even $k > \frac{m}{2}$ leads to consistently elevated AUROC scores.

$\mathcal{G}$-based Model Selection. This part evaluates the $\mathcal{G}$-based model selection for considering various metrics for *GS*. First, we study the correlation between *GS* and *DS* to find the best metric for *GS* (one with the strongest correlation), and then use that metric in the model selection. Fig. 8(a) shows the absolute values of the correlation coefficients of different *GS* metrics with *DS* (calculated using AUROC). It is evident that *TSTR* demonstrates the strongest correlation (≈ 0.25), making it the preferred choice for $\mathcal{G}$-based model selection. Other metrics exhibit negligible correlations, rendering them ineffective. Additionally, Fig. 8(b) presents the mutual correlation coefficients, indicating that most metrics yield similar *GS*, with the exceptions of *KS* and *AD*. Lastly, Fig. 8(c) contrasts the AUROC scores achieved through both $\mathcal{G}$ and $\mathcal{D}$-based selections for VehiGAN$_m^k$ with $m = k$. The results reveal that $\mathcal{D}$-based selection is more stable and effective, while $\mathcal{G}$-based selection shows suboptimal performance (with higher variance), particularly when fewer than five models are included in the ensemble. Thus, we advocate for the use of $\mathcal{D}$-based selection for VehiGAN in the remainder of this study. Additionally, the lack of correlation and poor detection performance indicate that existing *GS* metrics may not adequately assess the performance of G when trained on V2X time-series datasets. This suggests either that these metrics are ineffective for this context or that there may be little to no strong correlation between the performances of $\mathcal{G}$ and $\mathcal{D}$, warranting further investigation.

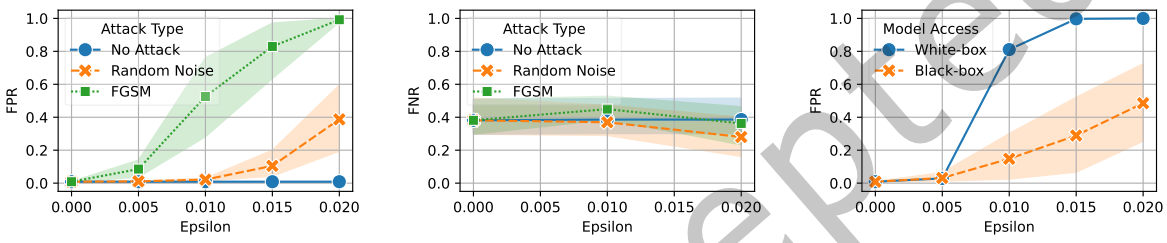
6.2 Adversarial Robustness

To assess the adversarial robustness of VehiGAN, we also consider the top 10 WGAN models based on the highest average AUROC scores across all the misbehaviors in the validation dataset. We first examine the robustness of individual WGAN-based VehiGAN$_1^1$ under both AFP and AFN attacks (as outlined in Section 3.1.2), followed by the robustness of the ensemble-based VehiGAN$_m^k$. For this evaluation, we set a threshold at the 99.0 percentile of benign anomaly scores, ensuring an FPR of less than 1% without adversarial attacks. For this evaluation, we randomly select 100 benign and 100 misbehavior snapshots from each misbehavior type, resulting in a total of 3500 benign and 3500 misbehavior samples. Considering the proposed FGSM attack, as elaborated in Section 3.1.2 , we explore values of ϵ within the range of 0.0 to 0.02 to generate the adversarial samples. To



(a) Correlation with AUROC (b) Self-correlation coefficient matrix (c) Performance of VehiGAN^k_m

Fig. 8. Performance analysis of $\mathcal{G}$-based model selection. (a) The correlation of various metrics (*GS*) with *DS*, where *TSTR* shows the strongest correlation, indicating its effectiveness for $\mathcal{G}$-based model selection; (b) The mutual correlation among the metrics (c) A comparison of AUROC scores illustrating the performance improvements with both $\mathcal{G}$ and $\mathcal{D}$-based selections, where $\mathcal{D}$-based selection demonstrates superior stability and effectiveness.



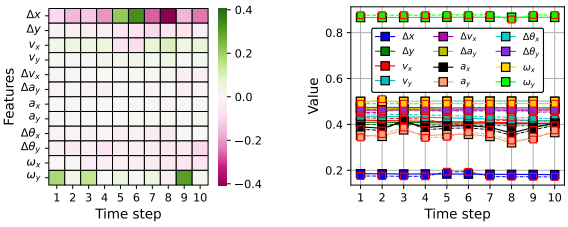
(a) FPR under AFP attacks (b) FNR under AFN attacks (c) Transferability of AFP attacks

Fig. 9. Adversarial robustness of single WGAN-based VehiGAN$^1_1$ under various attacks.

better contrast the impact of such adversarial attacks, we randomize each adversarial perturbation and use it as a random noise baseline to evaluate the model's response under a noisy but benign environment.

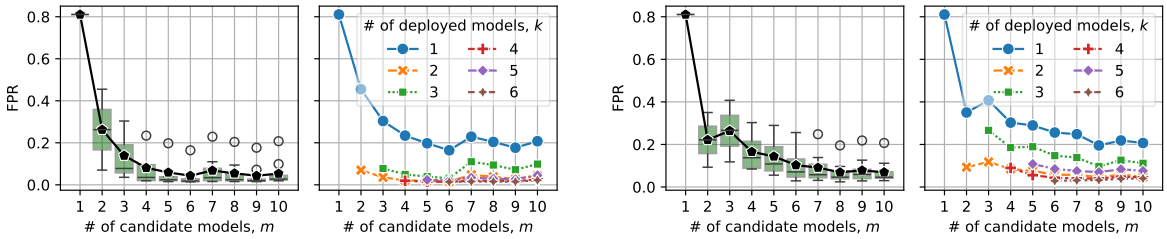
6.2.1 Robustness of Single WGAN-based VehiGAN$^1_1$. Fig. 9(a) illustrates the FPRs of the top 10 single WGAN-based VehiGAN$^1_1$ models under white-box AFP attacks and random noise. With $\epsilon = 0.01$ (*i.e.*, 1% change in the sensor values), such attacks lead to approximately 50% FPR on average. In contrast, random noise with the same strength does not increase the FPR at all. Besides, with approximately only a 2% change in the original values, all benign samples attain anomaly scores greater than the threshold and are labeled as anomalies, resulting in nearly 100% FPR in all the individual models. Random noise at this strength, however, exhibits less than 40% FPR on average. This underscores the vulnerability of single WGAN-based VehiGAN$^1_1$ to white-box AFP attacks. Fig. 10 illustrates an AFP attack with $\epsilon = 0.01$ on a benign input.

On the other hand, Fig. 9(b) demonstrates the FNR of the top 10 single WGAN-based VehiGAN$^1_1$ models under AFN attacks. It is evident from the figure that



(a) Gradient of the loss func (b) Adversarial perturbation

Fig. 10. A representation of the AFP attack on a benign input. (a) shows the gradient of the loss function with respect to the benign input, as outlined in (5). The signs of the gradients at each pixel determine the perturbation (*e.g.*, $\pm\epsilon$). (b) shows both the benign and adversarial inputs. The markers with the black edge (□), and red edge (□) indicate the corresponding original, and adversarial values, respectively, of each sensor at different time steps, which are either increased or decreased by $\epsilon = 0.01$ based on the sign of the gradient.



(a) FPR under gray-box single-model greedy AFP attack

(b) FPR under white-box multi-model adaptive AFP attacks

Fig. 11. Adversarial robustness of ensemble-based VehiGAN^k_m under adaptive APF attacks.

all single WGAN-based VehiGAN models exhibit inherent robustness against AFN attacks. Despite adversarial perturbations aiming to minimize anomaly scores, they push samples beyond the manifold of benign samples, still creating anomalies at the discriminator. As AFN attacks prove ineffective against all single WGAN-based VehiGAN$^1_1$, we exclusively consider AFP attacks in the remainder of this paper, leaving the development of an effective AFN attack as an open research question.

Subsequently, we consider a practical black-box transfer attack wherein the attacker generates adversarial samples using one model and deploys them against others. To study the transferability of AFP attacks against single WGAN-based VehiGAN$^1_1$, we designate the best model (with the highest AUROC scores) as open-box and the remaining 9 models as black-box. Thus, adversarial samples are generated using the white-box model and evaluated against all the top 10 single WGAN-based VehiGAN$^1_1$. Fig. 9(c) demonstrates that while the white-box attacks result in an 80-100% FPR, the black-box attacks demonstrate very limited adversarial response, exhibiting reactions akin to random noise.

While certain attacks may transfer at higher epsilon values, the extent is unclear, as any random perturbation with the same intensity produces a comparable effect. For example, the 25-70% FPR at epsilon 0.02 in a black-box attack may not solely be attributed to adversarial perturbation. Random noise with a similar strength can itself result in 20-60% FPR (as shown in Fig. 9(a)). We hypothesize that such adversarial non-transferability may arise from the distinctive learning approach (implicit density estimation) of GANs, differing from traditional DL methods. This may result in diverse loss landscapes among different WGANs (discriminators), impeding the transferability of adversarial samples, which serves as another motivation for considering ensemble-based approaches in VehiGAN. The following sections delve into evaluating the robustness of ensemble-based VehiGAN.

6.2.2 Robustness of Ensemble-based VehiGAN^k_m. In this analysis, we examine two practical adversarial scenarios.

Gray-box Single-Model Greedy AFP Attack. Firstly, we consider a less sophisticated attacker who generates AFP samples solely using the best-performing single-WGAN-based VehiGAN and employs them to attack the ensemble-based VehiGAN where the compromised model itself is present in the ensemble. We consider this a gray-box greedy attack, assuming that the attacker is constrained, either lacking white-box access to all the WGAN models in the ensemble or the ability to attack more than one model at a time. Thus, the attacker takes an opportunistic, greedy approach, anticipating that adversarial samples from the best model will transfer to all models in the ensemble.

The left panel of Fig. 11(a) depicts the FPRs of VehiGAN^k_m with different m and all the possible values of k under such gray-box AFP attacks ($\epsilon = 0.01$). Despite achieving an FPR of $> 80\%$ against the white-box VehiGAN$^1_1$, when applied to the ensemble-based VehiGAN^k_m, the FPR substantially decreases. Increasing the number of candidate models (m) in the ensemble increases uncertainty, diminishing the effectiveness of the attacks. The right panel of Fig. 11(a) shows the specific impact of the number of deployed models (k) for different m. The

figure shows that for the same m, deploying more models (higher k) further eradicates the impact of such AFP attacks. VehiGAN^k_m with $m \geq 5$ and $k \geq 2$ mostly provides FPRs of less than 5%, demonstrating the adversarial robustness of VehiGAN against gray-box greedy AFP transfer attacks.

White-box Multi-model Adaptive AFP Attack. Thereafter, we consider more advanced and adaptive attacks with the attacker having greater knowledge and computational capabilities. Under this scenario, such an adaptive attacker has complete knowledge of the defense and has access to all the weights and gradients of all the discriminators used in VehiGAN^k_m. During AFP sample generation, the attacker utilizes all these discriminators in loss calculation to increase anomaly scores for the ensembled model. The right panel of Fig. 11(b) shows FPRs of VehiGAN^k_m with different m and all the possible values of k. It is evident that VehiGAN^k_m still demonstrates high adversarial robustness against multi-model adaptive AFP attacks. There exist limited adversarial samples that are effective against all discriminators (when $m > 2$) simultaneously. It is also evident from the right panel of Fig. 11(b) that FPR falls below 5% for most VehiGAN configurations with $m > 5$ and $k \geq 5$. Such findings further support the discriminators' unique loss landscapes and the nontransferability property (Fig. 9(c)). Therefore, such adaptive adversarial attacks neither transfer nor are effective against multi-WGAN-based VehiGAN^k_m.

Table 3. AUROC scores of VehiGAN compared to other baselines (bold highlights the best) across attacks.

	Vehi-GAN$^{10}_{10}$	Vehi-GAN$^5_5$	Base-AE	Vehi-AE	Vehi-PCA	Vehi-KNN	Vehi-GMM
RandomPosition	**1.00**	**1.00**	0.98	**1.00**	**1.00**	**1.00**	**1.00**
RandomPositionOffset	**1.00**	**1.00**	0.49	**1.00**	0.95	0.99	**1.00**
PlaygroundConstantPosition	**0.87**	0.84	0.48	0.80	0.4	0.74	0.82
ConstantPositionOffset	0.49	0.48	0.51	0.49	**0.53**	0.51	0.51
RandomSpeed	0.99	0.99	0.77	**1.00**	0.98	0.99	**1.00**
RandomSpeedOffset	0.97	0.95	0.60	**1.00**	0.95	0.97	**1.00**
ConstantSpeed	0.94	0.94	0.56	0.98	0.37	0.79	**0.99**
ConstantSpeedOffset	0.93	0.92	0.48	0.96	0.54	0.85	**0.98**
HighSpeed	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
LowSpeed	0.89	0.86	0.48	0.86	0.42	0.8	0.86
RandomAcceleration	0.61	0.56	0.55	**0.98**	0.57	0.73	0.83
RandomAccelerationOffset	0.51	0.52	0.47	**0.92**	0.53	0.64	0.71
ConstantAcceleration	0.41	0.56	0.74	**1.00**	0.94	**0.99**	0.97
ConstantAccelerationOffset	0.44	0.54	0.59	**0.95**	0.62	0.78	**0.89**
HighAcceleration	0.95	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
LowAcceleration	0.97	0.99	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
RandomHeading	**1.00**	**1.00**	0.97	**1.00**	0.99	**1.00**	**1.00**
RandomHeadingOffset	**1.00**	**1.00**	0.84	**1.00**	0.99	0.99	**1.00**
ConstantHeading	**0.88**	0.86	0.25	0.82	0.48	0.75	0.84
ConstantHeadingOffset	**0.89**	0.88	0.79	0.83	0.6	0.81	0.83
OppositeHeading	**0.91**	0.89	0.66	0.86	0.52	0.83	0.86
PerpendicularHeading	**0.9**	0.89	0.70	0.81	0.45	0.76	0.81
RotatingHeading	**0.84**	**0.84**	0.47	0.78	0.51	0.65	0.81
RandomYawRate	0.97	0.96	0.46	**0.99**	0.87	0.82	0.97
RandomYawRateOffset	0.93	0.91	0.50	**0.98**	0.8	0.74	0.95
ConstantYawRate	0.95	0.93	0.57	0.96	0.81	0.67	**0.98**
ConstantYawRateOffset	**0.99**	**0.99**	0.43	**0.99**	0.95	0.93	**0.99**
HighYawRate	**1.00**	0.99	0.59	**1.00**	0.97	0.97	**1.00**
LowYawRate	**1.00**	0.99	0.54	**1.00**	0.96	0.96	**1.00**
RandomHeadingYawRate	**1.00**	**1.00**	0.76	**1.00**	0.97	0.98	0.99
RandomHeadingYawRateOffset	**1.00**	**1.00**	0.72	**1.00**	0.94	0.96	0.99
ConstantHeadingYawRate	**0.78**	0.77	0.39	0.77	0.49	0.71	**0.78**
ConstantHeadingYawRateOffset	**1.00**	**1.00**	0.89	**1.00**	**1.00**	**1.00**	**1.00**
HighHeadingYawRate	**1.00**	**1.00**	0.88	**1.00**	**1.00**	**1.00**	**1.00**
LowHeadingYawRate	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
Average	0.89	0.89	0.66	**0.93**	0.77	0.87	0.92

6.3 Baseline Comparison

In this analysis, we compare the performance of two representatives VehiGAN (*i.e.*, VehiGAN$^5_5$ and VehiGAN$^{10}_{10}$) with other baseline methods mentioned in Section 5.4. Table 3 provides the AUROC scores of individual detectors against individual attacks. As shown, in 31 out of the 35 attacks, VehiGAN$^{10}_{10}$ or VehiGAN$^5_5$ outperformed the raw-based BaseAE, indicating the effectiveness of VehiGAN. Moreover, to evaluate the effectiveness of the feature engineering step in VehiGAN, we further show the effectiveness of all the baselines trained on the extracted features. Such baselines are named with the prefix Vehi- in the table. As illustrated, feature engineering boosted the performance of all such VehiGAN-assisted baselines, indicating its widespread adaptability. However, in 20 out of the 35 attacks, VehiGAN$^{10}_{10}$ still provided the best performance.

While in the majority of the 15 other attacks, VehiGAN did not achieve the highest AUROC scores, it consistently demonstrated a level of detection performance nearly on par with the top-performing baselines. VehiGAN$^{10}_{10}$ particularly stands out from the other baselines to secure specific intricate features like heading with unique

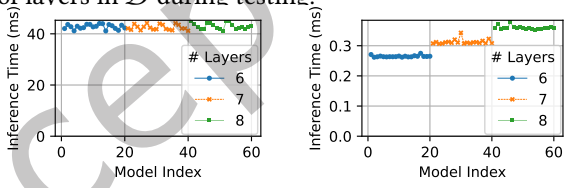
attacks, such as *RotatingHeading* or *PerpendicularHeading*, etc., characterized by their complex misbehaviors. Furthermore, in the threat model, we consider advanced attacks (last six rows in Table 3) that manipulate both the heading & yaw rate fields and VehiGAN$^{10}_{10}$ appeared as the most effective MBDS against such sophisticated attacks.

However, it is worth noting that VehiGAN showed low performance against some of the acceleration-related attacks. One possible explanation for this is the noisy acceleration produced by VASP, even under benign conditions. This unwanted simulation artifact has been reported on the VASP Github. Given the sensitivity of training WGAN, compared to AE, this noise could have potentially hindered the network's ability to effectively learn and mitigate acceleration-related misbehaviors. Conversely, all the models failed to detect *ConstantPositionOffset* attacks as they do not violate any physics, and the only way to detect them is to use additional features, such as raw positions in VehiGAN, or run consistency checks with map data, which can work parallel as an additional detector along with VehiGAN.

6.4 Scalability Analysis

Training WGAN models can be computationally intensive due to their implicit density estimation. However, since training occurs offline on high-performance systems or cloud platforms with GPUs, it does not impact the scalability of deploying VehiGAN. To assess VehiGAN's scalability, we focus on inference times (in milliseconds) for each of the 60 discriminators with varying numbers of layers in $\mathcal{D}$ during testing.

We implement both standard versions of each discriminator using Keras and lightweight versions with TensorFlow Lite (TFLite), measuring their inference times. All experiments are conducted on a server with an Intel Core i7-8700K CPU running at 3.70GHz on Ubuntu 18.04.3 LTS. As shown in Fig. 12(a), standard models require around 40 ms for inference, well within the 100 ms interval of Basic Safety Messages (BSM), allowing timely misbehavior detection in VehiGAN when models run in parallel. For systems without parallel inference capabilities, TFLite models offer a low-overhead alternative, with inference times under 0.40 ms across all $\mathcal{D}$ configurations (Fig. 12(b)). Although adding layers slightly increases TFLite model inference time, it remains negligible relative to the BSM interval.



(a) Standard Model (b) TFLite Model

Fig. 12. Scalability analysis of VehiGAN.

7 RELATED WORK

Different statistical approaches are utilized for misbehavior detection in the V2X scenario. Valentini et al. [54] used a statistical approach for anomaly detection in V2V communication. Their proposed method works at the medium access control (MAC) layer, focusing on identifying potentially malicious nodes and maintaining a reputation list. One downside of statistical approaches is that they mostly face limitations in identifying novel or zero-day attacks.

Due to the shortcomings of statistical methods for anomaly detection, ML-based methods are preferred since they can learn from data to generalize the problem. Different supervised ML-based MBDS for V2X [13, 20, 45] by previously explored research works. They explored the effectiveness of common algorithms such as logistic regression, support vector machine, KNN, naive Bayes, random forest, decision tree classifier, etc., for misbehavior detection in V2X. This line of research also investigated the plausibility of checks-based detectors and evaluated the performance of the existing misbehavior datasets. Ercan et al [13] proposed extracting new features to enhance the detection performance of such models and further studied the efficacy of ensemble-based approaches. DL-based supervised and semi-supervised models using convolutional neural networks (CNN), LSTM, and

transformer are also explored in [1, 31]. However, these works were implemented on limited features that skewed their detection range.

Moreover, such supervised models often encounter difficulties in terms of achieving robust generalization and struggle to detect unknown and evolving attack patterns, known as zero-day attacks. These challenges stem from the limitations of insufficient and imbalanced training datasets. The main factor contributing to this is the scarcity of real-world attack data due to a lack of deployment. Furthermore, simulated data is not always faithfully representative of real-world situations, magnifying the generalization issue. For example, simulated data will always struggle with long-tail data distribution and the generation of edge cases.

Reinforcement Learning (RL)-based misbehavior detection in V2X scenarios gained traction in recent years. For example, Sedar et al. evaluate the effectiveness of RL approaches for this task, focusing on real-time position and speed patterns [41]. Another work with deep RL employs transfer learning for collaborative misbehavior detection among roadside units (RSUs) [40]. In the presence of attacks, they perform selective knowledge transfer based on the trustworthiness of source RSUs to endorse relevant expertise in misbehavior detection. There are a few works that utilize RL for an ensembling approach for MBDS in V2X. For example, [39] proposed a data-driven ensemble framework that combines KNN-based clustering and RL to detect misbehaviors in unlabeled vehicular data. It highlights the potential challenges of inconsistent or mislabeled training data and assesses their performance against various attacks. Nevertheless, RL-based approaches require substantial labeled training data and computational resources and may not generalize well to real-life situations.

Another line of work for MBDS is based on trajectory verification and checkpoint tracking, which utilizes V2X messages. Nguyen et al. proposed an approach to verify the motion behavior of a target vehicle and the truthfulness of data in cooperative vehicular communications by using checkpoints in predicted trajectories [34]. Physical layer plausibility checks also seemed efficient. So et al. [49] introduced the idea of physical layer plausibility checks based on the received signal strength indicator (RSSI) of basic safety messages (BSMs). However, these types of defenses are only effective against the fake node-based attacker and location-based misbehaviors, leaving the rest of the fields undefended.

In the context of anomaly detection, individual GANs and their ensemble variants have been studied extensively. For example, Durugkar et al. introduced a multi-discriminator-based GAN architecture aimed at better approximating the data distribution, thereby enabling a more stringent critique of the generator [12]. Contrastingly, Zhang et al. proposed a framework comprising multiple generators within the GAN architecture [59] with a focus on optimizing the performance of the generator. Han et al. advocated for a GAN framework consisting of multiple generators and discriminators, where each generator undergoes critique from every discriminator [21]. Also, each discriminator evaluates synthetic samples from every generator. While these approaches serve as motivation for our work, none have been tested on the V2X misbehavior datasets. We adhere to the basic WGAN architecture, prioritizing faster and more stable training while maintaining greater control over the individual components of the WGAN architecture.

8 DISCUSSION

Our approach in designing VehiGAN, in contrast to the aforementioned studies, incorporates several practical considerations. Firstly, VehiGAN relies on GAN, an unsupervised DL model that does not necessitate any attack data for training. Hence, although VehiGAN is evaluated against 35 distinct types of misbehavior, it is inherently designed to detect zero-day attacks, as its learning process is exclusively based on the representation of benign behavior within the V2X communication paradigm. Furthermore, as discussed in Section 4.3, physics-guided feature engineering does not replace data-driven feature learning; rather, it enhances the learning process by incorporating domain-specific insights. Such a pipeline exhibits a high degree of flexibility, allowing VehiGAN to incorporate raw or processed features into the detection pipeline without modifying the overall system design.

This integration ensures that GAN focuses on learning complex patterns beyond what can already be derived from domain knowledge, thereby improving the overall effectiveness of the feature extraction process.

We leveraged the distinctive characteristics of implicit density estimation of the GAN learning framework to develop an adversarially robust MBDS capable of withstanding highly adaptive attacks, even in scenarios where the attacker has full access to all the models. To our knowledge, none of these prior works are reproducible, as they did not share their code. Notably, we make both our code and data publicly available to support transparency and reproducibility. Thus, we had to resort to common outlier detection algorithms to establish baselines.

9 CONCLUSION

This work presents VehiGAN, an ensemble-based MBDS for V2X networks, leveraging top-performing GANs to address security challenges through enhanced generalization and robustness. VehiGAN possesses physics-guided feature engineering, training of diverse GAN models, pre-evaluating using different metrics, and selecting top-performing GANs for the ensemble. For the evaluation, we leverage a state-of-the-art V2X attack simulator and generate a comprehensive V2X misbehavior dataset. The evaluation shows VehiGAN outperformed baseline models in 20 out of 35 attacks and displayed similar performance in the remainder. VehiGAN showed more effectiveness against advanced misbehavior targeting multiple fields (such as heading & yaw rate) in V2X messages simultaneously. Furthermore, VehiGAN shows approximately 92% improvement in FPR under powerful adaptive adversarial attacker AFP attacks and inherent robustness against AFN attacks. Our findings highlight the promise of GAN-based approaches for V2X misbehavior detection, particularly in dynamic and complex threat landscapes.

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under grants 2235232, 2154929, 2247560, 2312447, and 2332675, by the Office of Naval Research under grant N00014-24-1-2730, and by the Virginia Commonwealth Cyber Initiative (CCI).

REFERENCES

[1] Hayotjon Aliev and HyungWon Kim. 2021. Misbehavior detection based on multi-head deep learning for V2X network security. In *International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE.

[2] Theodore W Anderson and Donald A Darling. 1954. A test of goodness of fit. *Journal of the American statistical association* 49, 268 (1954), 765–769.

[3] Mohammad Raashid Ansari, Jonathan Petit, Jean-Philippe Monteuuis, and Cong Chen. 2023. VASP: V2X Application Spoofing Platform. In *Proceedings Inaugural International Symposium on Vehicle Security & Privacy, ndss-symposium*. https://doi.org/10.14722/vehiclesec.2023.23071

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR.

[5] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401* (2018).

[6] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schølkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* 22, 14 (2006), e49–e57.

[7] Mohammed Lamine Bouchouia, Houda Labiod, Ons Jelassi, Jean-Philippe Monteuuis, Wafa Ben Jaballah, Jonathan Petit, and Zonghua Zhang. 2023. A survey on misbehavior detection for connected and autonomous vehicles. *Vehicular Communications* (2023).

[8] Benedikt Brecht, Dean Therriault, André Weimerskirch, William Whyte, Virendra Kumar, Thorsten Hehn, and Roy Goudy. 2018. A Security Credential Management System for V2X Communications. *IEEE Transactions on Intelligent Transportation Systems* (2018). https://doi.org/10.1109/TITS.2018.2797529

[9] V2X Core Technical Committee. 2023. *V2X Communications Message Set Dictionary*. https://doi.org/10.4271/J2735_202309

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[11] DC Dowson and BV666017 Landau. 1982. The Fréchet distance between multivariate normal distributions. *Journal of multivariate analysis* 12, 3 (1982), 450–455.

[12] Ishan Durugkar, Ian Gemp, and Sridhar Mahadevan. 2016. Generative multi-adversarial networks. *arXiv preprint arXiv:1611.01673* (2016).

[13] Secil Ercan, Marwane Ayaida, and Nadhir Messai. 2021. Misbehavior detection for position falsification attacks in VANETs using machine learning. *IEEE Access* (2021).

[14] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633* (2017).

[15] Giovanni Fasano and Alberto Franceschini. 1987. A multidimensional version of the Kolmogorov–Smirnov test. *Monthly Notices of the Royal Astronomical Society* 225, 1 (1987), 155–170.

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* (2014).

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[18] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schølkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13, 1 (2012), 723–773.

[19] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. *Advances in neural information processing systems* (2017).

[20] Sohan Gyawali and Yi Qian. 2019. Misbehavior detection using machine learning in vehicular communication networks. In *International Conference on Communications*. IEEE.

[21] Xu Han, Xiaohui Chen, and Li-Ping Liu. 2021. Gan ensemble for anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[22] James A Hanley and Barbara J McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* (1982).

[23] Monowar Hasan, Sibin Mohan, Takayuki Shimizu, and Hongsheng Lu. 2020. Securing vehicle-to-everything (V2X) communication platforms. *IEEE Transactions on Intelligent Vehicles* (2020).

[24] Ke He, Dan Dongseong Kim, and Muhammad Rizwan Asghar. 2023. Adversarial machine learning for network intrusion detection systems: a comprehensive survey. *IEEE Communications Surveys & Tutorials* (2023).

[25] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

[26] Geoffrey E Hinton and Ruslan R Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *science* (2006).

[27] Victoria Hodge and Jim Austin. 2004. A survey of outlier detection methodologies. *Artificial intelligence review* (2004).

[28] Yifan Jia, Jingyi Wang, Christopher M Poskitt, Sudipta Chattopadhyay, Jun Sun, and Yuqi Chen. 2021. Adversarial attacks and mitigation for anomaly detectors of cyber-physical systems. *International Journal of Critical Infrastructure Protection* 34 (2021), 100452.

[29] Joseph Kamel, Michael Wolf, Rens W Van Der Hei, Arnaud Kaiser, Pascal Urien, and Frank Kargl. 2020. Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE.

[30] Daniel Krajzewicz. 2010. Traffic simulation with SUMO–simulation of urban mobility. *Fundamentals of traffic simulation* (2010).

[31] Zhikang Liu, Hongyun Xu, Yong Kuang, and Feng Li. 2023. SVMDformer: A Semi-Supervised Vehicular Misbehavior Detection Framework Based on Transformer in IoV. In *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 887–897.

[32] Md Julkar Nayeen Mahi, Sudipto Chaki, Shamim Ahmed, Milon Biswas, M Shamim Kaiser, Mohammad Shahidul Islam, Mehdi Sookhak, Alistair Barros, and Md Whaiduzzaman. 2022. A review on VANET research: Perspective of recent emerging technologies. *IEEE Access* (2022).

[33] Jean-Philippe Monteuuis, Jonathan Petit, Jun Zhang, Houda Labiod, Stefano Mafrica, and Alain Servel. 2018. Attacker model for connected and automated vehicles. In *ACM Computer Science in Car Symposium*.

[34] Van-Linh Nguyen, Po-Ching Lin, and Ren-Hung Hwang. 2020. Enhancing misbehavior detection in 5G vehicle-to-vehicle communications. *IEEE Transactions on Vehicular Technology* (2020).

[35] World Health Organization. 2018. Global status report on road safety. https://www.who.int/publications/i/item/9789241565684

[36] Ana Pereira and Carsten Thomas. 2020. Challenges of machine learning applied to safety-critical cyber-physical systems. *Machine Learning and Knowledge Extraction* (2020).

[37] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. 2000. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*.

[38] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40 (2000), 99–121.

[39] Roshan Sedar, Charalampos Kalalas, Paolo Dini, Jesus Alonso-Zarate, and Francisco Vázquez-Gallego. 2022. Misbehavior Detection in Vehicular Networks: An Ensemble Learning Approach. In *Global Communications Conference*. IEEE.

[40] Roshan Sedar, Charalampos Kalalas, Paolo Dini, Francisco Vázquez-Gallego, Jesus Alonso-Zarate, and Luis Alonso. 2024. Knowledge Transfer for Collaborative Misbehavior Detection in Untrusted Vehicular Environments. *IEEE Transactions on Vehicular Technology* (2024).

[41] Roshan Sedar, Charalampos Kalalas, Francisco Vázquez-Gallego, and Jesus Alonso-Zarate. 2022. Reinforcement learning based misbehavior detection in vehicular networks. In *International Conference on Communications*. IEEE.

[42] Md Hasan Shahriar, Mohammad Raashid Ansari, Jean-Philippe Monteuuis, Cong Chen, Jonathan Petit, Y. Thomas Hou, and Wenjing Lou. 2024. MisbehaviorX: Comprehensive V2X Misbehavior Detection Dataset Enabled by the V2X Application Spoofing Platform. https://doi.org/10.21227/s44z-8616

[43] Md Hasan Shahriar, Mohammad Raashid Ansari, Jean-Philippe Monteuuis, Cong Chen, Jonathan Petit, Y Thomas Hou, and Wenjing Lou. 2024. Vehigan: Generative Adversarial Networks for Adversarially Robust V2X Misbehavior Detection Systems. In *2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 1294–1305.

[44] Md Hasan Shahriar, Nur Imtiazul Haque, Mohammad Ashiqur Rahman, and Miguel Alonso. 2020. G-ids: Generative adversarial networks assisted intrusion detection system. In *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE.

[45] Prinkle Sharma and Hong Liu. 2020. A machine-learning-based data-centric misbehavior detection model for internet of vehicles. *IEEE Internet of Things Journal* (2020).

[46] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. 2003. A novel anomaly detection scheme based on principal component classifier. In *Proceedings of the IEEE foundations and new directions of data mining workshop*. IEEE Press.

[47] Santokh Singh. 2018. Critical reasons for crashes investigated in the National Motor Vehicle Crash Causation Survey. https://crashstats.nhtsa.dot.gov/Api/Public/Publication/812506

[48] Nickolay Smirnov. 1948. Table for estimating the goodness of fit of empirical distributions. *The annals of mathematical statistics* 19, 2 (1948), 279–281.

[49] Steven So, Jonathan Petit, and David Starobinski. 2019. Physical layer plausibility checks for misbehavior detection in V2X networks. In *Proceedings of the 12th conference on security and privacy in wireless and mobile networks*.

[50] Christoph Sommer, David Eckhoff, Alexander Brummer, Dominik S Buse, Florian Hagenauer, Stefan Joerer, and Michele Segata. 2019. Veins: The open source vehicular network simulation framework. *Recent Advances in Network Simulation: The OMNeT++ Environment and its Ecosystem* (2019).

[51] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[52] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[53] V2X Core Technical Committee. 2020. *On-Board System Requirements for V2V Safety Communications*. https://doi.org/10.4271/J2945/1_202004

[54] Edivaldo Pastori Valentini, Geraldo Pereira Rocha Filho, Robson Eduardo De Grande, Caetano Mazzoni Ranieri, Lourenço Alves Pereira, and Rodolfo Ipolito Meneguette. 2023. A Novel Mechanism for Misbehaviour Detection in Vehicular Networks. *IEEE Access* (2023).

[55] Rens W Van Der Heijden, Thomas Lukaseder, and Frank Kargl. 2018. Veremi: A dataset for comparable evaluation of misbehavior detection in vanets. In *Security and Privacy in Communication Networks: 14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018, Proceedings, Part I*. Springer.

[56] Andras Varga. 2010. OMNeT++. In *Modeling and tools for network simulation*. Springer.

[57] Apostol Vassilev, Alina Oprea, Alie Fordyce, and Hyrum Anderson. 2024. *Adversarial machine learning: A taxonomy and terminology of attacks and mitigations*. Technical Report. National Institute of Standards and Technology.

[58] Xuan Xia, Xizhou Pan, Nan Li, Xing He, Lin Ma, Xiaoguang Zhang, and Ning Ding. 2022. GAN-based anomaly detection: A review. *Neurocomputing* (2022).

[59] Hongyang Zhang, Susu Xu, Jiantao Jiao, Pengtao Xie, Ruslan Salakhutdinov, and Eric P Xing. 2018. Stackelberg GAN: Towards provable minimax equilibrium via multi-generator architectures. *arXiv preprint arXiv:1811.08010* (2018).

Received 15 November 2024; revised 10 March 2025; accepted 30 May 2025