DARE: A Reports Dataset for Global Misbehavior Authority Evaluation in C-ITS

Farah Haidar, Joseph Kamel, Ines Ben Jemaa, Arnaud Kaiser, Brigitte Lonc, Pascal Urien

DARE: A Reports Dataset for Global Misbehavior Authority Evaluation in C-ITS

Farah HAIDAR
Renault
Guyancourt, France
farah.haidar@renault.com

Joseph KAMEL
IRT SystemX
Palaiseau, France
joseph.kamel@irt-systemx.fr

Ines Ben Jemaa
IRT SystemX
Palaiseau, France
ines.ben-jemaa@irt-systemx.fr

Arnaud Kaiser
IRT SystemX
Palaiseau, France
arnaud.kaiser@irt-systemx.fr

Brigitte LONC
Renault
Guyancourt, France
brigitte.lonc@renault.com

Pascal Urien
Telecom ParisTech
Paris, France
pascal.urien@telecom-paristech.fr

***Abstract*—European and North American governments are actively working on improving road safety and traffic efficiency. To this end, their corresponding standardization bodies: ETSI and IEEE are developing the Cooperative Intelligent Transport Systems (C–ITS). In this system, vehicles and road side units communicate in order to enable new services and propose cooperative safety applications. However, the system is vulnerable to new types of threats if not adequately secured. The security and privacy protection is crucial to the user acceptance of such new system. Currently, the ETSI and IEEE proposed using a specific vehicular Public Key Infrastructure (PKI) to protect the C–ITS system. The PKI can protect the system against external attackers but it still vulnerable to internal attacks. Registered vehicles with valid certificates can still disturb the system by misusing its applications. The aim of misbehavior detection is to detect and mitigate the effect of internal attackers. The current misbehavior detection architecture includes a local embedded component and a cloud component. In this paper, we propose a misbehavior reports dataset of derived from the local embedded detection of misbehaving entities. This dataset can be used to further develop and evaluate the cloud component. The set includes different road topology, varying attacker penetration rates and attack scenarios.**

***Index Terms*—Misbehavior Detection, Dataset, C–ITS**

I. Introduction

C–ITS is an ongoing technology that will change our driving experience in the near future. This system is based on the cooperation of Intelligent Transport Systems (ITS) Stations (ITS–Ss). These stations such as On–Board Units (OBUs) on vehicles or Road–Side Units (RSUs) send and receive Vehicle–to–Everything (V2X) messages over the vehicular network. The safety messages could be regular broadcast (e.g. position, speed, acceleration) or specific warnings (e.g. road works, emergency break). Safety applications use these messages to detect and avoid dangerous situations on time. The security of V2X communications is currently only based on the use of a vehicular PKI. The PKI that delivers digital certificates to the local stations used to sign the transmitted messages. The digital certificates called also pseudonyms are used to authenticate the communicating ITS–S.

Consequently, the PKI enables the protection of the vehicular network against external attackers (i.e. ITS–S with no valid certificates or key materials). However, the system is still vulnerable against internal attackers. An ITS–S with valid certificates can still perform an attack. Moreover, insider security threats on C–ITS have been demonstrated by researchers [1] are currently demonstrated by Field Operational Tests (FOTs) [2]. Misbehavior detection is considered as the solution to the insider attacker problem. Multiple misbehavior detection systems for C–ITS have been proposed in the literature. However, the comparative evaluation of these solutions is difficult due to the lack of publicly available logs or datasets.

In the current literature, misbehavior detection includes: a local embedded component on board vehicles and a global component in the cloud. In this paper, we publish the misbehavior reports logs used to evaluate global component of the detection systems. The dataset consists of reports of misbehaving entities collected from individual ITS–S performing simple plausibility checks. The dataset includes multiple scenarios and parameters that could help researchers evaluate their solutions for different use cases. This dataset was also used in our previous work, where we proposed a Misbehavior Authority (MA) architecture. This MA system used machine learning to investigate and decide on the correct reaction to mitigate attacks [3] [4]. The publication of this dataset will enable researchers to better evaluate our systems and to improve and adequately compare future proposals.

The remainder of the paper is structured as follows. Section II presents the related works. Section III presents the C–ITS architecture, a misbehavior detection overview and the attacker model. Section IV details the proposed Dataset format, categories and parser. And section V concludes the paper.

II. Related Work

Similarly to other ad-hoc networks, C–ITS is vulnerable to attacks such as sybil, message falsification. Due to the sensitive nature of the C–ITS applications relating to the user's safety or privacy, an adequate misbehavior detection system is crucial. Recently, after the demonstration of some of these attacks, standardization bodies became more conscious about the importance of misbehavior detection in C–ITS.

In the US, the Institute of Electrical and Electronics Engineers (IEEE) proposed a PKI that considers a misbehavior authority composed of: a global misbehavior detection entity and a Certificate Revocation List (CRL) generator. Their goal is to publish a list of non-trusted entities. In the EU, the European Telecommunications Standards Institute (ETSI) is working on the standardization of the misbehavior authority. Many challenges still face this task such as the definition of reporting protocol, report format and the specification of the global investigation process. With the most critical part being the protection of the user's privacy in the overall procedure.

Numerous misbehavior detection systems are available in the literature. However, relative effectiveness of a certain solution compared to other is seldom correctly evaluated. To this end researchers require an adequate dataset for comparable evaluations. Multiple types of datasets exist in the literature described in [5], they can be divided into two categories described below:

- Real-world datasets: these datasets collected using digital video cameras or deployment projects are particularly valuable due to the unprecedented level of detail and accuracy. Their limitation however is the difficulty to capture cases of misbehavior detection. E.g it is difficult legally to deploy real misbehaving entities on the road.
- Core simulation datasets: these datasets rely on software implementations of mathematical models that replicate fundamental driver behavior logic. For example, SUMO and VEINS are well developed simulation software often used to test vehicular solutions.

Gozálvez et al. presents a field test campaign as part of the iTETRIS European research project [6]. Their work is an example of a deployment project dataset. The project aims at testing the quality of IEEE 802.11p Vehicle to Infrastructure communications. Their dataset includes 22 different RSU broadcast messages to a vehicle moving in an urban environment. Additionally, it contains the local positioning information and the Received Signal Strength Information (RSSI) of the received messages for different positions of the deployed vehicles. However, this dataset includes only normal behavior without any attacker data, which is restrictive for misbehavior detection evaluation. In our proposed dataset, we have normal and misbehaving entities. Which can be used to evaluate misbehavior schemes.

Van der Heijden et al. propose a Vehicular Reference Misbehavior Dataset (VeReMi) for local misbehavior detection validation [7]. Their work is an example of a simulation dataset. They used Vehicles in Network Simulation (VEINS), a co-simulation of SUMO and OMNET++. The dataset consists of message logs for every vehicle in the simulation (vehicle position and speed and the message RSSI). The number of vehicles, the number of attackers, as well as the variation of attacker rates, and many other parameters are also provided.

In our previous work, similarly to VeReMi, we also use VEINS to test our proposed misbehavior detection system [3] [4]. However, VeReMi was specific for local misbehavior detection component and our solutions targeted the global detection component. Therefore, we resolved to create a new dataset tailored for testing the global detection component. In this work, we publish the global misbehavior reports dataset used in our previous work. The dataset includes multiple scenarios, attacker percentage and vehicle densities. This work can help other researchers working on this topic to test their global misbehavior detection solutions and compare their contributions to ours.

III. System Model

A. C–ITS General Architecture

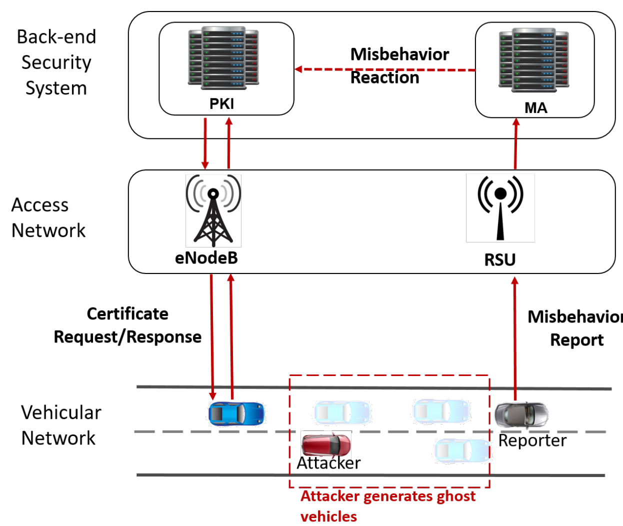


Fig. 1: C–ITS security architecture

Figure 1 presents the overall system architecture in the case of a presence of an attacker. In the current C–ITS system, ITS Station (ITS–S) like vehicles and RSUs cooperate by exchanging messages. Only the stations with valid certificates are allowed to exchange valid messages. However, in the current security system, a station with a valid certificate is still able to forge messages by generating and adding fake information. The C–ITS system is composed of the following components:

- *Vehicular Network* consist of communicating ITS–Ss. Vehicles generally exchange messages that serves various purposes in insuring road safety. For instance a vehicle could send beacon messages such as CAMs or BSMs contain kinematic information (position, heading, velocity etc...) to inform other vehicles of its presence on the road. However, an attacker could send these messages with fake information. The attack could be performed by one or multiple collaborating vehicles. Many attack use cases exist with varying motivations and capabilities.
- *Access Network* is the layer used relay local information to the back-end system in the cloud. This information

includes certificate requests to the PKI and misbehavior reports to the MA. These relays could rely on cellular connectivity (eNodeB) or the ITS-G5 (RSU).
- *Back-end Security System* is currently composed of the Public Key Infrastructure (PKI) and the Misbehavior Authority (MA). The PKI delivers digital certificates to the vehicles. The MA collects the Misbehavior Reports (MBRs) form local vehicles then investigates and issues the suitable reaction.

B. Misbehavior Detection Overview

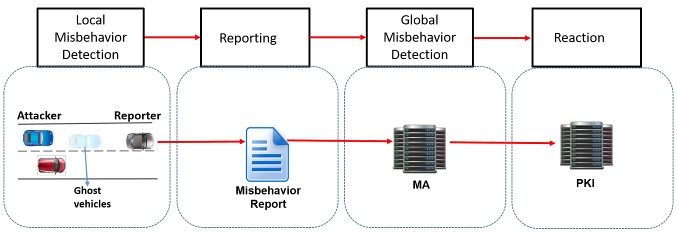


Fig. 2: Misbehavior detection steps

Figure 2 shows the misbehavior detection process. This process is divided into four steps:

1) Local detection: the local misbehavior detection is performed by every vehicle. Received messages should pass a set of simple and fast checks to estimate their plausibility and consistency. The results of these checks is used to decide if the vehicle should issue a misbehavior report [8].

2) Misbehavior reporting: the reporting process begins when an implausibility or inconsistency is detected in the local detection. The vehicle collects evidence, prepares and sends the reports to the MA (see Section III-C3).

3) Global Misbehavior detection: the global MA is responsible of collecting and analyzing of the received MBRs. Using the evidence in the MBRs the MA should be able to recreate the local events and determine if a vehicle is genuine or misbehaving. The severity and the type of misbehavior determines the suitable reaction required to protect the system.

4) Misbehavior reaction: the issued reaction is transmitted to the enforcing entity. Currently, the only discussed reaction type is the certificate revocation enforced by the PKI. The revocation and reaction procedure is still not well defined.

C. Local Detection

1) Local Detection Checks: local misbehavior detection is based on simple and fast to calculate checks on the data. These checks consist mostly of verifying the plausibility and consistency of some message field. In this study, 18 selected checks are executed simultaneously on every message. Here, we briefly summarize the general purpose of the implemented checks. In contrast, a more detailed and technical description could be found in our previous studies [9] [3]. Additionally, the implementation is open source and could be found on GitHub [10]. Here is the brief summary of the implemented checks:

- The *plausibility* of the claimed transmission range values, the position values and the speed values with respect to the physical limits of the environment.
- The *consistency* of the change for the position values, speed values and the speed and heading values with respect to the change in the claimed positions.
- The broadcasting *frequency* should correspond with the value determined by the standard in a certain situation.
- The *intersection* of the claimed multiple position values derived from multiple neighboring vehicles.
- The *sudden appearance* of one or multiple claimed neighbor vehicles within an unreasonable range.
- The consistency of the claimed neighbor's information with the predicted information from a *Kalman filer* [11].

2) Local Detection Applications: The local detection application is a layer used to evaluate the previously calculated checks and determine if a report to the global MA is required. For the generation of this dataset we use a non-cooperative trust based approach. The application follows a similar detection logic as [12] [13]. The current trust value is derived from the results of the plausibility checks (see algorithm 1). The final trust value is calculated based on the previous trust level combined with the previously calculated current trust. A report is deemed necessary if the global trust level falls below a certain value (see algorithm 1).

$$Trust(x) = -\frac{e^{(10\times(1-x))}+1}{2\times 10^4} \tag{1}$$

Algorithm 1: Non-Cooperative Trust Based Solution

c_x: Check Value, Tr: Threshold, T_G: Global Trust;
for $c_0 \dots c_n$ **do**
  if $c_i < c_{min}$ **then**
    $c_{min} = c_i$
  end
end
$T_I = Trust(c_{min})$
if $T_I > -\varepsilon$ ***and*** $T_G < 0$ **then**
  $T_G = T_G + 0.1$
else
  $T_G = T_G + T_I$
end
if $T_G < Tr$ **then**
  Misbehaving
else
  Genuine
end

3) Local Reporting Protocol: in this study we follow the misbehavior reporting protocol described in our previous work [14]. It is designed to include the necessary information for the back-end detection while simultaneously reducing network overhead (see Fig. 3). Briefly speaking, a vehicle performs the following actions upon detection of a malicious node:

i. Send an initial report including the necessary information described in Section IV-A.

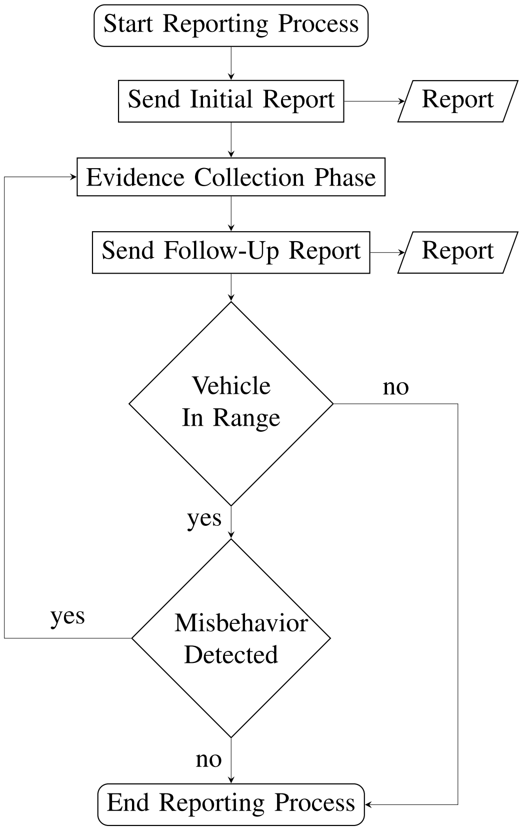


Fig. 3: Flowchart description of the reporting protocol

ii. Collect evidence for a predefined time period.
iii. Send a follow-up report with the collected evidence.
iv. According to the current status, if:
   a) A new anomalous behavior is detected while collecting evidence: go back to collecting evidence in step 2.
   b) The vehicle is out of range: End reporting process
   c) No new misbehavior is detected: End reporting process

D. Attacker Model and malfunctions

In this paper we consider the attacker as insider, i.e. the attacker is a registered vehicle with a set valid certificates. We also consider that the attacker has full unrestricted access his previously acquired certificates and could change it at will. Our dataset includes the types of misbehavior described and used in our previous studies. These types are enumerated here but for a more detailed description please refer to [3] [4]. These types are divided into malfunctions and malicious attacks. **Malfunctions:**

- *Position Malfunction*: the broadcasted position could be fixed, random, modified with a fixed offset or modified with a random offset.
- *Speed Malfunction*: the broadcasted Speed could be fixed, random, modified with a fixed offset or modified with a random offset.
- *Delayed messages*: the broadcasted information is correct, however it is sent with a multi-second delay.

Attacks:

- *Sudden Stop*: the attacker simulates a fake sudden stop.
- *Denial of Service*: the attacker unnecessarily and maliciously increases the message broadcasting frequency.
- *Data Replay*: the attacker chooses a genuine target vehicle as victim and instantly replays their messages data.
- *Disruptive*: the attacker randomly replays messages from the previously received messages with neighbors data.
- *Random*: the attacker uses their previously acquired and valid certificates to sign and maliciously transmit messages with false random data.
- *Sybil*: the attacker gains access to its local certificate library then maliciously changes the used pseudonym.
- *Traffic Congestion*: the vehicle uses its or a neighbor position to simulate a grid like fake traffic congestion.
- *Combination of previous attacks*: The combinations include *DoS-Disruptive*, *Dos-Random*, *Dos-Random-Sybil*, *DoS-Disruptive-Sybil* and *DataReplay-Sybil*.

IV. Proposed Dataset Description

A. Dataset Format

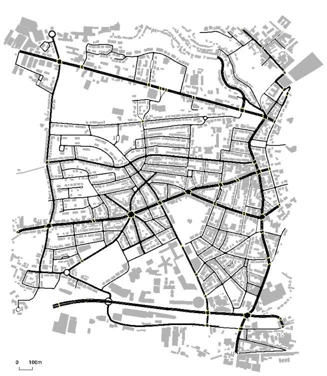
Our proposed dataset includes the initial reports and follow-up as described in III-C3. Both types of reports are encoded in JavaScript Object Notation (JSON), a lightweight data-interchange format. The format has the following description:

```
{"Report":{
  "Metadata":{
    "senderId":ℤ[0,+∞],
    "reportedId":ℤ[0,+∞],
    "generationTime":ℝ[0,+∞],
    "senderRealId":ℤ[0,+∞],
    "reportedRealId":ℤ[0,+∞],
    "attackType":"String"
  },
  "Messages":[
    {
      "CreationTime":ℝ[0,+∞],
      "Pos":[ℝ[−∞,+∞],ℝ[−∞,+∞],ℝ[−∞,+∞]],
      "Speed":[ℝ[−∞,+∞],ℝ[−∞,+∞],ℝ[−∞,+∞]],
      "Accel":[ℝ[−∞,+∞],ℝ[−∞,+∞],ℝ[−∞,+∞]],
      ...
    },
    ...
  ]
  "Checks":[
    {
      "rangePlausibility":ℝ[−∞,+1],
      "posPlausibility":ℝ[−∞,+1],
      "posConsistency":ℝ[−∞,+1],
      "speedPlausibility":ℝ[−∞,+1],
      ...
    },
    ...
  ]
}}
```

TABLE I: DARE dataset specifications for each scenario

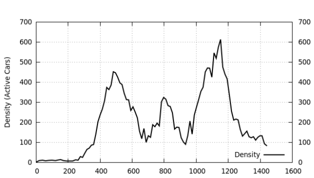
Id	Scenario		Attacker				Genuine		Reports		File Size	
	Net	*Time*	*Rate*	*Type*	*Vehicles*	*Messages*	*Vehicles*	*Messages*	*Initial*	*Follow-up*	*Plain*	*Zipped*
Lust-0024-25S	Lust	0h-24h	25%	Shuffle-All	20,318	109,823,057	61,607	223,214,564	8,862,467	19,972,105	167.1GBs	14.41GBs
Lust-0611-25M	Lust	6h-11h	25%	Malfunctions	6,556	27,677,626	19,845	82,911,912	566,002	2,482,071	29.71GBs	2.983GBs
Lust-0611-25A	Lust	6h-11h	25%	Attacks	6,560	52,930,127	19,845	85,646,078	5,828,229	11,339,048	91.24GBs	7.537GBs
Lust-0611-15M	Lust	6h-11h	15%	Malfunctions	3,937	16,815,969	22,491	93,773,569	391,395	1,714,996	20.69GBs	2.072GBs
Lust-0611-15A	Lust	6h-11h	15%	Attacks	3,933	32,977,691	22,491	95,319,109	3,957,115	7,802,816	64.14GBs	5.368GBs
Lust-0611-05M	Lust	6h-11h	5%	Malfunctions	1,314	5,640,072	25,137	104,949,466	154,812	661,708	7.849GBs	0.795GBs
Lust-0611-05A	Lust	6h-11h	5%	Attacks	1,307	11,231,920	25,137	105,539,193	1,424,943	2,773,928	23.41GBs	1.973GBs
Lust-1115-25M	Lust	11h-15h	25%	Malfunctions	3,757	9,311,363	11,353	27,619,797	201,393	859,738	10.13GBs	1.006GBs
Lust-1115-25A	Lust	11h-15h	25%	Attacks	3,748	18,298,605	11,353	28,558,086	2,049,910	3,966,993	30.673GBs	2.471GBs
Lust-1115-15M	Lust	11h-15h	15%	Malfunctions	2,255	5,547,090	1,760	31,384,070	138,351	582,879	6.885GBs	0.679GBs
Lust-1115-15A	Lust	11h-15h	15%	Attacks	2,250	10,883,990	12,867	31,936,083	,1325,323	2,561,363	20.23GBs	1.649GBs
Lust-1115-05M	Lust	11h-15h	5%	Malfunctions	751	1,788,546	14,381	35,142,614	52,958	215,489	2.562GBs	0.247GBs
Lust-1115-05A	Lust	11h-15h	5%	Attacks	751	3,548,232	14,381	35,333,220	420,467	854,075	6.966GBs	0.581GBs
Lust-1521-25M	Lust	15h-21h	25%	Malfunctions	7,776	34,550,797	23,475	104,635,169	685,534	3,144,205	37.35GBs	3.773GBs
Lust-1521-25A	Lust	15h-21h	25%	Attacks	7,766	68,001,064	23,482	108,001,498	7,132,715	15,683,151	117.4GBs	9.486GBs
Lust-1521-15M	Lust	15h-21h	15%	Malfunctions	4,651	20,918,031	26,605	118,267,935	472,852	2,164,576	25.65GBs	2.577GBs
Lust-1521-15A	Lust	15h-21h	15%	Attacks	4,661	40,517,058	26,612	120,265,804	4,487,167	9,837,161	79.67GBs	6.669GBs
Lust-1521-05M	Lust	15h-21h	5%	Malfunctions	1,549	6,930,299	29,735	132,255,667	183,613	820,235	9.844GBs	0.969GBs
Lust-1521-05A	Lust	15h-21h	5%	Attacks	1,550	13,183,858	29,735	132,950,041	1,550,561	3,517,203	28.72GBs	2.403GBs
Paris-0024-05S	Paris	0h-24h	5%	Shuffle-All	619	635,311	11,914	7,830,985	60,788	123,413	1.031GBs	0.091GBs

B. Datasets Networks

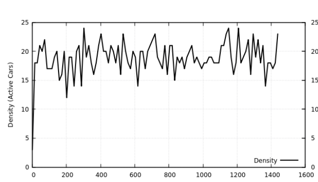


(a) Luxembourg city Network

(b) Paris Saclay Network



(c) Luxembourg Vehicle Density



(d) Paris Vehicle Density

Fig. 4: Networks Description

In this dataset, we use two networks from different European cities: Luxembourg and Paris Saclay (Figure 4). The vehicle traces of the Luxembourg city provided by the vehicular lab of the university of Luxembourg [15]. Luxembourg SUMO Traffic (LuST) is a synthetic set validated with real data [16]. The selected part of the LuST scenario network is $6.51km^2$ and of peak density of $104.5Vehicle/km^2$. The vehicle traces for the Paris Saclay scenario are randomly generated. It is mainly used as a test-Bench. This scenario has a network size of $1.11km^2$ and semi constant density around $17.1Vehicle/km^2$. The dataset is generated using the F^{2}MD framework [17]. F^{2}MD is a VEINS [18] extention, an open source framework for vehicular network simulations. VEINS is built on a network simulator (OMNeT++) and a road traffic simulator (SUMO). For further technical details, the F^{2}MD source code and configuration of the described scenarios are available on our GitHub [10].

C. Datasets Categories

Table I shows all the subsets categories available in our dataset. In this dataset the evidence collection phase is set to 10 seconds. We mainly altered four variables: the vehicle traces, time of day, attacker rate and attacker type. The vehicle traces change between the Luxembourg and the Paris network. We generally use the Paris network as our Test-Bench. The changes in the time of day entails a change in the general vehicle density as shown in Figure 4c. The attacker rate is stated for every scenario. The attack launcher is designed to inject a new attacker when the rate drops below the set value. All the listed datasets could be downloaded individually from our Cloud Drive [19].

D. Local Detection Results

TABLE II: Dataset scenarios local detection

Id	Local Detection Metrics		
	Accuracy	F_1*Score*	*C's Kappa*
Lust-0024-25S	0.91379	0.86995	0.65324
Lust-0611-15M	0.98191	0.93798	0.86432
Lust-0611-15A	0.90524	0.82466	0.55098
Lust-1115-25A	0.89421	0.86547	0.61923
Lust-1115-05M	0.99183	0.91278	0.82463
Lust-1521-25M	0.97060	0.93797	0.85216
Lust-1521-05A	0.96135	0.79448	0.54616
Paris-0024-05S	0.98840	0.91767	0.83622

Table II shown the local detection metrics for some scenarios. The list of all the local detection results with more detection metrics is included with the dataset. The detection quality is based on the detection checks and applications described in Section III-C. To evaluate the detection quality, we used the *Accuracy*, F_1*Score* and *Cohen's kappa*. The *Accuracy* is the rate of positive agreement, which in our case refers to the ratio of true detection in the system. The F_1*Score* is also a measure of accuracy, however it is a more suitable for unbalanced sets (i.e. the set includes more genuine vehicles than attackers). Finally, *Cohen's kappa* is a measure of the positive agreement, similar to the *Accuracy*, but where we subtract the agreement by chance.

These results show that the attacker rate and type alteration affect the local detection quality. Consequently, it also affects the number and quality of collected misbehavior reports. We suspect a relation between these variables and the global detection results. Accordingly, these metrics should be considered when analyzing results extracted from this dataset.

E. Datasets Parser

We provide a specific parser designed to read and store this type of data. This parser is supplied in order to facilitate the treatment and study of the proposed datasets. This parser is implemented in python to facilitate any machine learning application. The data is sorted automatically by reported pseudonym. A simple threshold application example is provided along with evaluation metrics and graph plotting mechanisms. This parser can also be found in open source format on our GitHub [10]

V. Conclusion and Future Work

Global Misbehavior detection in C–ITS is still a developing subject. The demand for robust and concrete detection solution is rising especially in ITS standardization working groups of the ETSI and IEEE. In this paper we provide a large dataset of misbehavior reports for comparative evaluation of global detection solutions. This dataset is aimed at facilitating and catalyzing the currently in demand work on the global side.

Future works, includes the development of a unified misbehavior authority framework. Additionally we have plans of deployment and testing of variants of global solutions on the C–ITS field tests in France

Acknowledgment

This research work has been carried out in the framework of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program *Investissements d'avenir*.

References

[1] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in vanets," 2004.

[2] Charlie Miller, Chris Valasek, "Adventures in automotive networks and controls units," in *DefCon 21 - Las Vegas, Nevada*, 2013, pp. 1–4.

[3] J. Kamel, F. Haidar, I. B. Jemaa, A. Kaiser, B. Lonc, and P. Urien, "A Misbehavior Authority System for Sybil Attack Detection in C-ITS," in *The IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference – IEEE UEMCON 2019*, New York, United States, Oct. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02316391

[4] I. Mahmoudi, J. Kamel, I. B. Jemaa, A. Kaiser, and P. Urien, "Towards a Reliable Machine Learning Based Global Misbehavior Detection in C-ITS: Model Evaluation Approach," in *Third International Workshop on Vehicular Adhoc Networks for Smart Cities – IWVSC 2019*, Paris, France, Nov. 2019.

[5] Department of Transportation (DoT) - USA, "Next generation simulation (ngsim) vehicle trajectories and supporting data," 2006.

[6] J. Gozálvez, M. Sepulcre, and R. Bauza, "IEEE 802.11p vehicle to infrastructure communications in urban environments," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 176–183, 2012.

[7] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," in *Security and Privacy in Communication Networks*, R. Beyah, B. Chang, Y. Li, and S. Zhu, Eds. Cham: Springer International Publishing, 2018, pp. 318–337.

[8] J. Kamel, I. Ben Jemaa, A. Kaiser, L. Cantat, and P. Urien, "Misbehavior detection in C-ITS: a comparative approach of local detection mechanisms," in *2019 IEEE Vehicular Networking Conference (VNC) (IEEE VNC 2019)*, Los Angeles, USA, Dec. 2019.

[9] J. Kamel, A. Kaiser, I. Ben Jemaa, P. Cincilla, and P. Urien, "CaTch: a confidence range tolerant misbehavior detection approach," in *2019 IEEE Wireless Communications and Networking Conference (WCNC) (IEEE WCNC 2019)*, Marrakech, Morocco, Apr. 2019.

[10] J. Kamel, "Github repository: Framework for misbehavior detection (f^2md)," 2019. [Online]. Available: https://github.com/josephkamel/f2md

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

[12] R. K. Schmidt, T. Leinmller, E. Schoch, A. Held, , and G. Schaefer, "Vehicle behavior analysis to enhance security in vanets," in *V2VCOM 2008*, 2008, pp. 1–8.

[13] N. Bimeyer, C. Stresing, and K. M. Bayarou, "Intrusion detection in vanets through verification of vehicle movement data," in *2010 IEEE Vehicular Networking Conference*, Dec 2010, pp. 166–173.

[14] J. Kamel, I. Ben Jemaa, A. Kaiser, and P. Urien, "Misbehavior reporting protocol for c-its," in *2018 IEEE Vehicular Networking Conference (VNC)*, Dec 2018, pp. 1–4.

[15] VehicularLab. University of luxembourg. [Online]. Available: http://vehicularlab.uni.lu

[16] L. Codeca, R. Frank, and T. Engel, "Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research," in *IEEE Vehicular Networking Conference (VNC)*, Dec 2015, pp. 1–8.

[17] Framework For Misbehavior Detection (F^2MD). (2019) F^2MD website. [Online]. Available: https://www.irt-systemx.fr/f2md

[18] 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, no. 1, pp. 3–15, Jan 2011.

[19] Kamel, Joseph, "Dare: A reports dataset for global misbehavior authority evaluation in c-its," 2020. [Online]. Available: https://github.com/josephkamel/DARE-Dataset