VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs

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

Michael Wolf
Institute of Distributed Systems
University of Ulm, Germany
michael.wolf@uni-ulm.de

Rens W. van der Heijden
Institute of Distributed Systems
University of Ulm, Germany
rensvdheijden@gmail.com

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

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

Frank Kargl
Institute of Distributed Systems
University of Ulm, Germany
frank.kargl@uni-ulm.de

***Abstract*—Cooperative Intelligent Transport Systems (C–ITS) is a new upcoming technology that aims at increasing road safety and reducing traffic accidents. C–ITS is based on peer-to-peer messages sent on the Vehicular Ad hoc NETwork (VANET). VANET messages are currently authenticated using digital keys from valid certificates. However, the authenticity of a message is not a guarantee of its correctness. Consequently, a misbehavior detection system is needed to ensure the correct use of the system by the certified vehicles. Although a large number of studies are aimed at solving this problem, the results of these studies are still difficult to compare, reproduce and validate. This is due to the lack of a common reference dataset. For this reason, the original VeReMi dataset was created. It is the first public misbehavior detection dataset allowing anyone to reproduce and compare different results. VeReMi is used in a number of studies and is currently the only dataset in its field. In this Paper, we extend the dataset by adding realistic a sensor error model, a new set of attacks and larger number of data points. Finally, we also provide benchmark detection metrics using a set of local detectors and a simple misbehavior detection mechanism.**

***Index Terms*—Misbehavior Detection, Intelligent Transport Systems, Vehicular Networks, Dataset**

I. Introduction

Improving road safety is a major challenge currently addressed by European and North American governments. Their standardization bodies, the European Telecommunications Standards Institute (ETSI) and Institute of Electrical and Electronics Engineers (IEEE) are working on a novel solution in the form of the C–ITS. This system is based on the exchange of safety messages between different Intelligent Transport Systems (ITS) Stations (ITS–Ss) over the VANET. The security of the C–ITS messages are critical to the operation of the system. As a result, both the ETSI and IEEE standardized and mandated the use of a vehicular Public Key Infrastructure (PKI). The PKI is tasked with distributing digital certificates to the local vehicles. The vehicles use the digital certificates to sign messages exchanged over the VANET. These signatures ensure the authenticity of the message's sender and enables receivers to verify the sender's identity. Nevertheless, the authenticity of a message does not ensure the correctness of the included information. In other words, internal ITS–Ss with valid certificates could misuse the safety applications by sending inaccurate or fake information. To protect the system and mitigate the effect of these stations we employ a misbehavior detection system.

Misbehavior detection for C–ITS is a well research subject with a large number of published solutions and results. However, due to the lack of a reference dataset, many of these results are not reproducible or verifiable. To this end, the original VeReMi was published as the first public dataset for vehicular misbehavior detection [1]. VeReMi has proven to be very useful for researchers in this domain, being applied in multiple studies [2]–[5]. Nevertheless, the VeReMi dataset still has room for improvement, especially considering the small number of attacks and the lacking physical error model.

In this paper, we aim to upgrade VeReMi by treating these issues. Accordingly, we devise and implement a realistic sensor error model on the vehicle's physical layer. Moreover, we implement a larger more complex set of attacks. The new attacks enable the manipulation of the message frequency and the digital certificates as well as the manipulation of the message contents. Finally, we describe and implement a set of local plausibility detectors and a simple fusion detection mechanism. The dataset is then tested against this misbehavior fusion mechanism and the results are provided as a benchmark for future researchers.

The remainder of the paper is organized as follows: Section II discusses the related works. Section III presents the current C–ITS system architecture with the overall misbehavior detection process. Section IV details the provided dataset components. Section V provides testing and experimental results. Finally, section VI concludes this work and provides some future perspectives.

II. Related Works

Multiple studies used the original VeReMi dataset to test and validate their various misbehavior detection mechanisms.

978-1-7281-5089-5/20/$31.00 ©2020 IEEE

Steven et al. [2] analyzed the dataset with the K-Nearest Neighbors (K-NN) and Support Vector Machine (SVM) Machine Learning (ML) classification algorithm with positive results. In their experiment, they only made minor adjustments to the labeling of the data in order to help with the classification, but the attacker types and messages were left unmodified. In their discussion, they pointed to a small weakness of VeReMi. The vehicles performing an *EventualStop* are constantly labeled as attackers, though they may behave normally for a certain time before the start of their malicious behavior. They suggest that the vehicle should be labeled as an attacker only when the attack is ongoing. This suggestion is retained and this issue is now resolved in this new VeReMi version. Attacker nodes are only labeled as such when actively performing a certain attack, otherwise their behavior is labeled as normal.

Singh et al. [3] used VeReMi to perform feature engineering for a ML misbehavior detection solution. They tested various feature selections and scored different results in accuracy. They performed their experiments with Logistic Regression and an SVM. Their best feature set was achieved with an SVM and contained the position, the speed and the difference between the position and speed of the sender and receiver. In the current version of VeReMi, the acceleration and heading are added as new features to the dataset. This could enable more feature combinations and a better feature selection in future studies.

Gyawali et al. [4] also used the dataset for an ML application using a Feed Forward Neural Network (FFNN) and an SVM. They calculated their solution's detection metrics including the accuracy, precision, recall and F_1Score. They then compared these metrics to the ones included in the original version of VeReMi. This type of comparison is still possible in the latest version of VeReMi as these detection metrics were calculated for each newly added scenario.

III. System Model

A. C-ITS Model

C–ITS is a new technology that aims to increase road safety. It is currently in the standardization phase in ETSI and the IEEE. The technology is based on the exchange of safety messages between the different ITS–Ss (e.g. vehicle's On-Board Unit (OBU), Road–Side Unit (RSU)). These messages include the Cooperative Awareness Messages (CAMs) [6] and Basic Safety Messages (BSMs) [7] proposed by the ETSI and IEEE respectively. Currently, both standards include a vehicular PKI system in place to issue digital keys in the form of certificates to the ITS–Ss. These keys are used to sign the exchanged messages thus validating the sender identity and ensuring the message authenticity. These signatures are a form of identification for a certain vehicle. Therefore, a vehicle signing all its messages with the same key would be easily trackable. As a result, the vehicular PKI issues multiple pseudonym certificates to the same vehicle to ensure user's privacy. A vehicle is therefore able to regularly change its signature to avoid tracking. This pseudonymization of the vehicles enables a new vector of attack where one vehicles pretends to be simultaneously multiple entities on the road. This type of malicious behavior is called a Sybil attack and is also included in this new version of VeReMi.

Although the vehicular PKI successfully protects the network against external attackers, the network is still vulnerable against internal malicious stations. A station with a valid set of keys could send erroneous information all while successfully signing the messages. Hence the need for a MisBehavior Detection (MBD) system capable of ensuring a level of semantic correctness of the exchanged messages.

B. MBD System Model

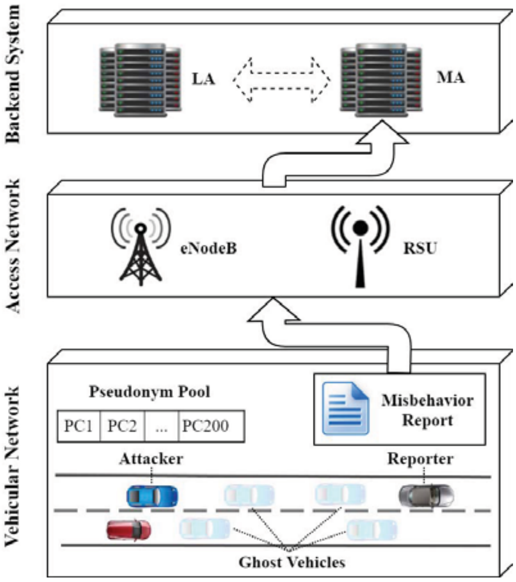


Fig. 1. System model

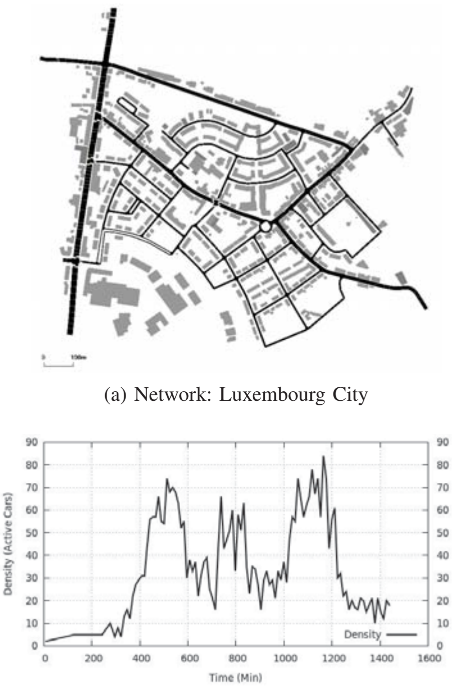
The current C–ITS MBD system model consists of the following consecutive steps (Figure 1):

1) **Local Detection** where every vehicle performs plausibility and consistency checks on every received message. These checks are then analyzed by a fusion application that determines if a vehicle is indeed misbehaving.
2) **Reporting process** is initiated if the fusion application determines that a vehicle is misbehaving. The receiver proceeds to collect evidence of the occurring misbehavior. The evidence is used to create a Misbehavior Report (MBR). The MBR is then sent to a global authority.
3) **Global investigation** is performed in the cloud by the misbehavior authority. The role of this authority is to first collect the MBRs issued by the local vehicles. These reports are then analyzed in order to determine the suitable reaction to protect the system.
4) **Reaction** is the process of executing the decision issued by the misbehavior authority in order to mitigate the effects of an attack. This execution is normally done by a third party. For example, if the reaction against a certain attack is a certificate revocation, the reaction is entrusted

to the PKI. Currently, the misbehavior reaction is still not well defined in the literature or the published standards.

IV. Dataset

A. Simulation Platform



(a) Network: Luxembourg City

(b) Vehicle Density

Fig. 2. Simulation Scenario

In order to generate our dataset, we make use of the Framework For Misbehavior Detection (F^{2}MD). F^{2}MD is a VEINS extension that enables the recreation and detection of various MBD use cases. VEINS [8], an open source simulator for Inter-Vehicular Communication is based on OMNeT++ [9] and SUMO [10]. OMNeT++ is a C++ simulation library for building network simulators and SUMO is a well-known open source road traffic simulation suite.

Additionally, we use the vehicle traces provided by the Luxembourg SUMO Traffic (LuST) scenario [11] which is an open source synthetic traffic scenario validated with real data provided by the VehicularLab of the University of Luxembourg. For our dataset, we use a subsection of the LuST network with a size $1.61km^2$ and a peak density of 67.4 Veh/km^2 (See Fig. 2).

B. Sensor Error Models

In this version of VeReMi, we aim to render the provided data more realistic and in line with real world field tests. Accordingly, we add sensor error models to the four main data fields: Position, Velocity, Acceleration and Heading.

1) Position Error: Several positioning systems exist with different levels of precision. These levels also differ by region, i.e. the Global Positioning System (GPS) precision is limited to about 3 to 5 m in open sky environments and up to 20 m in urban areas [12]. However, we consider an internal correction system on-board every vehicle [13]. Consequently the position error model is as follows:

P_t	$\triangleq$	Real Position at time t
$P_t^{\mathcal{E}}$	$\triangleq$	Broadcasted Position at time t
$\mathcal{E}_t^P$	$\triangleq$	Position Error at time t
U	$\triangleq$	Uniform Distribution
$\mathcal{N}$	$\triangleq$	Normal Distribution

$$\mathcal{E}_0^P = U([-5,5])$$
$$\mu = \frac{\mathcal{E}_0^P + \mathcal{E}_{t-1}^P}{2} \qquad \sigma = 0.03\mathcal{E}_0^P$$
$$\mathcal{E}_t^P = \mathcal{N}(\mu, \sigma^2)$$
$$P_t^{\mathcal{E}} = P_t + \mathcal{E}_t^P$$

2) Velocity Error: The majority of vehicles estimate velocity from the wheel spin with an average error around $0.05m/s$ [14]. The error is also proportional to the velocity. As a result the speed error model is as follows:

V_t	$\triangleq$	Real Velocity at time t
$V_t^{\mathcal{E}}$	$\triangleq$	Broadcasted Velocity at time t
$\mathcal{E}_t^V$	$\triangleq$	Velocity Error at time t

$$\mu = 0 \qquad \sigma = 0.00016$$
$$\mathcal{E}_0^V = \mathcal{N}(\mu, \sigma^2)$$
$$V_t^{\mathcal{E}} = V_t + V_t * \mathcal{E}_0^V$$

3) Acceleration Error: In our model, the acceleration error is inferred from the velocity error. Therefore the acceleration error model is as follows:

A_t	$\triangleq$	Real Acceleration at time t
$A_t^{\mathcal{E}}$	$\triangleq$	Broadcasted Acceleration at time t

$$A_t^{\mathcal{E}} = A_t + \frac{\mathcal{E}_t^V - \mathcal{E}_{t-1}^V}{\delta t}$$

4) Heading Error: The vehicle heading could be calculated using a magnetic compass or inferred from successive positions. The accuracy of a magnetic compass is dependent on the device quality. A study by Hölzl et al. found that the probability of having an error below 20° is around 85% for most mobile devices [15]. Whereas Deelertpaiboon et al. successfully equipped a vehicle with a magnetic compass with a $0.1°$ accuracy [16]. Additionally, accuracy of the heading derived from successive positions is dependent on the vehicle velocity. At a high velocity the position heading is more probable to have higher accuracy than the compass heading, whereas the opposite is true at lower speeds or when a vehicle is stationary. For our solution, we propose the following heading error model:

H_t	$\triangleq$	Real Heading at time t
$H_t^{\mathcal{E}}$	$\triangleq$	Broadcasted Heading at time t

$$\mathcal{E}_0^H = U([-20,20])$$
$$\mathcal{E}_t^H = \mathcal{E}_0^H * e^{-0.1*V_t}$$
$$H_t^{\mathcal{E}} = H_t + \mathcal{E}_t^H$$

C. *Misbehavior Models*

In this study, we also aim to expand the VeReMi attacks library with a set of new attacks aggregated from the models used in the literature [17]. We make the distinction between malfunctions and attacks. The former constitutes non-malicious behaviors that results from a malfunctioning OBU or vehicle sensors while the latter is malicious behavior of vehicles intentionally sending wrong information.

1) *Malfunctions:*

- **Position malfunctions** are usually a result a positioning system failure (e.g. GPS). These failures affect the longitude and latitude fields of the safety messages and could manifest as one these four use cases:

Lon_t	$\triangleq$	Longitude at time t
Lat_t	$\triangleq$	Latitude at time t

  1) The position is constant throughout the simulation:
     $Lon_t = Lon_c$
     $Lat_t = Lat_c$
     The constant values are determined at each introduction of a new malfunctioning vehicle.
  2) The position is random at every time-step:
     $Lon_t = U([Lon_{min}, Lon_{max}])$
     $Lat_t = U([Lat_{min}, Lat_{max}])$
     The minimum and maximum values are determined by the size of the simulation playground.
  3) A constant offset is added to the real position:
     $Lon_t = Lon_t + \Delta Lon_c$
     $Lat_t = Lat_t + \Delta Lat_c$
  4) A random offset is added to the real position:
     $Lon_t = Lon_t + U([-Lon_c, Lon_c])$
     $Lat_t = Lat_t + U([-Lat_c, Lat_c])$

- **Speed Malfunctions** could be the result of an OBU error or a physical sensor failure. The speed malfunction is generated similarly to the previously described position malfunction. This results in a *Constant*, *Random*, *Constant Offset* and *Random Offset* modification of the following fields:

Vx_t	$\triangleq$	Speed X component at time t
Vy_t	$\triangleq$	Speed Y component at time t

- **Delayed Messages** could be a result of a large network overhead or a low-cost or slow on-board processing unit. These messages contain all the correct data and required information but are sent with delay Δt from reality.

2) *Attacks:*

- **DoS** attacks consists of a vehicle sending messages with a frequency higher than the limit set by the corresponding IEEE or ETSI standards.
- **DoS Random** are DoS attacks with all the messages fields set to random values. It could be a strategy to flood the network and prevent genuine messages from being broadcasted. This attack could also be executed in *sybil* mode with the attacker changing its identity on every send message to avoid detection.
- **Data Replay** consists of sending information previously received from a specific target neighbor. The replayed information is signed with the attacker's certificate. It could be executed in *Sybil* mode with the attacker changing its identity on every new chosen target to avoid detection.
- **Disruptive** attacks are an information replay of previously received data from random neighbors. It could also be a strategy to flood the network and prevent genuine messages from being broadcasted. This attack could also be executed in *Sybil* and *DoS* modes.
- **Eventual Stop** are attacks where a vehicle simulates a sudden stop by freezing the position values and setting the speed values to null.
- **Traffic congestion Sybil** is an attack aimed at creating a fake traffic congestion. The attacker generates a grid of fake vehicles in a chosen position by maintaining a new identity and a correct message frequency for each fake vehicle.

D. *Generated Datasets*

TABLE I
DATASETS INFORMATION PER DESCRIBED SCENARIO

		Dataset Id *Attack_0709*	 *Attack_1415*	 *MixAll_0024*
Scenario	Time Density	07h-09h 37.03 V/km^2	14h-16h 16.36 V/km^2	00h-24h 23.29 V/km^2
Attacker	Vehicles Messages	1,220 924,251	505 249,612	7,399 7,505,418
Genuine	Vehicles Messages	2,846 2,221,825	1,179 569,723	17,264 11,951,021
Average Size	Plain Gzipped	1.92 GBs 0.40 GBs	0.59 GBs 0.12 GBs	0.91 GBs 0.19 GBs
Total Size	Plain Gzipped	40.51 GBs 8.41 GBs	11.92 GBs 2.42 GBs	10.90 GBs 2.25 GBs

Table. I shows a brief description of the newly generated datasets parameters. The data is encoded in JSON following the same format as the previous VeReMi dataset:

{"type":$\mathbb{Z}_{[0,20]}$,
"rcvTime":$\mathbb{R}_{[0,+\infty]}$,
"sendTime":$\mathbb{R}_{[0,+\infty]}$,
"sender":$\mathbb{Z}_{[0,+\infty]}$,
"senderPseudo":$\mathbb{Z}_{[0,+\infty]}$,
"messageID":$\mathbb{Z}_{[0,+\infty]}$,
"pos":[$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$],
"pos_noise":[$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$],
"spd":[$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$],
"spd_noise":[$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$],
"acl":[$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$],
"acl_noise":[$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$],
"hed":[$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$,$\mathbb{R}_{[-\infty,+\infty]}$],
"hed_noise":[$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$,$\mathbb{R}_{[0,+\infty]}$]}

In the updated VeReMi dataset, two subsets are created for each type of misbehavior described in section IV-C. One subset in rush hour time (7h-9h) and another in low traffic time (14h-16h) (see Fig. 2b). Every subset includes a file for the ground truth and a list of files containing the received message data. Additionally, one test-bench subset is created with a mix of all the previously described misbehavior attacks spanned on the whole simulation day (0h-24h). All the subsets are a result of the simulation of the Luxembourg network described in section IV-A. The misbehavior attacker penetration rate is set at 30% for all the simulations. All the 39 resulting datasets are published and could be found on our Cloud Drive [18]. Additionally, replicating this dataset or creating new scenarios is possible using F^{2}MD which is open source our GitHub [19].

V. Results

In this section we run a simple detection algorithm on the previously described dataset. The detection is based on a set of plausibility and consistency checks on the received message data. These checks are done on the Position, Speed, Heading and Acceleration fields. The tests include: (1) Absolute plausibility, (2) Temporal consistency, (3) Relative consistency of a field with respect to another, (4) Consistency with respect to a Kalman Filter, (5) Overlap of two vehicles, (6) Beacon frequency compliance, (7) Sudden appearance plausibility, (8) Transmission range plausibility. For a more detailed description of the used checks please refer to our previous publications [20] [21]. Additionally, the implementation is open source and available on GitHub [19].

The plausibility checks are calculated then passed through a threshold mechanism. If the threshold is reached for a certain check, the vehicle message is considered misbehaving. The output of the detection mechanism is partitioned into four groups: True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (PN) (see Table II).

TABLE II
Detection output partition

	Genuine	**Misbehaving**
Detected	FP	TP
Not Detected	TN	FN

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$$

$$Recall = \frac{TP}{TP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$F_1 Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

Using this partition, we are able to calculate the following detection metrics: $Accuracy$, $Precision$, $Recall$ and $F_1 score$. The $Accuracy$ is the ratio of all correctly detected over all the considered detections. In our case, it does not constitute a good detection metric since the data is unbalanced, i.e. more genuine nodes exist than attackers. The $Precision$ indicates the classifiers ability to distinguish between misbe-

TABLE III
Testing results

Id	**Results**			
	Accuracy	*Precision*	*Recall*	$F_1 Score$
DoSRandom_1416	0.9994	0.9993	0.9996	0.9995
DoSRandom_0709	0.9994	0.9994	0.9995	0.9994
RandomPos_1416	0.9991	0.9981	0.9989	0.9985
RandomPos_0709	0.999	0.9979	0.9987	0.9983
ConstPos_0709	0.9958	0.9979	0.9878	0.9928
ConstPos_1416	0.9954	0.998	0.9869	0.9924
DoS_1416	0.9876	0.9993	0.9788	0.9889
DoSRandomSybil_0709	0.9898	0.999	0.9784	0.9886
DoSRandomSybil_1416	0.989	0.9991	0.9773	0.9881
DoSDisruptive_1416	0.9862	0.9857	0.9896	0.9876
DoSDisruptive_0709	0.9861	0.9864	0.9887	0.9876
DoS_0709	0.9859	0.9993	0.9752	0.9871
RandomPosOffset_1416	0.9877	0.9979	0.9614	0.9794
RandomPosOffset_0709	0.9865	0.9973	0.9566	0.9765
Disruptive_1416	0.9827	0.9805	0.9622	0.9713
Disruptive_0709	0.9829	0.9777	0.9638	0.9707
RandomSpeed_1416	0.981	0.998	0.9394	0.9678
RandomSpeed_0709	0.9787	0.998	0.9294	0.9625
ConstPosOffset_0709	0.9665	0.9979	0.8879	0.9397
ConstPosOffset_1416	0.9606	0.9979	0.8726	0.9311
DelayedMessages_0709	0.9512	0.9971	0.8362	0.9096
ConstSpeed_1416	0.9441	0.9974	0.8187	0.8992
MixAll_0024	0.9293	0.9912	0.8228	0.8992
DataReplay_0709	0.9393	0.9372	0.8503	0.8916
DelayedMessages_1416	0.9402	0.998	0.8052	0.8913
ConstSpeed_0709	0.939	0.9976	0.7942	0.8843
DataReplay_1416	0.9307	0.9318	0.8333	0.8798
RandomSpeedOffset_1416	0.8928	0.9972	0.65	0.787
RandomSpeedOffset_0709	0.895	0.997	0.6444	0.7828
TrafficSybil_0709	0.8204	0.9973	0.5902	0.7415
TrafficSybil_1416	0.8001	0.9972	0.5842	0.7367
EventualStop_1416	0.8827	0.9952	0.5301	0.6918
DoSDisruptiveSybil_1416	0.7787	0.988	0.5318	0.6914
EventualStop_0709	0.8856	0.9942	0.5163	0.6796
DoSDisruptiveSybil_0709	0.7699	0.9822	0.5014	0.6639
ConstSpeedOffset_0709	0.8263	0.9953	0.4107	0.5814
ConstSpeedOffset_1416	0.8157	0.9957	0.3969	0.5676
DataReplaySybil_1416	0.7948	0.892	0.3705	0.5235
DataReplaySybil_0709	0.8011	0.92	0.3527	0.5099

having and genuine nodes, for example a low precision means the system is yielding a lot of false positives. The $Recall$ mark the classifiers ability to detect a misbehaving node, i.e. a low recall means an attack is difficult to detect. The $F_1 score$ is the harmonic mean between the $Recall$ and $Precision$. In our case, it could be considered as a measure of the overall detection quality.

Table III shows the results of our basic detection algorithm for each subset of our dataset. The first thing we notice is that the detection quality is largely dependent on the type of misbehavior. The $F_1 score$ almost doubles when comparing between the least and the most detected attack. We also notice that this is mainly due to a lower $Recall$ than a lower $Precision$. This means that our detection solution has a tendency to reduce the False Positives at the expense of some missed detections. We also notice that the time of day and consequently the general vehicle density does not greatly affect the detection quality. For each attack, both the peak time scenario (07h - 09h) and the low density scenario (14h - 16h) are within the same detection range. This effect could be specific to our benchmarking detection solution. Future research with more advanced solutions, especially cooperative detection schemes, could be more revealing in this regard. Last thing we notice is that the $Precision$ is generally lower for the attacks that contains replaying other vehicle's data. This means these attacks are successfully tricking the detection system into flagging the target genuine vehicles as misbehaving. Future detection solutions should consider the effects of this new attack vector and find mechanisms that are resistant to this category of disruptive nodes.

VI. Conclusion

In this paper we provide an extension to the VeReMi dataset for misbehavior detection in VANETs. This extension includes a new set of more elaborate attacks, a realistic physical error model and a larger collection of data. This study also includes the detection results of a simple misbehavior detection mechanisms as an initial benchmark. Our dataset will enable other researchers to further improve their detection mechanisms, create new mechanisms for the newly provided attack vectors and compare their results to our benchmark.

Future works, include the development of a misbehavior reports dataset for global misbehavior evaluation. Additionally, we plan on testing misbehavior solutions on the current ongoing field tests in France and Germany.

Acknowledgment

This research has been carried out with public funds within the scope of the French Program *Investissements d'avenir* in the Technological Research Institute SystemX. This work is also in collaboration with the SecForCARs project by the German Federal Ministry of Education and Research.

References

[1] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "VeReMi: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs," *arXiv:1804.06701 [cs]*, Apr. 2018.

[2] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Dec 2018, pp. 564–571.

[3] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, and S. Nandi, "Machine learning based approach to detect position falsification attack in vanets," in *Security and Privacy*, S. Nandi, D. Jinwala, V. Singh, V. Laxmi, M. S. Gaur, and P. Faruki, Eds. Singapore: Springer Singapore, 2019, pp. 166–178.

[4] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, May 2019, pp. 1–6.

[5] S. So, J. Petit, and D. 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*, ser. WiSec '19. New York, NY, USA: ACM, 2019, pp. 84–93. [Online]. Available: http://doi.acm.org/10.1145/3317549.3323406

[6] "ETSI EN 302 637-2 V1.4.1: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," *ETSI WG5 Technical Specification*, pp. 1–45, January 2019.

[7] "Dedicated short range communications (DSRC) message set dictionary™, sae j2735," *SAE International*, 2016.

[8] 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.

[9] A. Varga, "The omnet++ discrete event simulation system," in *In ESM01*, 2001.

[10] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138, December 2012. [Online]. Available: http://elib.dlr.de/80483/

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

[12] 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.

[13] J. A. DiLellio, E. Carolipio, J.-S. W. Chien, and K. Ghassaei, "Global positioning system accuracy enhancement," United States Patent and Trademark Office (USPTO) US7 969 352B2.

[14] K. Kobayashi, K. C. Cheok, and K. Watanabe, "Estimation of absolute vehicle speed using fuzzy logic rule-based kalman filter," in *Proceedings of 1995 American Control Conference - ACC'95*, vol. 5, June 1995, pp. 3086–3090 vol.5.

[15] M. Hölzl, R. Neumeier, and G. Ostermayer, "Analysis of compass sensor accuracy on several mobile devices in an industrial environment," in *Computer Aided Systems Theory - EUROCAST 2013*, R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 381–389.

[16] C. Deelertpaiboon and M. Parnichkun, "Fusion of gps, compass, and camera for localization of an intelligent vehicle," *International Journal of Advanced Robotic Systems*, vol. 5, no. 4, p. 46, 2008.

[17] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on Misbehavior Detection in Cooperative Intelligent Transportation Systems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 779–811, 2019.

[18] Kamel, Joseph, "Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets," 2020. [Online]. Available: https://github.com/josephkamel/VeReMi-Dataset

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

[20] 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.

[21] 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.