A misbehavior detection framework for cooperative intelligent transport systems

MANGLA, C.; RANI, S.; HERENCSÁR, N.

Pharmacology & Therapeutics
Volume 230, February 2022, 107969, Pages 1-11

ISSN: 0163-7258

DOI: https://doi.org/10.1016/j.isatra.2022.08.029

Accepted manuscript

A Misbehavior Detection Framework for Cooperative Intelligent Transport Systems

Cherry Mangla[a], Shalli Rani[a] and Norbert Herencsar[b]

[a]Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura-140401, Punjab., India
[b]Department of Telecommunications, Faculty of Electrical Engineering and Communications, Brno University of Technology, Technicka 3082/12, Brno, 616 00, Czechia

ARTICLE INFO

Keywords:
C-ITS Architecture
Cooperative Intelligent Transportation
Misbehavior Detection
Security

Abstract

With changing times, the need for security increases in all fields, whether we talk about cloud networks or vehicular networks. In every place, it has its importance, but in vehicular networks where the lives of human beings are involved, security becomes the topmost priority. Therefore, this article aims to shed light on Misbehavior Detection Framework (MDF) used in the Cooperative Intelligent Transport Systems community. Here, MDF keeps an eye on malicious entities on the roads. It is done by regularly evaluating two main checks: consistency and local plausibility. These checks are done by Intelligent Transport System Stations. All the messages received through Vehicle-to-Everything are scrutinized through this model. After that, all the messages are evaluated by local detection mechanisms to decide the holistic message's plausibility. This article mainly focuses on the logic behind the proposed Misbehavior Detection Framework providing more security, evaluating various Machine Learning-based models to ensure one best out of all based on quality and computation latency of all models along with the results of various parameters, such as Recall, Precision, F1 Score, Accuracy, Bookmaker Informedness, Markedness, Mathews Correlation Coefficient, Kappa, and achieved the best results.

1. Introduction

With changing times and automation of vehicles, it is becoming necessary to have technology that can safeguard people from road accidents. C-ITS is an emerging field; work is being done on a regular basis to enhance roads' safety. With automation, cyber-security also becomes a component to be taken care of. In C-ITS, it is more necessary to provide security to vehicles as the lives of people are involved in it. It exchanges messages among different units known as ITS-S. Collaborate people (ETSI and IEEE) of the C-ITS community have agreed to use PKI to secure messages in this vehicle network, also known as the vehicular network. All the entities involved in this network (vehicles and RSUs request pseudonym certificates from PKI, also known as digital certificates. After that, certificates are used by ITS-S to sign all the messages transmitted during the process; hence, integrity, authenticity, and non-repudiation are ensured. All the V2X messages, which can contain various information about the vehicle, such as Velocity, Heading, GPS location, warnings of traffic conditions, and more, are digitally signed using these certificates. ITS-S must secure these messages for the safety of other vehicles and people on the sides of roads. In European Union, ETSI publishes the standards for European C-ITS and in the United States, IEEE does. It is paramount to secure V2X messages to maintain the reliability of C-ITS-based security applications. Nevertheless, certificates provided by PKI's can not provide security against all types of threats. Therefore, there arises a need for better solutions to enhance security. In addition to all this, ITS-S keep changing their certificates to secure themselves from tracking from intruders to ensure privacy [1]. Still, it is not appropriate to trust only digital signatures for providing security in such high-risk areas, as they can not ensure the messages are accurate and valid all the time. To exemplify, let us suppose a vehicle is malicious, but it does have a valid certificate, which can send false information on the C-ITS network. In the USA and EU nations, none less than three million people got injured in various traffic accidents in a year as per NHTSA and DGMT reports [2, 3]. Resultant, it is necessary to use the MD system for security or to mitigate the effects of faulty or malicious ITS-Stations. The main purpose of using the MD system is to detect any abnormal behavior of vehicles and to prevent the entities from doing anything out of usual behavior.

The paper is detailed as follows: Table 1, presented here, has outlined the expanded terms of all abbreviations used in the paper. All the related work is given in Section 2. In Section 3, the system model is laid down along with the general architecture of C-ITS, an overview of the misbehavior detection system, and local detection checks. In Section 4, the proposed system model and its misbehavior detection framework (MDF) are given. Section 5 has the detection mechanisms (threshold-based, non-cooperative trust-based, cooperative trust-based, and machine learning-based), which is concluded with the results of the proposed model MDF chosen by us in Section 6 the result and analysis part. Finally, in Section 7 article is concluded.

cherrymangla@gmail.com (Cherry Mangla);
shalli.rani@chitkara.edu.in, shallir79@gmail.com (Shalli Rani);
herencsn@ieee.org (Norbert Herencsar)
ORCID(s): 0000-0002-2476-9063 (Cherry Mangla); 0000-0002-8474-9435 (Shalli Rani); 0000-0002-9504-2275 (Norbert Herencsar)

Table 1
Abbreviations

Abbreviations	Description
C-ITS	Cooperative Intelligent Transport Systems
DGMT	Directorate-General for Mobility and Transport
DoS	Denial of Service
eNodeB	Evolved Node B
ETSI	European Telecommunications Standards Institute
EWMA	Exponentially Weighted Moving Average
IEEE	Institute of Electrical and Electronics Engineers
ITS-S	Intelligent Transport Systems-Stations
K-NN	K-Nearest Neighbors
LSTM	Long Short Term Memory
LuST	Luxembourg SUMO Traffic
MA	Misbehavior Authority
MD	Misbehavior Detection
MDF	Misbehavior Detection Framework
MDR	Misbehavior Reports
ME	Misbehavior Reaction
MI	Misbehavior Investigation
ML	Machine Learning
MLP	Multi-layer Perceptron
MR	Misbehavior Reporting
NCTB	Non-Cooperative Trust Based
NHTSA	National Highway Traffic Safety Administration
OBUs	On-Board Units
PKI	Public Key Infrastructure
SVM	Support Vector Machines
T-VNets	Trust architecture with standard messaging service
V2X	Vehicle-to-Everything
VeReMi	Vehicular Reference Misbehavior Data-set

2. Related Work

For the last two decades, MBD has ruled over the internet field. The survey [4] contains different studies on MBD where the detection mechanism has four families based on data-centric, node-centric, cooperative, and ML. Combining all the mechanisms discussed above can work altogether, but each mechanism must fit under one of the families. First, the data-centric mechanism for plausibility estimation depends on the message content. In the node-centric mechanism to every neighboring ITS-S, a true value must be assigned. For detecting the implausibilities, the cooperative mechanism depends upon the information-sharing part. The ML part trains the models to make anomaly detection easier. The article [5] introduced the Analysis of Vehicle Behavior and Evaluation scheme, which relies upon consistency and plausibility checks. The checks mentioned are further differentiated into two modules, namely positive and negative ratings. For vehicle behavior evaluation, the combination of modules takes place by using EWMA. In the article [6], a similar technique related to vehicle behavior analysis is discussed, but with an addition of a plausible model to keep a check on intersections of a vehicle. The plausible model has some uncertain calculations for an eye on sensor errors. This plausibility module is also used to calculate a trust value in support of the sensor error calculation. Based on the trust and plausibility values, three types of detection occur: benign, erroneous, or unknown. The article [7] initiated with a novel Trust architecture with standard messaging service (T-VNets), which used a large set of complex detectors and assembled the various trust mechanisms such as data-centric, event-based, watchdog, and RSU. The article [8] introduces the VeReMi, a misbehavior detection dataset used by the VEINS simulator and the LuST network. This model consists of the following types of misbehavior: Fixed Position, Fixed Position Offset, Random Position, Random Position Offset, and an Eventual stop. In the article [9], the author trained and tested multiple machine learning models using the dataset VeReMi. The plausibility check will be the input feature vector for these machine learning models. A machine learning solution was made, trained, and tested for the two models, namely, SVM and K-NN. With a slight difference, both algorithms performed on a similar basis. In [10] and [11], authors presented an Alexnet model for Covid 19 and abnormal brain prediction. However, it is applied to the static data. With the same type of solution to the VeReMi dataset in the article [12], the SVM and SVM with Logistic Regression, both the models were tested with the conclusion of which one is better. In [13], authors have proposed about scalable decision tree algorithm. However, accuracy is still a constraint in this work. Also, both the deep learning models such as MLP and LSTM in the study [14] and [15] were tested and concluded LSTM [16] performed better but with more computation time. Authors in [17, 18, 19, 20, 21], and [22] have proposed various deep learning models for industrial applications because of a huge dataset. However, this work needs to be checked with machine learning models. Table 2 describes the considered ML algorithms along with the proposed MDF, which performed better than the traditional algorithms in terms of accuracy and computation time.

3. System Model

In the upcoming Section, a generation introduction to C-ITS architecture is given, along with the misbehavior detection system and attacker model.

3.1. General Architecture of C-ITS

The V2X messages having various details (information: heading, position, speed, and Road warnings) are the base of the C-ITS system. All the V2X messages shared among OBUs and RSUs are digitally signed with PKI-issued certificates to ensure the identities of ITS stations. Every ITS-S receives two types of identities: many short-term (pseudonym identity: disposal certificates) and one long-term. For securing the ITS-S from trace-back, all the pseudonym identities

Table 2
Comparison of various models

Name of Model	Based on	Trained with	Assumption	Result
[9] SVM	Used as baseline ML solution	Checks Feature Set	Not designed for large datasets	Not suitable as required large datasets
[9] LinearSVC	Used as baseline ML solution	Checks Feature Set	Designed for large datasets	Worst results than SVM
[13] XGBoost	Tree-based model	V2X messages with the Checks Feature Set	No time dependency	Some information is lost and Better for treated data
[15] Multi-Layer Perceptron (MLP) first implementation (MLP-T1)	Feed forward back propagation ANN	Checks Feature Set	1 Dense layer with 18 nodes	Not chosen for our dataset
[15] MLP second implementation (MLP-T10)	ANN	Minimum and the Average of the Checks Feature Set	1 Dense layer with 36 nodes	Not chosen for our dataset
[16] LSTM	RNN	Temporal based, Kinematic Feature Set	Single bidirectional LSTM layer with 20 nodes	Results as per assumed by authors
Proposed MDF	Data-centric based, Node-centric based and SVM, MLP & LSTM	Checks Feature Set & Kinematic Feature Set trained using LuST	ML algorithms are better than deterministic algorithms	NCTB gains better results for security of vehicular networks

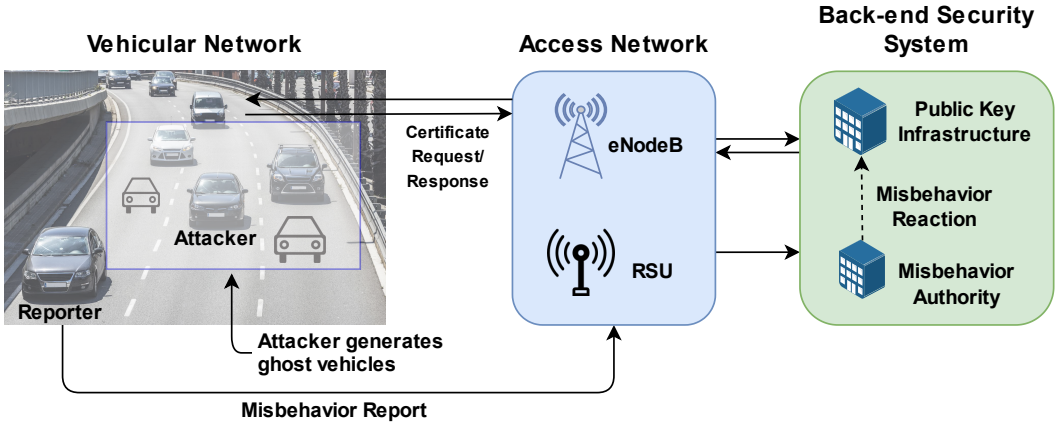

Figure 1: Cooperative Intelligent Transport Systems (C-ITS) Security Architecture having three network areas.

are changed periodically. Figure 1 illustrates the data transmitted for the certificate request among the black vehicle and the PKI via an eNodeB. Although for the authenticity of data (data is not altered during the process) on the receiver side, certificate signatures are necessary even then, data integrity could be compromised. Even if a vehicle (ITS-S) has a valid certificate, it can send inaccurate data to the vehicular network. This could happen in two scenarios: (i) the ITS station is faulty or (ii) a harmful adversary. MA handles these types of misbehavior or semantic attacks. The task of local ITS stations is to do MD checks and send reports Misbehavior Reports (MDR) to Misbehavior Authority (MAs) [23].

3.2. Overview of the Misbehavior Detection System

There are mainly four steps followed by Misbehavior Detection System (Figure 2):

- *Misbehavior Detection (MD)*: Corresponding Vehicles on the road and RSUs in Access Network can detect any abnormal behavior of an entity locally. All the messages received by MA pass through some consistency and plausibility checks [24]. After that, all the checks are sent to the MD application for analysis and to decide whether an MR is required or not.

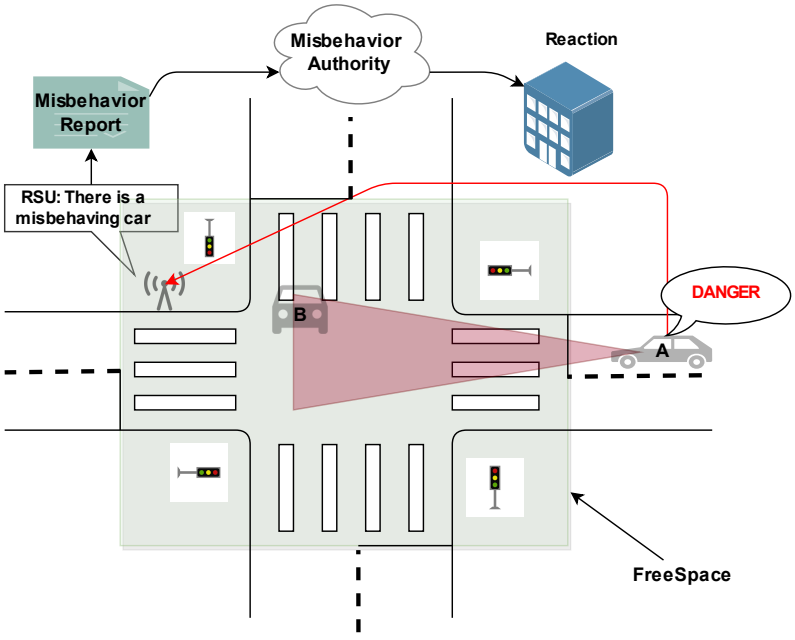

Figure 2: Misbehavior Detection Steps, Various reports generated by vehicles are shown in the figure.

- *Misbehavior Reporting (MR)*: A misbehaving entity is identified by RSU or vehicles soon after detecting implausibility. Then the ITS-S starts looking for any pieces of evidence to prove and recreate to global misbehavior investigation. Then misbehavior report is sent to MA, also known as the central authority by some writers, which is positioned in the cloud [24].
- *Misbehavior Investigation (MI)*: After receiving a report, MA decides whether the misbehaving vehicle is doing it or just a false alarm. After deciding between genuine and wrongly accused reports, MA determines its type. The seriousness of misbehaving ascertains the action that must be taken to protect the system and also alleviates the effect done by misbehavior.
- *Misbehavior Reaction (ME)*: Locally, the only reaction that ITS-S can give is dropping the message that seems malicious. Whereas the global system is more advanced, it only starts once the MA is certain about the detection being done. Firstly, the authorities in charge are informed. For instance, the certificates are revoked by PKI, as shown in Figure 2 [9].

3.3. Attacker Model

In the misbehavior detection system, it is necessary to check how much the attacker influences the traffic system. Initially, a degree is measured to which an attacker has achieved his goal. It is assumed that the attacker is hacking the start position (P) and destination (D) (the total distance between start and destination is d_z).

$$d_z = Speed \times p_z. \tag{1}$$

The time d_z and $d_{z,fair}$ will let us calculate the advantage attacker will have to make the attack. This time (p_z) is the gap time of the network, in which an ITS-S unit that has become an attacker vehicle is behaving like a normal one. This is known as an advantage to the attacker and is calculated as follows:

$$s_z = 1 - \frac{p_z}{p_{z,fair}}. \tag{2}$$

4. Proposed System Model: Misbehavior Detection Framework (MDF)

The proposed model (MDF) came up with a real-time simulation and evaluation of misbehavior detection as a complete solution. This particular architecture has the following levels such as input data, local detection, record data output, and global detection. The complexity, attacks, and detection methods can be chosen based on the level of misbehavior evaluation. Figure 3 illustrates the flowchart of the proposed framework.

4.1. Low rank based proposed MDF

The input dataset to the framework is weight optimization, where data input and output are used to obtain the normalized weight matrix. Then the positive and negative outcome is obtained along with relative closeness and ranking. For local misbehavior detection, a rich module with easy methods for customizing different algorithms must be tested using a simple methodology. This local detection works on basic plausibility and consistency checks on every message the vehicle receives. The transmission of results to the local misbehavior app decides whether to report the misbehavior to the authority or not. As a result, the customization of local detection takes place in two forms: the basic plausibility module, also known as the detectors, and the data fusion

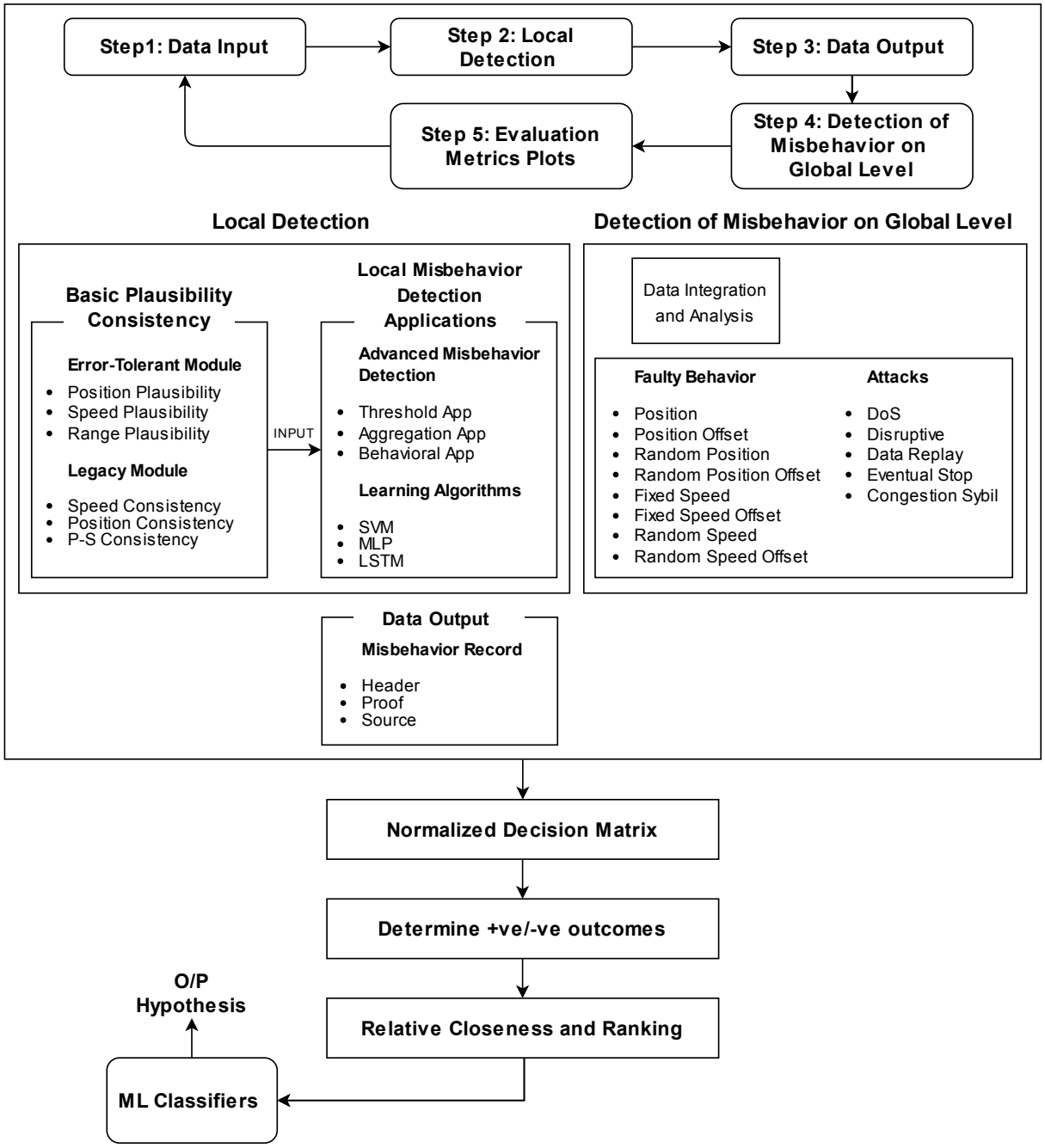

Figure 3: Flow Chart of the Proposed Framework

module. Also, there exist the applications of real-time machine learning-based misbehavior. The weighted normalized decision matrix can be calculated as:

$Y_{a_b} = W_a X x_{a_b};\ b = 1 \cdots p, a = 1 \cdots q$

Let $W_a = [w_1, w_2 \cdots w_q]$ is local criteria weight vector with value $\sum_{a=1}^{n} W_a = 1$.

For positive (E^+) and negative (E^-) ideal solution:

$$(E^+) = Y_1^+ \cdots Y_q^+ = (\max_a Y_{ab}\ b \in B)(\min_a Z_{ab}\ b \in B') \quad (3)$$

$$(E^-) = Y_1^- \cdots Y_q^- = (\min_a Y_{ab}\ b \in B)(\max_a Y_{ab}\ b \in B') \quad (4)$$

where B is benefit criteria and B' is cost criteria. Relative Closeness (RC) to the ideal solution is:

$$RC_a = \frac{N_e^-}{N_a^+ - N_a^-} \quad (5)$$

Ranking as per $RC_a = (a = 1, 2 \cdots n)$ where $RC_a = 1$ indicates highest rank and $RC_a = 0$ indicates the lowest rank.

4.2. Local Detection

4.2.1. Plausibility Checks

Extraction of various sets of basic misbehavior detection. The implementation of checks and detectors in both the versions, such as the legacy and Error-Tolerant versions. The computation of plausibility checks in the legacy version is faster, and binary output is obtained to check the message plausibility. The error-tolerant version has low computation to check plausibility and somehow returns with the uncertainty factor, resulting in the implausibility of the message. Some of the local plausibility checks are as follows:

a) *Range plausibility*: ITS-S position should be checked inside the system's maximum range, which is the predefined maximum value for the ITS system.

b) *Position plausibility*: The position of the sending ITS system is being checked whether it is at a plausible place or not, for example, road rules or overlapping of physical obstacles, etc.

c) *Speed plausibility*: The speed of ITS advertised is checked whether it is less than that of the predefined threshold.

d) *Position consistency*: The plausible separating distance among two consecutive beacons traveling from the same system is being checked.

e) *Speed consistency*: The plausible acceleration and deceleration of two consecutive beacons coming from the same system are being checked.

f) *Position-Speed consistency*: The consistent speed and the distance separating the two consecutive beacons from the system are being checked.

4.2.2. Advanced Misbehavior Detection

The decision-making part of logic detection of the misbehavior detection application. The fusion of multiple factors, such as the plausibility check, node history, etc., is known as fusion applications. Some simple examples are being implemented here. There are already existing machine learning algorithms analyzed in this article with proposed low-rank MDF. At the time of implementation of learning applications in python, the existing algorithms were directly implemented into VEINS, and they can be accessed through specific APIs.

a) *Threshold App*: At the time of failure of a certain message, at least one of the plausibility checks of a node is reported. If the check falls below a certain threshold, a failure is determined.

b) *Aggregation App*: The aggregation app is based on the history of the node. The last n results with the check results of certain messages are aggregated. There is a certain threshold, and in the case of aggregated results falling below the threshold value, a node is reported.

c) *Behavioral App*: The following application is used for the significance of the misbehavior event. Based on the significance of the misbehavior event, there exists a timeout where a node is being put, and the misbehavior authority is being reported when the node sends the data. There is a significant deduction from the plausibility check deduction.

This module also works on various ML algorithms, such as SVM, MLP, and LSTM.

a) *SVM*: This model is used to train genuine vehicles from misbehaving vehicles whose accuracy depends upon the density, network, attacks, etc.

b) *MLP*: Training of MLP is on the same type of data as that of the SVM, but the conclusion being the accuracy of MLP is better than that of the SVM.

c) *LSTM*: The training of this classifier is the same as that for the SVM. LSTM is a part of the Recurrent Neural Network family of ML algorithms to work upon the time-dependent data. Compared to both the SVM and MLP, the classifier LSTM provides the best accurate results. But with a restriction of slow computation power.

4.3. Misbehavior Record

The main aim of our MDF is to send the record to the main misbehavior authority, and then the algorithm will decide whether to generate a misbehavior report or not. The records are then moved to the global misbehavior authority through hypertext transfer protocol. All the records take place in a local folder. This record contains the three headers: namely Header, Source, and Proof. The Header has basic records such as sender and receiver id, record type, generation time, etc. The Source will check the plausibility and consistency of the beacon, and The Proof will help the misbehavior authority in the investigation part and support the conclusion.

4.4. Detection of Misbehavior on Global Level

After ML classifiers, the data will be gathered and analyzed, and the record is sent to the misbehavior authority, then the misbehavior will be detected globally.

- *Integration*: The integration of records added to the database. Using certain criteria, the records get access permission.

- *Analysis*: The misbehavior authority is responsible for the analysis of records. The requirement of the number of records can be modifiable. As a result, evaluation metrics are plotted.

4.4.1. Faulty Behavior

Before the time of transmission, to ensure plausibility, the onboard treatment of each vehicle is a must. In the budget case vehicles, the preventive system lacks behind because this system is prone to failure. Therefore, here we consider

a case where the system's pre-treatment is expected to be failed. Some set of faulty behaviors are being discussed here [25]:

a) *Position*: In each beacon, the vehicle is responsible for broadcasting the same position (P, Q).

b) *Position Offset*: The vehicle itself is responsible for the broadcasting of its real position, but the offset will be fixed (ΔP, ΔQ).

c) *Random Position*: A random position from the playground is being broadcasted by the vehicle.

d) *Random Position Offset*: A real position of a vehicle is broadcasted with a random offset limited to a max value ($\Delta(0\rightarrow P_{max})$, $\Delta(0\rightarrow Q_{max})$.

e) *Fixed Speed*: For each beacon, the same speed is broadcasted by the vehicle (Sp).

f) *Fixed Speed Offset*: The real speed with a fixed offset (ΔSp) is being broadcasted by a vehicle.

g) *Random Speed*: A random speed with an upper limit ($0 \rightarrow S_{max}$) is broadcasted by a vehicle.

h) *Random Speed Offset*: The real speed with a random offset limited to a max value ($\Delta(0\rightarrow S_{max})$) is being broadcasted by a vehicle.

4.4.2. Attacks

The scheme of attacks varies in complexity. Following are the attacks that take place in the following framework.

a) *DoS*: The increase in beaconing frequency by a certain factor took place by the attacking vehicle so that the access of vehicles to the network can be denied by the attacker. In this, the vehicle itself can be able to choose whether to send a valid message or any random one. Moreover, to avoid detection, the attacker chooses to change the data frequently and can also manipulate the already loaded pseudonyms.

b) *Disruptive*: In this type of attack, the attacker is responsible for the flooded network with the old beacons to make the network disruptive. A random beacon is chosen from the history received by the attacker and replays its data. Also, the effect of beaconing frequency can be maximized by the attacker. As a result, the data is generated by the genuine vehicles, but the attacker makes it plausible on some levels, decreasing C-ITS quality. Same as the DoS attack, the attacker can be able to choose the alternatives between the pre-loaded pseudonyms.

c) *Data Replay*: A target is being chosen by an attacker and replays the data with some delay. As a result, an observer observes that there exist two vehicles that are following each other. Furthermore, to avoid detection, the attacker would be able to choose to change the pseudonym at the time of changing the vehicle [20, 21].

d) *Eventual Stop*: An attacker is responsible for the sudden stop as the update of the beacon position stops after a random delay and sets the speed to zero.

e) *Congestion Sybil*: The generation of ghost vehicles gives rise to the Sybil attack. To make this end, the following should be done. The ghost vehicles' speed, position, and heading should be calculated according to the attack or target vehicle. Many pseudonyms are generated and maintained per ghost one pseudonym. The ghost vehicles are multiplexed beacons; for example, the vehicle sends one beacon at each cycle [10, 11, 22].

5. Detection Mechanisms

Based on the proposed MDF, the following are the mechanisms for detecting misbehavior.

– *Threshold Based:* This solution is based on a simple data-centric baseline application. It tests all the results of the checks with the threshold being set. It will give a message of misbehaving if any check fails (as shown in Algorithm 1).

– *Non-Cooperative Trust Based*: This solution is based on a simple trust evaluation node-centric based on the data-centric system. Its main purpose is to look for the behavior of the node based on the trust level a V2X message, and a particular ITS-S has in between them [5, 6]. Trust level is calculated based on the trust combined in the long term and the plausibility calculated currently. The plausibility and trust have a negatively exponential relation (Equation (3)). As it is shown in Algorithm 2.

$$Trust(y) = \frac{e^{(10\times(y-1))}+1}{2\times 10^4}. \tag{6}$$

– *Cooperative Trust Based*: This solution shares information among ITS-Ss. Its main goal is to determine a common level of trust by testing the behavior of a node for a certain ITS-S. The same algorithm used

Algorithm 1 Threshold Based Solution

Require: $d_y : CheckValue, O : Threshold$
  while d_n **do**
    if $d_i < d_{\min}$ **then**
      $d_{\min} = d_i$
    end if
  end while
  if $d_{\min} < O$ **then**
      Misbehaving
  else
      Genuine
  end if

Algorithm 2 Non-Cooperative Trust Based Solution

Require: d_y : $CheckValue, O$: $Threshold$, L_T : $Long-TermTrust$
  while d_n **do**
    if $d_i < d_{\min}$ **then**
      $d_{\min} = d_i$
    end if
  end while
  $L_s = \text{Trust}\ (d_{\min})$
  if $L_s >$ –c and $L_T < 0$ **then**
      $L_T = L_T + 0.1$
  else
      $L_T = L_T + L_s$
  end if
  if $L_T <$ O **then**
      Misbehaving
  else
      Genuine
  end if

for Non-Cooperative Trust-Based is used in it. At the same time, a common level of trust is being shared among all ITS-Ss of the given network.

– *ML based*: Here, many ML algorithms XGBoost [13], MLP [15], LSTM [16], and SVM [26]) are there to train for solution to detect V2X message misbehaving. Following are the details of the parameters and model used for this study. A set of common features is created for every V2X message received. These messages help to find the message's plausibility. Here we took two feature sets for ML algorithms.

  (i) *Checks Feature Set*: Local detection checks performed on V2X messages are Range plausibility, Position plausibility, Speed plausibility, Position consistency, Position-Speed consistency, Position heading consistency, Beacon frequency, Intersection check, Sudden appearance, and Kalman Filter Tracking.

  (ii) *Kinematic Feature Set*: For the Last Beacon: Heading, Position, Acceleration, Time, and Speed. Among the two Beacons: Δ Position, Δ Acceleration, Δ Time, Δ Heading, and Δ Speed.

6. Result, Analysis, and Discussion

6.1. Simulation

We have used VEINS [27] (open source vehicular network simulator) module's framework named as F_2MD [28]. VEINS used for SUMO [29, 30] based on Objective Modular Network Testbed in C++ (OMNeT++) [17, 31].

For estimating correct results out of ML algorithms, LuST is used for training Size of the chosen network is $1.61/km^2$ and 67.4 vehicle/km^2. A total of 17,098,930 messages from 24,773 vehicle units are exchanged, having a 25% attacker rate. This selection of scenarios enables significantly different training and testing datasets. In total, the test bench contains 12,542 vehicles with 8,475,371 exchanged messages with an attacker rate of 5%. The training and testing set ratio is 80%-20% [18, 19].

Alongside, testing is performed on a different area with randomly generated vehicle traces (thus unstable vehicle density). Network size is 1.11 km^2 and density is 17.1 vehicle/km^2. These two datasets are very different from each other. Data test has exchanged 84,75,371 messages from 12,645 vehicles having a 5% attacker rate in total.

For the stable value for s_a, multiple simulations with different values must exist. For each simulation run b and s_a both p_z^b and $p_{z,fair}^b$ can be measured as:

$$s_a = 1 - \frac{1}{M} \sum_{b=0}^{M-1} \frac{p_z^b}{p_{z,fair}^b}, \tag{7}$$

where M is the total number of simulations. For large M there exist an individual traffic situations change with s_a.

6.2. Analysis and Discussion

F1 Score (FS), Markedness (MK), Precision (P), Recall (R) Accuracy, Mathews Correlation Coefficient (MCC), Bookmaker Informedness (BM), and Kappa (κ) are considered as a base for metrics evaluation, Equations (5)–(15) are illustrating the formulas for evaluating these metrics depicting the results of the test datasets. Almost all the detection mechanisms show an accuracy of 98%, as the dataset was unbalanced and had only an attacker rate of 5%. Here, due to an unbalanced dataset, we have considered the F_1 score, Kappa, and MCC for evaluation as shown in Table 3. The results are depicted in Figure 4. In Table 4, the definitions of PR, AB, AC, and PQ are included.

Recall (R): The correctly identified misbehaving messages out of all received misbehaving messages is measured.

$$Recall\ (R) = \frac{AB}{AB + PQ}. \tag{8}$$

Precision (P): The correctly flagged misbehaving messages of all flagged messages.

$$Precision\ (P) = \frac{AB}{AB + PR}. \tag{9}$$

F1 Score (FS): Single metric for evaluating the performance of the system, same as Recall (R) and Precision (P).

$$F_1\ Score\ (FS) = 2 \times \frac{R \times P}{R + P}. \tag{10}$$

Accuracy (ACC): It is defined as the ratio of a positive agreement to the true detection.

$$ACC = \frac{AB + AC}{AB + PR + AC + PQ}. \tag{11}$$

Table 3
Evaluation Metrics

Detection Solution	Recall	Precision	F1 Score	Accuracy	BM	MK	MCC	κ
[5] N-CTB	0.08509	0.9964	0.9264	0.998	0.8601	0.9745	0.9201	0.8301
[9] LinearSVC	0.8506	0.9316	0.8892	0.9928	0.8449	0.9188	0.8711	0.7564
[9] SVM SVC	0.8726	0.9987	0.9312	0.9991	0.8718	0.9888	0.9218	0.8541
[12] CTB	0.9231	0.9699	0.9459	0.9914	0.9201	0.9527	0.9411	0.8664
[13] XGBoost	0.8997	0.8678	0.8832	0.9908	0.8874	0.8589	0.873	0.7262
[15] MLP-T1	0.8611	0.9904	0.9269	0.9984	0.8698	0.9794	0.9228	0.8448
[15] MLP-T10	0.9118	0.9887	0.9387	0.9923	0.9102	0.9808	0.9449	0.8846
[16] LSTM	0.9412	0.9703	0.9555	0.993	0.9381	0.9647	0.9513	0.8954
[28] Threshold	0.8501	0.9362	0.8911	0.9922	0.9327	0.9131	0.883	0.7604
Proposed MDF	0.9554	0.9999	0.9664	0.9999	0.9481	0.9909	0.9613	0.9123

$$\kappa = \frac{ACC - \frac{(AB+PR)\times(AB+AC)+(AC+PR)\times(AC+PQ)}{(AC+AB+PR+FN)^2}}{1 - \frac{(AB+PR)\times(AB+AC)+(AC+PR)\times(AC+PQ)}{(AC+AB+PR+FN)^2}}. \tag{15}$$

Table 4
PR, AB, AC, PQ definitions

	Genuine	Misbehaving
Reported	PR	AB
Not Reported	AC	PQ

Bookmaker Informedness (BM): The proportion of a decision of a system is better than that of a random guess.

$$BM = \frac{AB}{AB+PQ} + \frac{AC}{PR+AC} - 1. \tag{12}$$

Markedness (MK): It is the probability of a certain detection opposed by chance.

$$MK = \frac{AB}{AB+PR} + \frac{AC}{AC+PQ} - 1. \tag{13}$$

Mathews Correlation Coefficient (MCC): The measured classes are of different sizes in case of attackers attacking C-ITS.

$$MCC = \frac{AB\times AC + PR\times PQ}{\sqrt{(AB+PR)(AB+PQ)(AC+PR)(AC+PQ)}}. \tag{14}$$

Kappa (κ): The measure of the positive agreement is similar to the Accuracy, subtraction of the agreement by chance (15).

The evaluation metrics, illustrated in Figure 4, show the Accuracy, F1 Score, Precision, Recall, Bookmaker Informedness, Markedness, Matthews Correlation Coefficient,

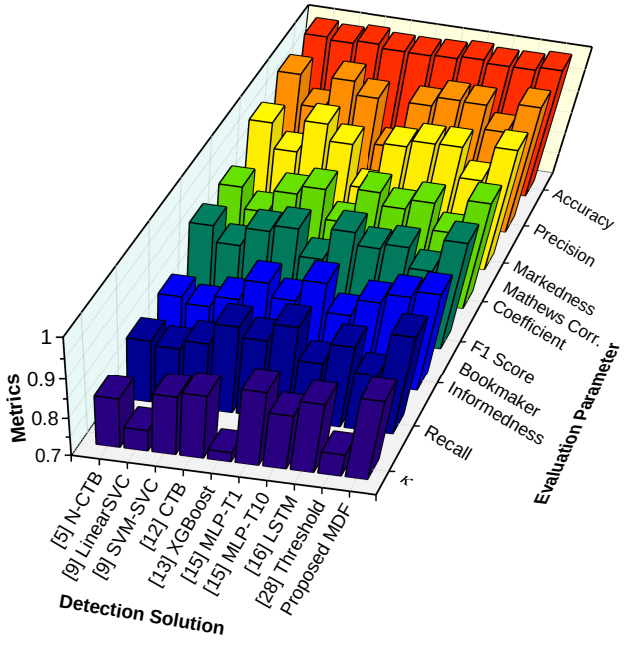

Figure 4: Performance evaluation metrics showing various evaluation parameters and comparative algorithms.

and Cohen's Kappa of the proposed misbehavior detection with that of the detection solutions, and our framework achieved the best results with 0.9554 (Recall), 0.9999 (Precision), 0.9999 (Accuracy) and 0.9664 (F1 Score), 0.9481 (Bookmaker Informedness), 0.9909 (Markedness), 0.9613 (Matthews Correlation Coefficient), and 0.9123 (Cohen's Kappa).

7. Conclusion

Our main focus in this article was to test C-ITS in local misbehavior detection. Various detection solutions were extracted and evaluated. Based on this, we concluded that Machine Learning solutions are better than deterministic algorithms but with not so significant margin value. In our paper, we tried to prove through our analysis that Non-Cooperative Trust-Based solutions are a better option. No doubt, Misbehavior detection is an important aspect of the security of vehicular networks. It is necessary to exchange messages more securely and in a better way. This need to be worked on in the future. A low rank-based weight optimization is used where a normalized decision matrix is obtained, and positive and negative ideal solution is determined, resulting in ranking and relative closeness. This resulted in the best results such as recall 95.54%, precision 99.99%, f1 score 96.64%, Accuracy 99.99%, BM 94.81%, MK 99.09%, MCC 96.13%, and k is 91.23% of the proposed approach as compared to traditional approaches. Along with better global detection and an efficient record system and achieving the best results in various parameters such as Recall, Precision, F1 Score, Accuracy, Bookmaker Informedness, Markedness, Mathews Correlation Coefficient, and Kappa.

References

[1] H. Zhong, J. Ni, J. Cui, J. Zhang, L. Liu, Personalized Location Privacy Protection Based on Vehicle Movement Regularity in Vehicular Networks, IEEE Systems Journal (2021) 1–12. doi: 10.1109/JSYST.2020.3047397.

[2] Directorate-General for Mobility and Transport (DGMT), Annual Accident Report 2018, European Commission (EC) (2018) 1–85.

[3] National Highway Traffic Safety Administration (NHTSA), Summary of Motor Vehicle Crashes, Department of Transportation (DOT) (2018) 1–8.

[4] R. W. van der Heijden, S. Dietzel, T. Leinmüller, F. Kargl, Survey on Misbehavior Detection in Cooperative Intelligent Transportation Systems, IEEE Communications Surveys & Tutorials 21 (2019) 779–811. doi: 10.1109/COMST.2018.2873088.

[5] R. K. Schmidt, T. Leinmüller, E. Schoch, A. Held, G. Schäfer, Vehicle Behavior Analysis to Enhance Security in VANETs, in: Proc. of the 4th IEEE Vehicle-to-Vehicle Communications Workshop (V2VCOM'2008), 2008, pp. 1–8.

[6] N. Bißmeyer, C. Stresing, K. M. Bayarou, Intrusion detection in VANETs through verification of vehicle movement data, in: Proc. of the 2010 IEEE Vehicular Networking Conference (VNC'2010), Jersey City, NJ, USA, 2010, pp. 166–173. doi: 10.1109/VNC.2010.5698232.

[7] C. A. Kerrache, N. Lagraa, C. T. Calafate, J.-C. Cano, P. Manzoni, T-VNets: A novel trust architecture for vehicular networks using the standardized messaging services of ETSI ITS, Computer Communications 93 (2016) 68–83. doi: 10.1016/j.comcom.2016.05.013.

[8] R. W. van der Hei, T. Lukaseder, F. Kargl, Veremi: A dataset for comparable evaluation of misbehavior detection in vanets, Security and Privacy in Communication Networks. SecureComm 2018. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 254 (2018) 318–337. doi: 10.1007/978-3-030-01701-9_18.

[9] S. So, P. Sharma, J. Petit, Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET, in: Proc. of the 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA'2018), Orlando, FL, USA, 2018, pp. 564–571. doi: 10.1109/ICMLA.2018.00091.

[10] S. Lu, S. H. Wang, Y. D. Zhang, Detection of abnormal brain in MRI via improved AlexNet and ELM optimized by chaotic bat algorithm, in: Neural Computing and Applications, 10799–10811, 2021, pp. doi: 10.1007/s00521-020-05082-4.

[11] S. Lu, Z. Zhu, J. M. Gorriz, S.-H. Wang, Y.-D. Zhang, NAGNN: classification of COVID-19 based on neighboring aware representation from deep graph neural network, in: International Journal of Intelligent Systems, 37, 2, 1572–1598, 2022, pp. doi: 10.1002/int.22686.

[12] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, S. Nandi, Machine Learning Based Approach to Detect Position Falsification Attack in VANETs, in: Proc. of the International Conference on Security & Privacy (ISEA-ISAP'2019), volume 939, Springer, Singapore, 2019, pp. 166–178. doi: 10.1007/978-981-13-7561-3_13.

[13] T. Chen, C. Guestrin, XGBoost: A Scalable Tree Boosting System, in: Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16), San Francisco, California, USA, 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[14] P. K. Singh, M. K. Dash, P. Mittal, S. K. Nandi, S. Nandi, Misbehavior Detection in C-ITS Using Deep Learning Approach, in: Proc. of the 2018 International Conference on Intelligent Systems Design and Applications (ISDA'2018), Springer, Vellore, India, 2018, pp. 641–652. doi: 10.1007/978-3-030-16657-1_60.

[15] C. Van Der Malsburg, Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, in: Brain Theory, 1986, pp. 245–248. doi: 10.1007/978-3-642-70911-1_20.

[16] S. Hochreiter, J. Schmidhuber, Long Short-Term Memory, Neural Computation 9 (1997) 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

[17] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. Pham, S. K. Padannayil, K. Simran, A Visualized Botnet Detection System Based Deep Learning for the Internet of Things Networks of Smart Cities, in: IEEE Transactions on Industry Applications, 56, 4, 4436-4456, 2020, pp. doi: 10.1109/TIA.2020.2971952.

[18] R. Vinayakumar, M. Alazab, S. Srinivasan, A. Arunachalam, P. K. Soman, Adversarial Defense: DGA-Based Botnets and DNS Homographs Detection Through Integrated Deep Learning, in: IEEE Transactions on Engineering Management, 1-18, 2021, pp. doi: 10.1109/TEM.2021.3059664.

[19] R. Vinayakumar, M. Alazab, A. Jolfaei, K.P. Soman, P. Poornachandran, Ransomware Triage Using Deep Learning: Twitter as a Case Study, in: 2019 Cybersecurity and Cyberforensics Conference (CCC), 67-73, 2019, pp. doi: 10.1109/CCC.2019.000-7.

[20] Y.-D. Zhang, Z. Dong, S. H. Wang, X. Yu, X. Yao, Q. Zhou, H. Hu, M. Li, C. Jimenez-Mesa, J. Ramirez, F. J. Martinez, J. Manuel Gorriz, Advances in multimodal data fusion in neuroimaging: overview, challenges, and novel orientation, in: Information Fusion, 64, 2, 149–187, 2020, pp. doi: 10.1016/j.inffus.2020.07.006.

[21] Y.-D. Zhang, S. Chandra Satapathy, D. S. Guttery, J. Manuel Gorriz, S.-H. Wang, Improved breast cancer classification through combining graph convolutional network and convolutional neural network, in: Information Processing & Management, 58, 2, 102439, 2021, pp. doi: 10.1016/j.ipm.2020.102439.

[22] S.-H. Wang, Y.-D. Zhang, DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification, in: ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 16, 2s, 1–19, 2020, pp. doi: 10.1145/3341095.

[23] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, P. Urien, Simulation Framework for Misbehavior Detection in Vehicular Networks, IEEE Transactions on Vehicular Technology 69 (2020) 6631–6643. doi: 10.1109/TVT.2020.2984878.

[24] J. Kamel, I. Ben Jemaa, A. Kaiser, P. Urien, Misbehavior Reporting Protocol for C-ITS, in: Proc. of the 2018 IEEE Vehicular Networking Conference (VNC'2018), 2018, pp. 1–4. doi: 10.1109/VNC.2018.8628407.

[25] R. Ansari, J. Petit, V2X Validation Tool. (2018). [Online]. Available: https://www.blackhat.com/us-18/arsenal/schedule/index.html#vx-validation-tool-11980, BlackHat.

[26] B. E. Boser, I. M. Guyon, V. N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, in: Proc. of the Fifth Annual Workshop on Computational Learning Theory (COLT'92), Pittsburgh, Pennsylvania, USA, 1992, pp. 144–152. doi: 10.1145/130385.130401.

[27] C. Sommer, R. German, F. Dressler, Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, IEEE Transactions on Mobile Computing 10 (2011) 3–15. doi: 10.1109/TMC.2010.133.

[28] J. Kamel, Framework For Misbehavior Detection (F2MD). (2018). [Online]. Available: https://github.com/josephkamel/F2MD, accessed on 02-06-2021.

[29] D. Krajzewicz, J. Erdmann, M. Behrisch, L. Bieker, Recent Development and Applications of SUMO - Simulation of Urban MObility, International Journal on Advances in Systems and Measurements 5 (2012) 128–138.

[30] L. Codeca, R. Frank, T. Engel, Luxembourg SUMO Traffic (LuST) Scenario: 24 hours of mobility for vehicular networking research, in: Proc. of the 2015 IEEE Vehicular Networking Conference (VNC'2015), Kyoto, Japan, 2015, pp. 1–8. doi: 10.1109/VNC.2015.7385539.

[31] A. Varga, The OMNET++ Discrete Event Simulation System, in: Proc. of the 15th European Simulation Multiconference (ESM'2001), Prague, Czech Republic, 2001, pp. 1–7.