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

Md Hasan Shahriar*, Mohammad Raashid Ansari†, Jean-Philippe Monteuuis†, Cong Chen†, Jonathan Petit†
Y. Thomas Hou*, Wenjing Lou*
*Virginia Polytechnic Institute and State University, Blacksburg, VA, USA
{hshahriar, thou, wjlou}@vt.edu
†Qualcomm Technologies, Inc., Boxborough, MA, USA
m.raashid.ansari@gmail.com, {jmonteuu, congchen, petit}@qti.qualcomm.com

***Abstract*—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, 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 such 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.**

Our contributions are manifold. In the pursuit of an effective 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 and adversarial robustness. We present a physics-guided data preprocessing technique that generates effective features for ML-based misbehavior detection. To evaluate the adversarial robustness, we formulate two categories of adversarial attacks against the WGAN-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 baselines 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 rates under powerful adaptive adversarial attacks and possesses intrinsic robustness against other adversarial attacks that target false negative rates. Finally, we make the data and code available for reproducibility and future benchmarking, available at https://github.com/shahriar0651/VehiGAN.

***Index Terms*—vehicular network, misbehavior detection systems, generative adversarial networks, deep learning**

I. INTRODUCTION

Road traffic accidents take approximately 1.35 million lives every year around the world, leaving another 50 million non-fatally injured [1]. Approximately 94% of major accidents in conventional transportation systems are caused, at least in part, by human errors [2]. Conversely, a connected and 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 the Vehicle-to-Everything (V2X) communication that allows vehicles to communicate with their environment, such as other vehicles (V2V), infrastructure (V2I), and pedestrians (V2P) [3]. V2X technology provides vehicles with real-time traffic information along with alerts on potential hazards, which help coordinate traffic flow, avoid collisions, and minimize 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 [4]. 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 [5]. 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 the outsider attacker, it still poses several security challenges [6], especially from insider attackers. Insider attackers have valid access credentials but disseminate wrong information to achieve the attack goals [7]. Hence, while the 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, serving as an essential defense

for V2X communication system [6].

The MBDS, usually running on an ego vehicle, receives BSMs from another vehicle and checks whether the content has anomalies or is physically implausible [6]. Upon observing 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) [6]. 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 [5].

Nevertheless, MBDS confronts a multitude of formidable challenges [8], rendering it a complex and evolving research task. There are different MBDSs proposed in the existing research [8] to detect malicious or erroneous V2X messages. While the state-of-the-art threat landscape has become quite broad [9], mandating a comprehensive solution, most of the existing MBDSs provide a partial defense, focusing on specific types of attacks and features [6]. Although the conventional discriminative 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 insufficient and imbalanced training datasets [10]. Moreover, the traditional DL-based methods are proven vulnerable to adversarial attacks [11]. On the other hand, generative adversarial networks (GAN) [12], a generative DL model, has already proven the capability to overcome such data imbalance issues as they can synthesize unseen samples of minority classes (such as rare vehicular states) and utilize them during its own model training [13]. Moreover, 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 a robust anomaly detection system [14].

Hence, to the best of our knowledge, we are the first to explore the adaptability of GAN in designing a robust MBDS for V2X.

Our contributions are as follows:

- We study the feasibility of using Wasserstein GAN (WGAN), one of the most prominent and stable variants of the GANs [15], 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 present a physics-guided data preprocessing technique that generates effective features from raw V2X attributes for any ML-based MBDS.
- To investigate the robustness against adversarial adaptive attackers, we formulate two categories of attacks targeting the WGAN-based MBDS. We assess the impacts across a spectrum of scenarios, ranging from white-box to black-box settings and from single-model to multi-model attacks.
- Employing the state-of-the-art V2X attack simulation tool VASP [9], we generate an extensive V2X message dataset containing 68 distinct types of misbehaviors, representing a substantial enhancement compared to prior V2X misbehavior datasets [16], [17]. We make them publicly available to advance the 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. Moreover, VEHIGAN shows approximately 92% improvement in false positive rate under one type of powerful adaptive attack and intrinsic robustness against other type of attacks that aim for high false negatives.

The rest of the paper is organized as follows: We introduce an overview of the background and threat model in Section II. Section III presents the technical details of VEHIGAN. We provide an experimental setup and implementation details in Section IV. The evaluation results are in Section V. The related works are discussed in Section VI. Finally, we conclude the paper in Section VII.

II. Background and Threat Model

A. Generative Adversarial Networks

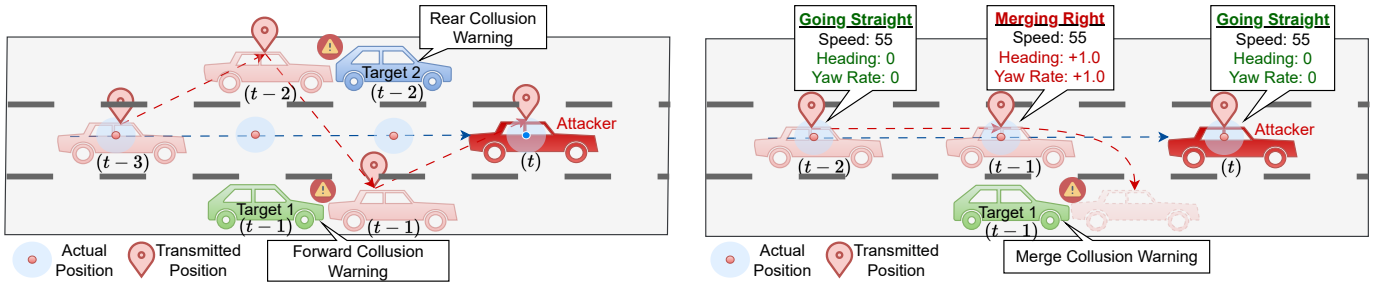
GAN, introduced by Ian Goodfellow in 2014 [12], is an implicit generative model based on artificial neural networks. It has become a popular technique for generating realistic data (e.g., image, video, audio) that resemble the distribution of the training dataset. Out of different variants of GAN, Wasserstein GAN (WGAN) with gradient penalty is the most popular due to its high performance, robustness, and training stability [18].

Like other GAN variants, WGAN 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 $\mathcal{G}(\mathbf{z})$. The discriminator, on the other hand, is tasked with distinguishing between real sample $\mathbf{x}$ from the training data distribution (P_r) and generated fake data sample $\mathcal{G}(\mathbf{z})$. 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. Thus, 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 [12]. Mathematically, the optimization problem can be expressed as:

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

TABLE I: 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	①	⑤	⑪	⑰	㉔	㉚
Random Offset	Value with random offset	②	⑥	⑫	⑱	㉕	㉛
Constant	Constant value	③	⑦	⑬	⑲	㉖	㉜
Constant Offset	Value with constant offset	④	⑧	⑭	⑳	㉗	㉝
High	Significantly high value		⑨	⑮		㉘	㉞
Low	Significantly low value		⑩	⑯		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. 1: 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.

Here, $\mathcal{D}(x)$ and $\mathcal{D}(\mathcal{G}(z))$ are the outputs of the discriminator on a real and a generated sample, 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 [19].

B. Fast Gradient Sign Method (FGSM) Attack

The Fast Gradient Sign Method (FGSM) is an adversarial attack technique to deceive DL classification models [20]. 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 can be expressed as:

$$\mathbf{x_{adv}} = \mathbf{x} + \epsilon \cdot \text{sign}(\nabla_x \mathcal{L}(f(\mathbf{x}), y)) \tag{2}$$

Here, $\mathbf{x_{adv}}$ represents the adversarial example, $f(\mathbf{x})$ is the model's prediction on input, y is the actual label, $\nabla_x \mathcal{L}$ denotes the gradient of the model's loss 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 [21]. Let us use the notation s to represent the anomaly detector that returns the anomaly score. In this notation, the equation becomes:

$$\mathbf{x_{adv}} = \mathbf{x} \pm \epsilon \cdot \text{sign}(\nabla_x s(\mathbf{x})) \tag{3}$$

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. We adopt this approach to evaluate the adversarial robustness of VEHIGAN, further elaborated in Section III-G.

C. Threat Model

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. They are *local* parties and transmit deceptive BSMs containing false data to achieve their malicious objectives. Table I summarizes the overall threat landscape, outlining various attack types, targeted field(s), and the description of the value transmitted in the targeted field(s). The circle ⓒ 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. 1a). 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 assume that to keep attack complexity low, most of the attacks (① − 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 when the attacker compromises both the heading & yaw rate (as illustrated in Fig. 1b) (㉚ – ㉟) and modifies these two fields together, coherently, following their inter-dependency.

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 use this threat matrix in Section IV-A to generate a misbehavior dataset and evaluate VEHIGAN's performance.

We further explore two types of adversarial attacks, namely, white-box and black-box attacks [22], wherein adversaries can generate adversarial input by making subtle adjustments to any/all of the sensor values, following the patterns outlined by the specific attack algorithms. The perturbations introduced are designed to be so unnoticeable that they closely resemble the inherent noise present in natural sensor data. In the white-box adversarial attacks, the attacker possesses complete knowledge of the detection mechanism, and 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).

III. VEHIGAN: GAN-BASED MBDS

This section first describes an overview of the VEHIGAN architecture, followed by the details of each part. Fig. 2 shows the workflow of VEHIGAN, its two phases (training and testing), and its different components. The training 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 testing 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.

A. VEHIGAN *Overview*

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.

1) Training Phase: The top part of Fig. 2 shows the training phase of VEHIGAN. In the first step, VEHIGAN collects BSMs from some 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 [23], 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 hyper-parameters on the combined dataset. After training all the models, VEHIGAN starts pre-evaluating the performance of the WGAN models using a validation dataset. Such validation dataset is assumed to have some representative anomalies of the testing data and can give a good estimate of the discriminator's testing performance. 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 testing 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 testing phase.

2) Testing Phase: The bottom part of Fig. 2 shows the testing phase of VEHIGAN, which is completely executed locally on the OBU/RSU. Similar to the training phase, VEHIGAN keeps collecting raw BSMs from individual testing vehicles, runs the feature engineering task, and creates a combined data 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 ensemble them for misbehavior detection. We define such detector as VEHIGAN$_m^k$ that predicts the misbehavior scores with different random k discriminator every time. If the score for any vehicle surpasses a predetermined threshold, which is the average threshold of the deployed k discriminator, VEHIGAN reports that as potential misbehavior. The following subsections explain the details of each part of VEHIGAN in each phase.

B. Collecting Raw V2X Data

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

C. 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

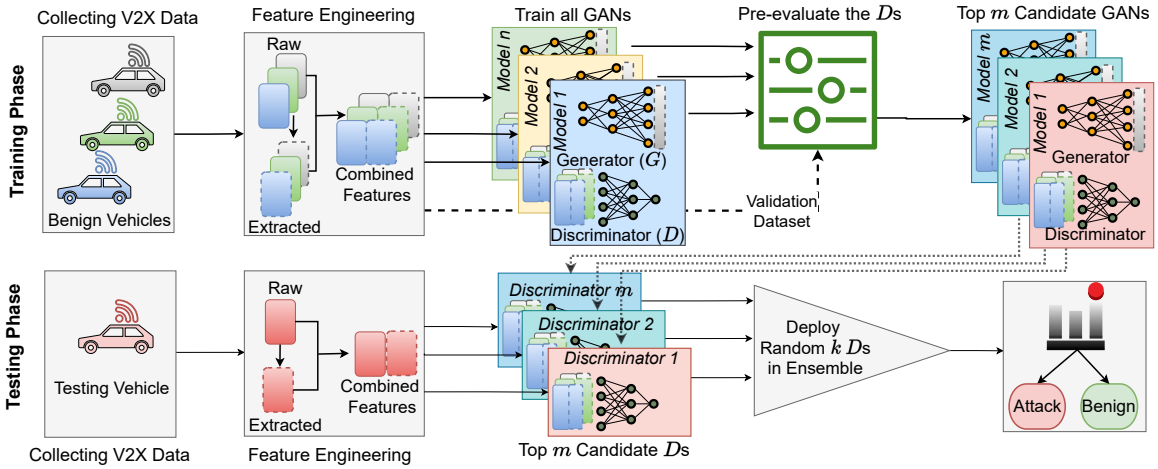


Fig. 2: Workflow of VEHIGAN, which has two phases of operation: i) training phase and ii) testing phase.

TABLE II: 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	$\omega_x = \omega \times cos(\theta)$	$\omega_y = \omega \times sin(\theta)$	$\Delta\theta_x = \omega_x \times \Delta t$	$\Delta\theta_y = \omega_y \times \Delta t$	–	–

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 II 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. 2). 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. These 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.

D. Model Training

To find the best-performing WGAN models, VEHIGAN explores a wide range of model architectures and hyper-parameters. 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 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 process.

E. 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 X_{valid}^{bsm}, containing both the benign and attack traces. We assume that the defender has such representative attack traces, which can be used to pre-evaluate the WGAN models. We define average discriminative score (ADS) as the average detection score (DS) of $\mathcal{D}$ over all the attacks 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 IV-A2). A higher score indicates that a certain $\mathcal{D}$ is more likely to be effective against any unseen

misbehavior in the test data. If there are A different unique attacks in X^{bsm}_{valid}, ADS of the i^{th} WGAN can be expressed as:

$$ADS_i = \frac{1}{A}\sum_{j=1}^{A} DS_i^j \tag{4}$$

Here DS_i^j is the i^{th} discriminator's score against the j^{th} attack in the validation dataset. Consequently, the top m WGAN models with the highest ADS are selected as the candidates for the ensemble model.

F. Threshold Selection and Attack Detection

As the discriminator of a WGAN is architecturally designed to output higher values for benign inputs, we take the negative of that value as anomaly score $\mathbf{s}$ to generalize the misbehavior detection process:

$$s(.) = -\mathcal{D}(.) \tag{5}$$

Hence, the benign anomaly scores of any model on all the snapshots in $\mathbf{X}^{bsm}_{train}$ is calculated as $s(\mathbf{X}^{bsm}_{train})$. The detection threshold τ for each of the individual discriminators is calculated based on the p-th percentile of that where p is a system parameter (usually 99 to 99.99).

Although there are top m candidates, VEHIGAN builds the final ensemble detector by averaging the prediction of randomly selected k discriminators. Thus, the ensemble discriminator $\mathcal{D}_{ens}$ is defined as: $\mathcal{D}_{ens}(.) = \frac{1}{k}\sum_{i=1}^{k}\mathcal{D}_i(.)$ Hence, the benign ensembled anomaly scores $\mathbf{S}_{train} = s_{ens}(\mathbf{X}^{bsm}_{train}) = -\mathcal{D}_{ens}(\mathbf{X}^{bsm}_{train})$ and the ensemble detection threshold τ_{ens} is the average of thresholds of the corresponding models. During the testing phase, the anomaly score of the most recent w BSMs of the target vehicle v is calculated using $\mathbf{s}_v = s_{ens}(\mathbf{x}^{bsm}_v)$, and the detection threshold τ_{ens} is used to check if vehicle v is misbehaving. A value of $\mathbf{s}_v > \tau_{ens}$ indicates the existence of misbehavior, and VEHIGAN immediately creates an MBR on vehicle v, including the corresponding BSMs, and sends it to the MA.

G. Adversarial Robustness of VEHIGAN

Any unsupervised DL-based anomaly detection model, including the discriminators of WGAN, differs from the supervised DL-based classification model. First, they may have different architectures (*e.g.*, the number of neurons in the last layer) and loss functions. While the adversarial attack algorithms are initially developed against the classification-based models [11], we extend them to the anomaly detection model, especially against the discriminators of WGANs. Given that there are eventually two possible outcomes from the discriminators — benign or anomaly — we categorize the FGSM attacks against any anomaly detection model into two types:

1) Adversarial False Positive (AFP) Attack: An APF attack on anomaly detection involves manipulating a benign (negative) input to deceive the model to output an anomaly score high enough to be flagged as an anomaly (false positive). Let us assume the adversarial perturbation on a benign sample $\mathbf{x_{ben}}$ under an AFP attack is $\mathbf{\Delta x^{AFP}_{adv}}$, which can be calculated using the gradient that maximizes the anomaly score. Thus, for the WGAN, combining (3) and (5), we get the following:

$$\begin{aligned}\mathbf{x^{AFP}_{adv}} &= \mathbf{x_{ben}} + \mathbf{\Delta x^{AFP}_{adv}}\\ \mathbf{x^{AFP}_{adv}} &= \mathbf{x_{ben}} + \epsilon\cdot\text{sign}(\nabla_x s(\mathbf{x_{ben}}))\\ \mathbf{x^{AFP}_{adv}} &= \mathbf{x_{ben}} - \epsilon\cdot\text{sign}(\nabla_x \mathcal{D}(\mathbf{x_{ben}}))\end{aligned} \tag{6}$$

The goal of the AFP attacks is to increase the FP rate by crafting adversarial samples that resemble benign data but are misclassified by the discriminator as misbehavior.

2) Adversarial False Negative (AFN) Attack: An AFN attack involves manipulating an anomalous (positive) input to deceive the model to output an anomaly score low enough to be determined as a benign (false negative) one. Let $\mathbf{x_{anom}}$ be an anomalous (misbehavior) sample. Similarly, the adversarial perturbation $\mathbf{\Delta x^{AFN}_{adv}}$ on an anomalous (misbehavior) sample $\mathbf{x_{anom}}$ under an AFN attack is calculated from the gradient that minimizes the anomaly score, following:

$$\begin{aligned}\mathbf{x^{AFN}_{adv}} &= \mathbf{x_{anom}} + \mathbf{\Delta x^{AFN}_{adv}}\\ \mathbf{x^{AFN}_{adv}} &= \mathbf{x_{anom}} - \epsilon\cdot\text{sign}(\nabla_x s(\mathbf{x_{anom}}))\\ \mathbf{x^{AFN}_{adv}} &= \mathbf{x_{anom}} + \epsilon\cdot\text{sign}(\nabla_x \mathcal{D}(\mathbf{x_{anom}}))\end{aligned} \tag{7}$$

The goal of the AFN attacks is to increase the FN rate by crafting adversarial samples that resemble anomalous data but are misclassified by the discriminator as benign. We follow these two algorithms and use X^{bsm}_{valid} dataset to evaluate the adversarial robustness of VEHIGAN under different practical scenarios.

IV. IMPLEMENTATION

A. Dataset

We implement VEHIGAN on the V2X misbehavior dataset simulated using VASP [9], an open-source framework. VASP allows the simulation of diverse types of V2X attacks and works as a sub-module for Veins [23], a well-established open-source framework for running vehicular network simulations. Veins runs on an event-based network simulator OMNeT++ [24], and road traffic simulator SUMO [25]. VASP currently supports the Boston traffic network, which is a good candidate to represent real-world traffic mobility. We ran VASP simulation 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 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 to facilitate future research endeavors. While running the attack, we selected the attack policy as *persistent*, where the attacker vehicle always transmits attack messages and with 25% malicious vehicles. We consider the sliding window size w as 10 and number of features f as 12.

1) Model Architecture. : Based on the hyperparameters mentioned previously, we train a diverse range of WGAN models. We consider different dimension of noise vector z as $\{8, 16, 32, 48, 64\}$, number of layers for $\mathcal{G}/\mathcal{D}$ networks as $\{6, 7, 8\}$, and training epoch as $\{25, 50, 75, 100\}$. Therefore, we train and save 60 WGAN model instances. While we adhere to 60 different instances, training additional models enables a deeper exploration of optimal architecture, albeit with the trade-off of higher training overhead. In training each model, we select a batch size of 128 and a learning rate of 1×10^{-3}. In both the 2D up-sampling layers of $\mathcal{G}$ and the 2D convolution layers of $\mathcal{D}$, we use the filters with the kernel size of 2×2 with the activation function $LeakyReLU$.

2) Evaluation Metrics: 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 [26]. 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.

B. Baseline Models

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. For instance, *Principal Component Analysis* (PCA) uses the sum of weighted projected distances to the eigenvector hyperplane as the outlier scores [27].

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 [28].

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 [29].

4) DL Models for Outlier Detection: DL learns complex and intricate data distribution and effectively detects stealthy and complex anomalies within a large dataset. While VEHIGAN also falls under this category, we consider CNN-based *Autoencoders* (AE) as the DL-based baselines in this study. AE are neural network architectures used for various tasks, including anomaly detection [30]. In the context of outlier detection, AE learns to reconstruct the input data, and data points that are not reconstructed accurately are flagged as outliers. We train the AE baseline on the raw features and name it as BASEAE. However, 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 III.

V. RESULTS

We evaluate the effectiveness of VEHIGAN from different perspectives. First, we analyze the detection performance of individual WGAN against different attacks. Later, we study the effectiveness of ensemble-based VEHIGAN$_m^k$ through the contribution of k deployed models out of m candidate ones. Moreover, we conduct an extensive adversarial robustness analysis of VEHIGAN$_m^k$. Lastly, we compare the performance of two representative VEHIGAN$_m^k$ models with other baseline models.

A. Misbehavior Detection Performance

1) Performance of Single WGAN-based VEHIGAN$_1^1$*:* Fig. 3 provides a comprehensive assessment of all the trained WGANs, specifically discriminators against all the 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 lines for the top three discriminators that provided the highest average AUROC, 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 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.

2) Performance of Ensemble-based VEHIGAN$_m^k$*:* We now evaluate an ensemble-based MBDS to check if combining the top-performing WGAN models harnesses the strengths of each model while mitigating their weaknesses. Fig. 4 shows the impact of m and k on the AUROC scores 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 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.

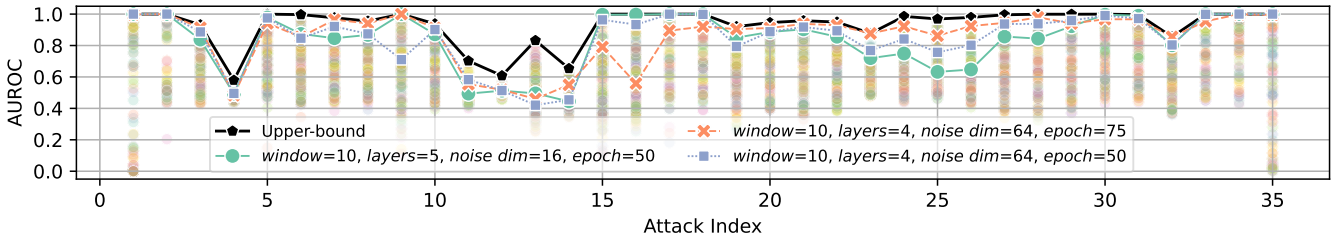


Fig. 3: Performance of all the WGAN models (as shown by different colors), including the top three ones (highlighted), against individual attacks. Different WGAN models perform differently against the same attack, indicating no single WGAN can achieve stable and robust performance.

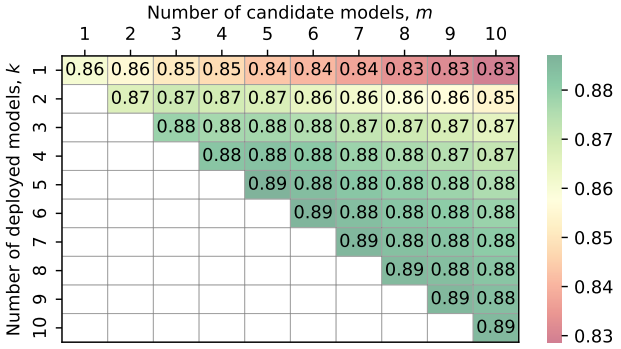


Fig. 4: Average AUROC of VEHIGAN^k_m, where $m \geq 5$ with $k \geq \frac{m}{2}$ leads to consistently elevated AUROC scores.

B. Adversarial Robustness

To assess the adversarial robustness of VEHIGAN, we examine the robustness of individual WGAN-based VEHIGAN under AFP and AFN attacks (outlined in Section III-G). 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. Considering the FGSM attack, we explore values of ϵ within the range of 0.0 to 0.02. To better contrast the impact of such adversarial attacks, we randomize each adversarial perturbation and use it as random noise baselines to evaluate the model's response under a noisy but benign environment.

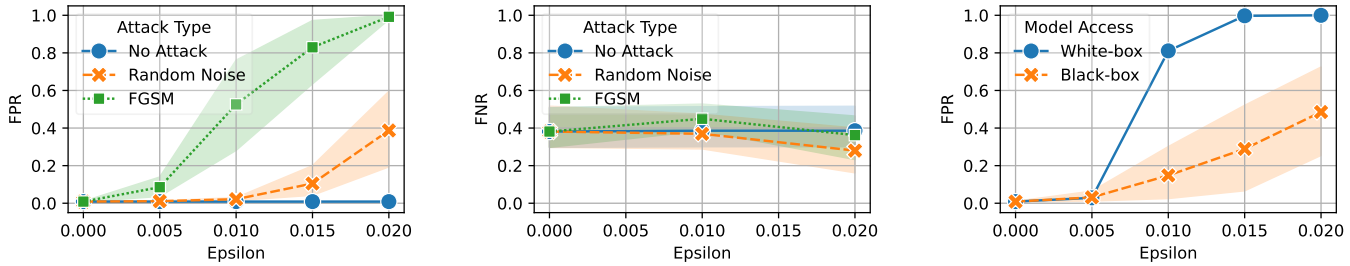
1) Robustness of Single WGAN-based VEHIGAN$^1_1$*:* Fig. 5a illustrates the FPRs of the top 10 single WGAN-based VEHIGAN$^1_1$ models under white-box AFP attacks and random noise. With $\epsilon = 0.01$ (*i.e.*, 1% change in the sensor values), such attacks lead to approximately 50% FPR on average. In 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 sufficient to be labeled as anomalous, resulting in nearly 100% FPR in all the 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. 6 illustrates such an AFP attack with $\epsilon = 0.01$ on a benign input.

On the other hand, Fig. 5b demonstrates the FNR of the top 10 single WGAN-based VEHIGAN$^1_1$ models under AFN attacks. It is evident from the figure that 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.

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 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. 5c 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.

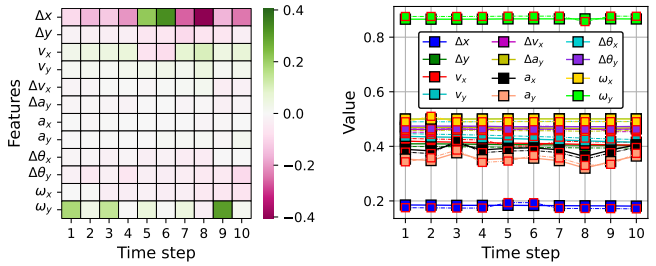
While certain attacks may transfer at higher epsilon values, the extent is unclear, as any random perturbation with the same intensity produces a comparable effect. For example, the 25-70% FPR at epsilon 0.02 in a black-box attack may not solely be attributed to adversarial perturbation. Random noise with a similar strength can itself result in 20-60% FPR (as shown in Fig. 5a). 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.

2) Attacking Ensemble-based VEHIGAN*:* In this analysis, we examine two practical adversarial scenarios. 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



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

Fig. 5: Adversarial robustness of single WGAN-based VEHIGAN$_1^1$ under various attacks



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

Fig. 6: A representation of AFP attack on a benign input. (a) shows the gradient of the loss function with respect to the benign input, as outlined in (6). 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 red edge (□) indicate the corresponding adversarial values of each sensor at different time steps, which are either increased or decreased by $\epsilon = 0.01$ based on the sign of the gradient.

the ensemble. We consider this as a gray-box 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 approach, anticipating that adversarial samples from the best model will transfer to all models in the ensemble.

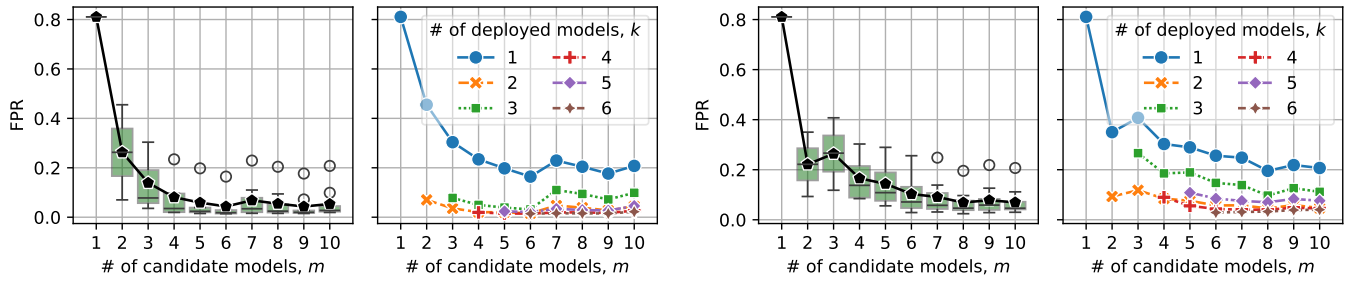
The left panel of Fig. 7a depicts the FPRs of VEHIGAN$_m^k$ with different m and all the possible values of k under such AFP attacks ($\epsilon = 0.01$). Despite achieving an FPR of $> 80\%$ against the white-box VEHIGAN$_1^1$, when applied to the ensemble-based VEHIGAN$_m^k$, the FPR substantially decreases. Increasing the number of candidate models (m) in the ensemble increases uncertainty, diminishing the effectiveness of the attacks. The right panel of Fig. 7a 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 AFP attacks. VEHIGAN$_m^k$ with $m \geq 5$ and $k \geq 2$ mostly provides FPRs of less than 5%, demonstrating the adversarial robustness of VEHIGAN against gray-box AFP transfer attacks.

Thereafter, we consider more advanced and adaptive attacks with the attacker having greater knowledge and computational capabilities. Under this scenario, the attacker has open-box access to all the discriminators used in VEHIGAN$_m^k$. During AFP sample generation, the attacker utilizes all discriminators in loss calculation to increase anomaly scores for the ensembled model. The right panel of Fig. 7b shows FPRs of VEHIGAN$_m^k$ with different m and all the possible values of k. It is evident that VEHIGAN$_m^k$ still demonstrates high adversarial robustness against multi-model 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. 7b 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. 5c). Therefore, adversarial attacks neither transfer nor are effective against multi-WGAN-based VEHIGAN.

C. Performance Comparison with Baselines

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 IV-B. Table III 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 wide-spread 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



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

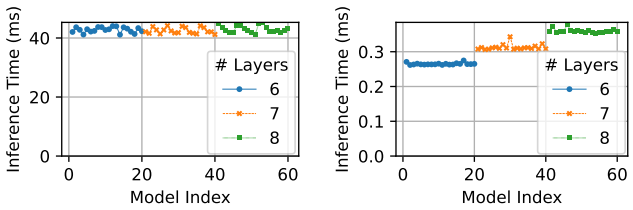
(b) FPR under white-box multi-model AFP attacks

Fig. 7: Adversarial robustness of ensemble-based VEHIGAN^k_m under various APF attacks

TABLE III: AUROC scores of VEHIGAN compared to other baselines (bold highlights the best) against different 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

model, we consider advanced attacks (last six rows in Table III) 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 notified on 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.



(a) Standard Model

(b) TFLite Model

Fig. 8: Scalability analysis of VEHIGAN

D. Scalability Analysis

Training of WGAN models is relatively costly due to their implicit density estimation. However, model trainings are done offline in a powerful computer or cloud, preferably equipped with GPUs. Hence, training overhead does not create scalability issues for deploying VEHIGAN. Instead, to study the scalability of VEHIGAN, we investigate the inference time (in milliseconds) in the testing phase for all the 60 discriminators, with different numbers of layers in the $\mathcal{D}$.

We implement each discriminator's standard version using Keras and the lightweight version using TensorflowLite (TFLite) and calculate their inference times. All experiments run on a server with an Intel Core i7-8700K CPU running at 3.70GHz and Ubuntu 18.04.3 LTS. Fig. 8a shows that the inference time for standard models takes around 40ms, far below the BSM transmission interval of 100ms. Hence, in this case, parallel inference of the ensemble models will ensure timely

misbehavior detection in VEHIGAN. However, if the system does not have the computational capability to run inference in parallel, the lightweight TFLite models can be adopted, which can provide similar performance at much lower overhead. As shown in Fig. 8b, the TFLite models only take less than 0.40ms for any $\mathcal{D}$, ensuring a swift detection. Although the additional layer increases the inference time slightly in TFLite models, it is negligible compared to the BSM transmission interval.

VI. Related Work

Several research explored different supervised ML-based MBDS for V2X [31]–[33]. They explored the effectiveness of common algorithms such as logistic regression, support vector machine, KNN, naive Bayes, random forest, decision tree classifier, etc., along with plausibility checks-based detectors and evaluated the performance of the existing misbehavior datasets. Ercan et al [33] 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 transformaer are also explored in [34], [35] but implemented on limited features, skewing their detection range. However, supervised models often encounter difficulties in 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 may not always faithfully represent real-world situations, exacerbating the generalization issue.

Sedar et al. assess the effectiveness of RL approaches for misbehavior detection in V2X scenarios, focusing on real-time position and speed patterns [36]. There are a few works on the ensembling approach for MBDS in V2X. [37] proposed a data-driven ensemble framework that combines KNN-based clustering and RL to detect misbehaviors in unlabeled vehicular data. It assesses performance in various attacks and highlights the potential challenges of inconsistent or mislabeled training data. Nevertheless, RL-based approaches require substantial labeled training data and computational resources and may not generalize well to real-life situations.

There are a few MBDS based on trajectory verification based on 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 [38]. Physical layer plausibility checks also seemed efficient. So et al. [39] introduced 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. Different statistical approaches in anomaly detection were also utilized in MBDS. Valentini et al. [40] used a statistical approach for anomaly detection in V2V communication. One downside of statistical approaches is that they mostly face limitations in identifying novel or zero-day attacks.

In contrast to the aforementioned studies, our approach in designing VEHIGAN incorporates several practical considerations. Firstly, VEHIGAN relies on GAN, an unsupervised DL model that doesn't necessitate labeled training data. Furthermore, VEHIGAN is efficient, adversarially robust, and versatile, which can seamlessly accommodate various types of features, making it effective against a broader spectrum of misbehaviors. Notably, we make our code and data available, whereas none of these prior works are reproducible as they did not share the code. Thus, we had to resort to common outlier detection algorithms for establishing baselines.

Individual GAN and their ensemble variants have been studied in different anomaly detection domains. 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 [41]. Similarly, Zhang et al. proposed a framework comprising multiple generators within the GAN architecture [42]. Han et al. advocated for a GAN framework consisting of multiple generators and discriminators, where each generator undergoes critique from every discriminator, and each discriminator evaluates synthetic samples from every generator [14]. While these approaches serve as motivation for our work, none have been tested on the V2X misbehavior datasets. Furthermore, 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.

VII. Conclusion

In this work, we leverage the potential of GAN to design an ensemble-based robust MBDS for V2X communications called VEHIGAN. The key elements of VEHIGAN are physics-guided feature engineering, training of diverse GAN models, and pre-evaluating and selecting top-performing GANs for the ensemble. For evaluation, we generate a comprehensive V2X misbehavior dataset and evaluate VEHIGAN against a diverse range of misbehaviors. Our comprehensive evaluation shows an ensemble-based VEHIGAN shows approximately 92% improvement in FPR under powerful adaptive attacker AFP attacks and inherent robustness against AFN attack. It outperformed baseline models in 20 out of 35 attacks and displayed similar performance in the remainder. Moreover, such VEHIGAN proved to be the most promising solution against the advanced misbehavior that manipulate multiple fields (such as heading & yaw rate) in the V2X messages simultaneously. Our finding emphasizes that GAN can be a potential tool for MBDS if the target V2X applications involve complex features like heading or if threat space is too complex for traditional detectors. Thus, this work advances the state-of-the-art by presenting GAN as a promising avenue for future MBDS research. To foster further research in this critical domain, we make both our code and datasets publicly accessible.

ACKNOWLEDGMENT

This work was supported in part by the US National Science Foundation under grants 1837519, 2235232 and 2312447, and by the Office of Naval Research under grant N00014-19-1-2621.

REFERENCES

[1] World Health Organization. Global status report on road safety, 2018.

[2] Santokh Singh. Critical reasons for crashes investigated in the national motor vehicle crash causation survey, 2018.

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

[4] V2X Core Technical Committee. *V2X Communications Message Set Dictionary*, sep 2023.

[5] Benedikt Brecht, Dean Therriault, André Weimerskirch, William Whyte, Virendra Kumar, Thorsten Hehn, and Roy Goudy. A security credential management system for v2x communications. *IEEE Transactions on Intelligent Transportation Systems*, 2018.

[6] Monowar Hasan, Sibin Mohan, Takayuki Shimizu, and Hongsheng Lu. Securing vehicle-to-everything (v2x) communication platforms. *IEEE Transactions on Intelligent Vehicles*, 2020.

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

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

[9] Mohammad Raashid Ansari, Jonathan Petit, Jean-Philippe Monteuuis, and Cong Chen. Vasp: V2x application spoofing platform. In *Proceedings Inaugural International Symposium on Vehicle Security & Privacy, ndss-symposium*, 2023.

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

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

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

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

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

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

[16] Rens W Van Der Heijden, Thomas Lukaseder, and Frank Kargl. 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, 2018.

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

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

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

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

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

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

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

[24] Andras Varga. Omnet++. In *Modeling and tools for network simulation*. Springer, 2010.

[25] Daniel Krajzewicz. Traffic simulation with sumo–simulation of urban mobility. *Fundamentals of traffic simulation*, 2010.

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

[27] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. 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, 2003.

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

[29] Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 2004.

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

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

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

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

[34] Hayotjon Aliev and HyungWon Kim. Misbehavior detection based on multi-head deep learning for v2x network security. In *International Conference on Consumer Electronics-Asia (ICCE-Asia)*. IEEE, 2021.

[35] Zhikang Liu, Hongyun Xu, Yong Kuang, and Feng Li. Svmdformer: A semi-supervised vehicular misbehavior detection framework based on transformer in iov. In *2023 IEEE 43rd International Conference on Distributed Computing Systems (ICDCS)*, pages 887–897. IEEE, 2023.

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

[37] Roshan Sedar, Charalampos Kalalas, Paolo Dini, Jesus Alonso-Zarate, and Francisco Vázquez-Gallego. Misbehavior detection in vehicular networks: An ensemble learning approach. In *Global Communications Conference*. IEEE, 2022.

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

[39] Steven So, Jonathan Petit, and David Starobinski. 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*, 2019.

[40] Edivaldo Pastori Valentini, Geraldo Pereira Rocha Filho, Robson Eduardo De Grande, Caetano Mazzoni Ranieri, Lourenço Alves Pereira, and Rodolfo Ipolito Meneguette. A novel mechanism for misbehaviour detection in vehicular networks. *IEEE Access*, 2023.

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

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