Simulation Framework of Misbehavior Detection and Mitigation for Collective Perception Services

Jiahao Zhang, Ines Ben Jemaa, Fawzi Nashashibi

Simulation Framework of Misbehavior Detection and Mitigation for Collective Perception Services

Jiahao Zhang*†, Ines Ben Jemaa*, Fawzi Nashashibi†
*IRT SystemX, †INRIA, France
{jiahao.zhang, ines.ben-jemaa}@irt-systemx.fr, fawzi.nashashibi@inria.fr

***Abstract*—Misbehavior detection which verifies the semantics of the V2X shared messages is a crucial research topic in Cooperative Intelligent Transport Systems (C-ITS). Misbehavior detection solutions aim to detect and identify the potential attackers which generate V2X messages with erroneous data. Providing efficient misbehavior detection solutions is even more challenging in the context of Cooperative Perception Services (CPS) in which communicating entities share their perception of the environment. This is because of the complexity of the attacks and the lack of the available experimental platforms that allow to evaluate and validate the misbehavior detection solutions. For these reasons, we propose a unified simulation framework to the research community that enables exploration and development of misbehavior detection and mitigation solutions as integrated parts of the CPS in various scenarios. We demonstrate the effectiveness of our framework in generating performance results and provide the corresponding datasets.**

***Index Terms*— C-ITS, Misbehavior Detection, Simulation, Collective Perception**

I. Introduction

Recently, much effort is conducted for the development of Cooperative Perception Services (CPS) in the research, industry and standardization communities. These services basically overcome the limitation of individual perception of an intelligent vehicle which is limited by the Field Of View of the local sensors and the occlusion scenarios. Sharing perceived objects improves not only the individual perception accuracy but increases its range by creating an extended perception of the environment. The extended perception data is an essential support for safety applications especially in challenging road environments such as intersections. The early stage work were based on sharing raw sensor data or perception models among autonomous vehicles. The main objective was mainly to deploy robust cooperative perception fusion algorithms among participating entities. Actually, with the progress of V2X technology, standardization entities such as ETSI deploy much effort to specify common data communication scheme for services such as cooperative perception. The data format is specified through the Collective Perception Message (CPM) as well as additional functionalities required for data transmission and reception [1]. However, the reliability of the CPS and its usefulness for safety application relies on the assumption of trustworthy collaborators. Consequently, one of the great challenges of the CPS system is to detect untrustworthy collaborators which have malicious behavior [2]. We call such malicious or *self-interest* behavior, a *misbehavior* and associate it to entities, which, while leveraging the shared information through V2X, disrupt the CPS by sending erroneous perception information. Thus, it is essential to tightly integrate misbehavior detection and mitigation functionalities to the CPS to ensure viable and resilient safety applications.

Besides some work [3] that experiment CPS in real world road scenarios, simulation remains a largely used alternative that allows test diversity, test repeatability and test flexibility. This is mainly needed in complex systems such as CPS, which require multidisciplinary knowledge on perception, fusion, communication, etc. and consequently heterogeneous technical solutions. Several existing simulation platforms recently couple simulators from various fields. However existing platforms tackle separately the CPS subsystems functionalities and do not integrate them in a unified framework. The challenge is even greater when the goal is simulate misbehavior on CP data and study the propagation of its impact on the safety application. In this work, we provide an open source simulation platform available in [4] that helps several research communities to work on a unified platform for cooperative perception services as specified recently by the standard [1]. We integrate to the platform misbehavior detection and mitigation solutions and complement it with a proof of concept of a safety application deployed in intersection areas. Our contributions are summarized as follows:

- We propose an open source unified simulation framework that supports the research on the CPS in general, and on misbehavior detection and mitigation for the CPS.
- We provide a benchmark results of several attack detection and reporting supported by a data set.
- We highlight the relevance of our unique misbehavior detection architecture [5] and show its relevance as a support for the safety application. We implement a Proof Of Concept of an Intersection Crossing Assist Application which uses the data resulted from the extended collective perception.

The rest of this paper is organized as follows: Section II presents the related work. Section III outlines our simulation framework architecture. Section IV details the integrated local misbehavior detection modules. Section V describes the external interfaces combined with the CARLA simulator. Section VI shows the experimental results. Finally, Section VII concludes the paper.

II. Related Work

To prove the feasibility of CPS, their potential and their limitation, extensive experimentation effort is needed. Real experimentation tests are crucial for the validation of the system [3]. However, they are often costly, limited and unrepeatable. Simulation is a relevant alternative that usually overcomes the limitation of real experimentation. Several simulation platforms exist in the literature. While early stage works were focusing on networking aspects, our approach's novelty is in tackling the Cooperative Perception as a perception/fusion problem, proving the benefits of collaborative perception for connected AVs. For instance, [6] proposes a simulation platform which utilizes Graph Neural Network to aggregate perception information in each autonomous vehicle. [7] provides an open source dataset that simulates several collaborative perception strategies among vehicles and RSU's. They use basically CARLA-Sumo simulation combined with several benchmark detection and perception techniques from the literature.

Simutack [8] presents an attack generation simulation platform targeting mainly local sensor attacks such as sensor jamming attacks. They demonstrate through their platform the impact of the attack on an autopilot controller application. The open source platform Veins [9] and its numerous extensions is a popular tool for V2X simulation based on the interconnection between Omnet++ and SUMO. Our previous work [10] extends Veins with additional functionalities of local and global misbehavior detection and generates massive datasets of misbehavior reporting [11].

Recently, much efforts were conducted on the Artery [12] simulation platform. Artery is a relevant candidate for V2X simulation as it provides an implementation of the ETSI communication stack including several functionalities of the service support layer. The work in [13] extends the Artery perception with sensor characteristics and measurement models to allow a close to reality perception simulation. A proof of concept of a simulation platform that combines Artery, SUMO and CARLA is presented in [14]. CARLA is bridged with a Robot Operating System (ROS) device to allow the simulation of ROS applications using the standardized ETSI messages. In our work, we build a unified simulation framework for misbehavior detection and reaction in the context of CPS based on Artery. We extend Artery with additional functionalities and couple it with external intersection support application to show the impact of misbehavior detection on safety.

III. Overview of the Simulation Framework Architecture

To simulate the whole architecture of misbehavior detection solutions on CPS, we use the existing Artery platform [12]. Artery is a simulation platform that combines SUMO, OMNet++ and additional libraries such as Vanetza. SUMO is a traffic simulator whereas OMNet++ is a communication simulator. Thanks to Vanetza, Artery is able to simulate the ETSI ITS communication stack. The implementation of the facility layer protocols is provided by Artery following the ETSI specifications. In addition, Artery adds some functionalities to the SUMO interface and provides a basic support of the sensor perception simulation. For these reasons, it is a suitable simulation platform candidate that corresponds to our testing and validation requirements. As shown in Fig.1, we extend Artery with additional modules and develop external interfaces to connect it with external modules. Especially, we develop an interface with the CARLA simulator and the Intersection Crossing Assist application [15].

The core implementation efforts are summarized as below:

- Adding the support of the CPS as lately specified and released by the latest ETSI standard [1].
- Adding the support of a global perception fusion module which generates an extended perception database in each vehicle equipped with the CPS and generating the corresponding dataset.
- Adding the support of misbehavior detection and mitigation modules and generating the misbehavior reporting dataset.
- Integrating our previous misbehavior detection framework F2MD [16] on Cooperative Awareness Messages (CAM) into Artery.
- Developing an interface that synchronizes CARLA and Artery simulation based on the initial effort of [17].

IV. The Misbehavior modules

A. The Attack Injection

This module contains the implemented attacks. The attack injection consists mainly in modifying the content of the perception list of CPM messages. We define three types of attacks as follows.

- Basic attacks: here, we implement attacks that modify the plausibility and the consistency of the perception data. Examples of attacks are Random, constant or shifted kinematic data of random or target perceived objects.
- Moderate attacks: in this category, the attacker sends perception data that are spatio-temporally plausible and consistent. However, it does not target a specific road situation but rather a random situation. Examples of moderate attacks are the injection of ghost objects in the scene or the omission of existing objects.
- Advanced attacks: The difference between the medium and the advanced attacks is that in this category, we implement attacks that target specific safety scenario. Examples of advanced attacks are the omission attacks and the injection attacks in intersection environments. The attacker sends plausible and consistent data, but his objective is to create a dangerous situation by sending a false perception scene data.

B. The Misbehavior Detection

Detection techniques rely on verifying the received perception data. We implement verification on several levels as follows.

- Source reliability verification: the objective of this step is to check whether the data received from one source has

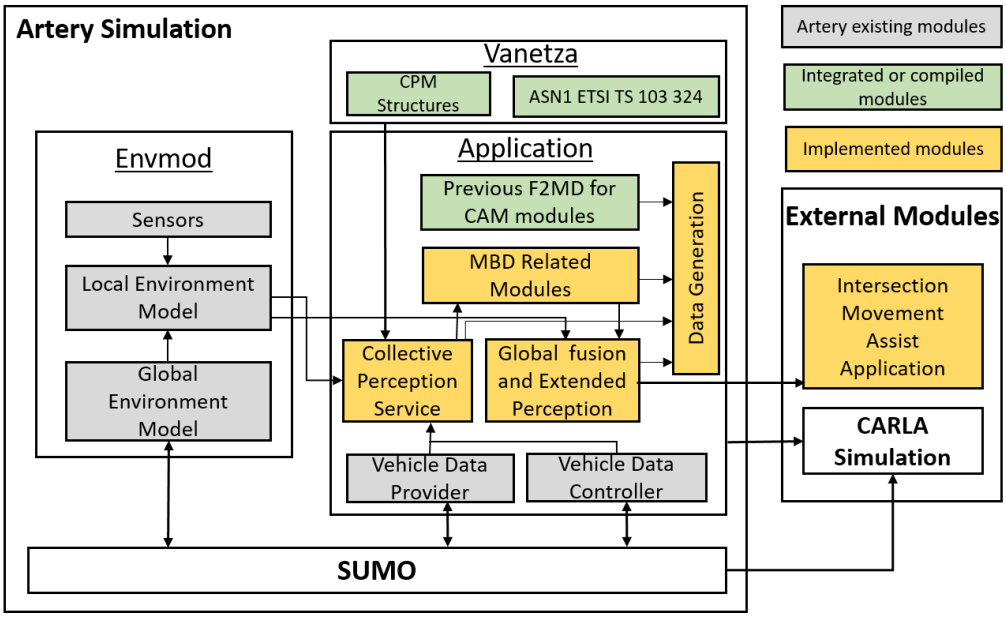


Fig. 1: Artery Architecture

to be processed or ignored. For instance, a transmitting source which is too far from the receiving node may not be considered because it is not persistently in the neighborhood.

- Data plausibility verification: This type of verification consists in checking whether the single attributes of the perceived objects are plausible or not in a received CPM. The verification is specifically focusing on comparing the attributes with pre-defined thresholds (signal-based) or known relations to other attributes (model-based).
- Data consistency verification: Consistency verification check if the received data in the actual CPM from one source are consistent with the past received data from the same source in a certain period of time. In this verification we use kinematic rules or filtering approaches such as Kalman Filtering.
- Data redundancy verification: Redundancy verification are based on the verification of data coming from several sources (i.e., nodes). The solution to this verification involves fusion approaches and or trust-based approach such as evidence theory and subjective logic. This step is very important and leads to merge the data coming from several sources to verify if there are some contradictory observations.

Additionally, we define a simulation validation strategy as shown in Table I by combining the attack type or severity as defined in Section IV-A with the road scenario complexity.

The *Basic test and validation* strategy requires to inject basic attacks in a simple road scenario such as highway scenarios or intersection scenarios. The *Large scale validation* requires to inject basic to moderate attacks in large scale road scenarios. The *General solution validation* requires to inject moderate attacks in simple to medium road scenarios. The

TABLE I: Simulation validation methodology

	Basic attacks	Moderate attacks	Advanced attacks
Simple road scenario	Basic validation	General solution validation	Specific safety scenario solution validation
Medium road scenario	Medium validation	General solution validation	Specific safety related solution validation
Complex road scenario	Large scale validation	Large scale validation	Not tested yet

Specific safety situation validation requires to inject advanced attacks in simple to medium road scenarios.

C. The Misbehavior Mitigation

As detailed in section IV-B, if the several CPM data verification lead to the identification of a given attacker entity, two misbehavior mitigation actions are triggered. The first mitigation action consists in sending a *a misbehavior report* to inform the back-end misbehavior authority about a suspicious attacker in the network. A misbehavior report is a message that contains the identity of the suspicious CPM source, the identity of the report generator and the evidences leading to the detected misbehavior. The misbehavior report follows a similar format as [18] with some adaptations that tailor it to the context of CPS. The misbehavior authority conducts additional report processing on all the received reports and generates a revocation request if needed. Notice that this mitigation measure is not performed in real time as the report processing may take some delay. Our contribution here consists in generating a massive report dataset that is useful for extensive processing and analysis at the misbehavior authority level.

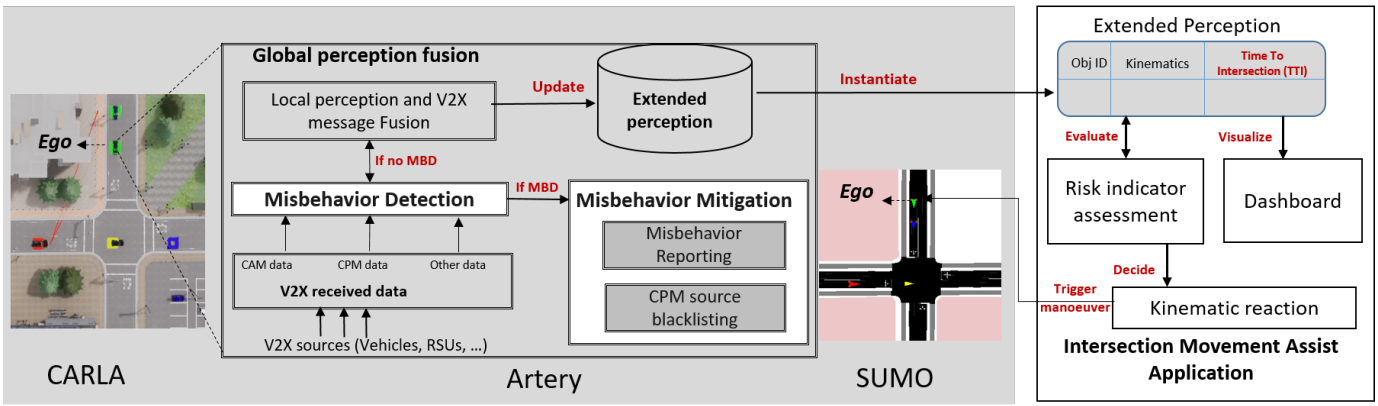


Fig. 2: Integration of the simulation platform with the external modules

The second mitigation action is undertaken in real time and aims at reacting immediately upon the detection of a misbehavior. As illustrated in 2, the misbehavior detection module is part of the global perception fusion process. It provides verified and reliable V2X perception data that are merged with the local perception of the receiving vehicle. This process generates consequently a consolidated perception of the environment called the extended perception. Once a misbehavior is detected on a given message, the source of the message is inserted to the list of the detected nodes (i.e, the blacklist) and its CPM message is ignored. Additionally, the following V2X messages received by the same misbehaving source are ignored.

V. The External Modules

A. The integration with CARLA

The integration between Artery and CARLA, as shown in Fig.2, aims to create a multidisciplinary simulation platform. CARLA is an open source graphical simulator which offers several simulation facilities for research on automotive sensor perception and control. We use an existing module that combines SUMO and CARLA and develop the interface between CARLA and Artery. At the time being, the interface is unidirectional and allows the synchronisation of the CPM messages and the detection of attackers through socket interface. We believe that this integration is beneficial to combine research works on perception and control on one side and on communication on the other side. In the future, we plan to use sensors and perception models from CARLA and integrate them to the Cooperative Perception Service in Artery. Another interesting benefit of this integration is to visualize the impact of using misbehavior detection on the safety application.

B. The Intersection Crossing Assist Application

The collective perception service is integrated with the Intersection Crossing Assist application through the extended perception list as shown in Fig.2. Specifically, the application accesses the data of the extended perception when it is updated through a specific interface. The update occurs for instance when a received CPM contains an object which is not known to the ego-vehicle and which is added to the extended perception. The extended perception list is generated in a fully simulated environment (i.e., the Artery simulation environment) whereas the application runs in a non-simulated environment. We use the socket programming interface to communicate between the client implemented in the Artery environment and the server that is implemented in the application side. The client is in charge of sending a temporal snapshot of the extended perception content at every update event to the application server. The risk indicator assessment functionality calculates, in our case, the Time-To-Intersection (TTI) for each vehicle object existing in the extended perception list. The TTI is the duration for a certain vehicle to reach the center of the intersection from its current location given its current velocity. The TTI is updated constantly for the objects existing in the extended perception. When the difference in TTIs reaches a certain threshold, the application triggers a braking manoeuvre by sending a signal to SUMO through the TRACI interface.

VI. Experimental Results

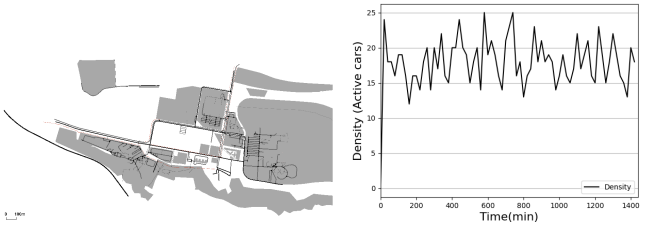
In this section, we evaluate the misbehavior detection verification on a large scale scenario and provide the corresponding datasets. We specifically consider basic attacks on perceived objects speed and position data.

A. Simulation Network

In this paper, we use the Paris Scalay network to validate our simulation framework as shown in Fig.3. This scenario contains a network size of 1.24 km^2 with a stable vehicle density of 18.2 vehicles/km^2. We use the randomly generated vehicle traces as the test benchmark. 80% of generated vehicles are equipped with the V2X service. We test three different attacker density, Low (5%), Medium (15%) and High (25%) with the same simulation seed. All simulation settings are shown in Tab II.

B. Attack Model

In our V2X attack model, we consider the internal attacks. This means that the attacker has the legitimate digital certificate that allows him to be authenticated and ensures the



(a) Paris Saclay Network (b) Paris Saclay Vehicle Density

Fig. 3: Network Description

TABLE II: Simulation Parameters

Simulation Duration	2h
Penetration Rate	0.8
Attacker Density	0.05 0.15 0.25
Scenario Size	1.24 km^2
Vehicle Density	18.2 Veh / km^2
Communication media	802.11p
Communication profile	ITS-G5
Communication type	Single Hop Broadcast
CPM interval	1 sec (fixed rate)
Front radar sensor	FoV range = 200m
	FoV angle = $\pm 20°$

integrity of his transmitted messages. We assume that the attacker has full control on the sensor data and it can modify the sensor measurements when encoding them in the CPM. The tested misbehavior in our simulations can be classified into two types. The first one is the *"random attack target"* chooses a random perceived object in each transmitted CPM and modifies its kinematic characteristics. The second type is the *"same attack target"* targets a given perceived object and constantly change its kinematic characteristics in all the transmitted CPMs. All the tested attack types are described as following:

- Alteration on the perceived object position
  1) Random position: For each transmitted CPM, the position is chosen with uniform distribution as a random point in the map (The range is the map size).
  $Position_x = U(Map\_X_{min}, Map\_X_{max})$
  $Position_y = U(Map\_Y_{min}, Map\_Y_{max})$
  2) Constant position: The position is a fixed value within the reasonable ego's perception range.
  $Const_x = U(0, max\_SensorRange_x)$
  $Const_y = U(0, max\_SensorRange_y)$
  $Position_x = Const_x$
  $Position_y = Const_y$
  3) Random position offset: For each transmitted CPM, add a noise to the actual distance data. The noise is obtained sampling from a gaussian distribution with $\mu = 0$ and $\sigma = \frac{max\_SensorRange}{10}$.
  $Position_x = current\_Position_x + N(0, \frac{max\_SensorRange}{10})$
  $Position_y = current\_Position_y + N(0, \frac{max\_SensorRange}{10})$
- Alteration on the speed
  1) Random speed: For each transmitted CPM, the speed is chosen from a uniform distribution.
  $Speed_x = U(0, Max\_Speed)$
  $Speed_y = U(0, Max\_Speed)$
  2) Constant speed: The speed is a fixed value within the max reasonable speed.
  $Const_x = U(0, Max\_Speed)$
  $Const_y = U(0, Max\_Speed)$
  $Speed_x = Const_x$
  $Speed_y = Const_y$
  3) Random speed offset: For each transmitted CPM, add a noise to the actual speed data. The noise is obtained sampling from a gaussian distribution with $\mu = 0$ and $\sigma = \frac{current\_Speed}{10}$.
  $Speed = current\_Speed + N(0, \frac{current\_Speed}{10})$

C. Local Detection Results

Table IV shows the local detection metrics at three different attacker densities. The detection quality is based on the performance of the misbehavior detection verification described in Section IV-B. The detection quality is evaluated using *Accuracy, Precision, Recall* and F_1score metrics. In order to calculate these metrics, the confusion matrix as shown in Table III is obtained based on the values of the True Positive (TP), the True Negative (TN), the False Positive (FP) and the False Negative (FN).

TABLE III: Confusion matrix

	Misbehaving	Genuine
Detected	TP	FP
Not Detected	FN	TN

- *Accuracy* is the ratio of correctly detected misbehaving vehicles to the total number of connected vehicles.
- *Precision* is the ratio of correctly detected misbehaving vehicles to the total reported misbehaving vehicles.
- *Recall* is the ratio correctly detected misbehaving vehicles to the total generated misbehaving vehicles.
- F_1score is the harmonic mean of the precision and recall.

First, we notice that the precision is very low in low attacker rate. The reason is that the default calibration of the detectors (i.e., the Kalman Filter settings) causes some systematic errors. Thus, the precision decreases when the generated attacker's rate decreases. We also notice that the *ConstSpeed* attack has generally the lowest detection performance. This is because the implemented attack injects consistent reasonable speed values. Data verification such as plausibility and consistency fails here to detect these attacks. The solution here is to perform data redundancy verification if several CPM sources transmit information about the same object. However, redundancy is very much dependant on higher network density and a high penetration rate. Last thing we notice is that the *Precision* is generally higher in *"random attack target"* type than in *"same attack target"* type. This means that it is hard to detect a consistent and plausible attack related to the same target object in consecutive CPMs.

Table V shows the total number of generated misbehavior reports at three different attackers densities, aggregated by the

TABLE IV: Testing Results

Attack Type		Results (Attacker rate 0.25)				Results (Attacker rate 0.15)				Results (Attacker rate 0.05)			
		Accuracy	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Random Attack Target	ConstPos	0.9382	0.9189	0..8252	0.8695	0.954	0.8655	0.824	0.8443	0.9612	0.5849	0.7561	0.6596
	ConstSpeed	0.8958	0.9	0.6553	0.7584	0.92	0.8132	0.6016	0.6916	0.96	0.5849	0.7381	0.6526
	RandomPos	0.9406	0.9198	0.8350	0.8753	0.9503	0.8522	0.8033	0.827	0.9673	0.6207	0.8780	0.7272
	RandomSpeed	0.9297	0.9157	0.7913	0.849	0.9564	0.8492	0.8629	0.856	0.9624	0.5893	0.8049	0.6804
	RandomPosOffset	0.9224	0.9127	0.7621	0.8307	0.943	0.8333	0.7724	0.8017	0.96	0.5741	0.7561	0.6526
	RandomSpeedOffset	0.9358	0.918	0.8155	0.8637	0.9479	0.8434	0.7951	0.8186	0.9673	0.6129	0.9268	0.7378
Same Attack Target	ConstPos	0.9394	0.9193	0.83	0.8724	0.9442	0.8291	0.7886	0.8083	0.9648	0.5964	0.85	0.7010
	ConstSpeed	0.8764	0.8881	0.5777	0.7	0.903	0.7647	0.52	0.619	0.9491	0.5	0.5476	0.5227
	RandomPos	0.9406	0.9198	0.835	0.8753	0.9539	0.8425	0.856	0.8492	0.9564	0.5484	0.8095	0.6538
	RandomSpeed	0.9382	0.9189	0.8252	0.8696	0.9455	0.8197	0.813	0.8163	0.96	0.5741	0.7561	0.6526
	RandomPosOffset	0.9261	0.9143	0.7767	0.8399	0.9467	0.839	0.7984	0.8182	0.9745	0.6842	0.9286	0.7879
	RandomSpeedOffset	0.9321	0.9167	0.801	0.8549	0.9515	0.8443	0.8306	0.8374	0.9624	0.5893	0.8049	0.6804

TABLE V: Total reports at different attacker densities

	Total Generated Reports		
	Low attacker density	Medium attacker density	High attacker density
Random Attack Target	6 740	15 722	24 446
Same Attack Target	5 866	15 092	24 271

type of attack target. As shown in Table IV, the similar results are observed. The type *"same attack target"* generates less number of reports. It also means that this type of attack is more challenging to detect due to the consistent and plausible injected data. It is unfortunate that because of the lack of a similar misbehavior detection algorithms in the literature, we are not able to compare our approach to existing ones.

VII. CONCLUSION

In this paper, we present a simulation framework of misbehavior detection and mitigation dedicated for Collective Perception Services. The platform combines the Artery simulator with the CARLA simulator and an external intersection assistance application. Through this platform, we were able first to demonstrate the feasibility of various types of attacks on CPM and to evaluate the performance of the implemented misbehavior solutions in several scenarios. Second we were able to show the impact of misbehavior detection solution as a mitigation module for a reliable and resilient intersection assistance application. Additionally, we provide a benchmark dataset of the misbehavior reporting and another dataset for extended perception in baseline and attack scenarios. For future work, we plan to update our simulation framework by integrating advanced misbehavior detection solutions, such as subjective logic for sophisticated attacks (e.g. omission and ghost attack). Furthermore, we plan to test our framework using other realistic road trajectories.

ACKNOWLEDGMENT

This work has been supported by the French government under the *France 2030* program, as part of the Technological Research Institute SystemX within the TAM project.

REFERENCES

[1] ETSI TS 103 324 V2.1.1, " Collective Perception Service," Standard, Jun. 2023.

[2] T. Huang, J. Liu, X. Zhou, D. C. Nguyen, M. R. Azghadi, Y. Xia, Q.-L. Han, and S. Sun, "V2x cooperative perception for autonomous driving: Recent advances and challenges," 2023.

[3] M. Shan, K. Narula, R. Wong, S. Worrall, M. Khan, P. Alexander, and E. M. Nebot, "Demonstrations of cooperative perception: Safety and robustness in connected and automated vehicle operations," *Sensors (Basel, Switzerland)*, vol. 21, 2020.

[4] J. Zhang and I. Ben Jemaa, "F2MD Github Repository," https://github.com/JiahaoZZhang/MBDSimulation.

[5] J. Zhang, I. B. Jemaa, and F. Nashashibi, "Trust management framework for misbehavior detection in collective perception services," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 596–603.

[6] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, J. Tu, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," 2020.

[7] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," 2022.

[8] A. Finkenzeller, A. Mathur, J. Lauinger, M. Hamad, and S. Steinhorst, "Simutack - an attack simulation framework for connected and autonomous vehicles," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023.

[9] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, 2011.

[10] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, "Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[11] F. Haidar, J. Kamel, I. B. Jemaa, A. Kaiser, B. Lonc, and P. Urien, "Dare: A reports dataset for global misbehavior authority evaluation in c-its," in *VTC2020-Spring*, 2020, pp. 1–6.

[12] R. Riebl, H. J. Günther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET applications," *2015 International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2015*.

[13] A. Willecke, C. Yazici, K. Garlichs, and L. C. Wolf, "Towards improving realism of perception in artery," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*.

[14] C. Anagnostopoulos, C. Koulamas, A. Lalos, and C. Stylios, "Open-source integrated simulation framework for cooperative autonomous vehicles," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022.

[15] 5GAA, "C-V2X Use Cases and Service Level Requirements Volume I," Standard, Jan. 2023.

[16] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. Ben Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," *IEEE Transactions on Vehicular Technology*, 2020.

[17] M. Bouchouia, J.-P. Monteuuis, H. Labiod, W. B. Jaballah, and J. Petit, "A simulator for cooperative and automated driving security," 2022.

[18] ETSI TS 103 759 V2.1.1 (2023-01), "Intelligent Transport Systems (ITS); Security; Misbehaviour Reporting service; Release 2," Standard, Jan. 2023.