CaTch: A Confidence Range Tolerant Misbehavior Detection Approach

Joseph Kamel, Arnaud Kaiser, Ines Ben Jemaa, Pierpaolo Cincilla, Pascal Urien

CaTch: A Confidence Range Tolerant Misbehavior Detection Approach

Joseph Kamel[1,2], Arnaud Kaiser[1], Ines Ben Jemaa[1], Pierpaolo Cincilla[1], and Pascal Urien[2]

[1]*IRT SystemX*, 8 avenue de la Vauve, 91120, Palaiseau, France.
Email: {joseph.kamel, arnaud.kaiser, ines.ben-jemaa, pierpaolo.cincilla}@irt-systemx.fr
[2]*Telecom ParisTech* 23 avenue d'Italie, 75013, Paris France. Email: pascal.urien@telecom-paristech.fr

***Abstract*—Misbehavior detection is a challenging problem that needs to be addressed in vehicular communications. Misbehavior detection consists of monitoring the semantic of the exchanged messages to identify potential misbehaving entities. This is achieved by performing plausibility and consistency checks on exchanged beacon and warning messages. However, existing misbehavior detection solutions ignore the mandatory information on data inaccuracy, being gathered by the vehicular sensors. In this paper, we propose CaTch, an embedded misbehavior detection solution that integrates data inaccuracy when performing plausibility and consistency checks. Through extensive simulations, we show that CaTch is able to attribute an accurate uncertainty factor to misbehaving nodes and that it performs better than the state-of-the-art solutions.**

***Index Terms*—C-ITS, misbehavior detection, tolerant checks**

I. Introduction

Vehicular communications have a great potential in improving road safety and enabling autonomous driving. Vehicles exchange messages with critical information (speed, position, steering, etc.) to establish cooperative awareness. At the standardization level, many efforts have been carried out. For instance, the European Telecommunications Standards Institute (ETSI) and the Institute of Electrical and Electronics Engineers (IEEE) both published a set of standards that defines vehicular communication protocols and safety messages format [1].

The exchange of vehicular messages brings to light a number of challenges. The research community is now focusing on open problems such as network performances [2], privacy [3] and misbehavior detection [4]. The latter is the focus of this paper. Misbehavior detection systems in Cooperative Intelligent Transport Systems (C–ITS) monitors the semantic of the exchanged data to identify internal misbehaving nodes. The system then issues the correct reaction in order to prevent the vehicular network from deviating from its expected functionality.

A current misbehavior detection system contains a local component embedded in each vehicle and a global Misbehavior Authority (MA) located in the cloud. The embedded system's role is to check for implausibilities in the received data then send a report to the global entity. The MA would collect and analyze the reports then issue the correct reaction to protect the system.

In this paper we focus on the embedded component. We differentiate the process of checking data plausibility and the local decision making process. In this model the local misbehavior detection application decides whether or not to report the detected node, based on the output of the plausibility checks.

The standard ETSI Cooperative Awareness Message (CAM) [5] and IEEE Basic Safety Message (BSM) [6] messages integrate a field called *confidence range* for each mobility parameter. This field is included based on the fact that sensor measurements could be inaccurate due to physical limitations or environmental characteristics. However, to the best of our knowledge, so far this information is not taken into consideration during the misbehavior detection checks computation.

In this paper, we present CaTch (Confidence range Tolerant misbehavior detection approach), a misbehavior detection library which takes into consideration the *confidence range*. Then we show through extensive simulations, that taking into consideration sensors inaccuracy during the checks computation increases the detection quality of the system.

The remainder of this paper is organized as follows. Section II presents the related work. In section III, we detail our approach in assessing the uncertainty factor while performing the misbehavior basic checks. Section V details the performance evaluation results. Finally, section VI concludes the paper and gives some future insights.

II. Related work

Several works in the literature rely on plausibility and consistency verification of mobility data to detect misbehaving vehicles. Plausibility tests consist of verifying if the received data are consistent with a predefined rule or model. Whereas consistency tests are based on comparing actual received data with previously received ones.

[7] is a recent work which provides data set used to evaluate the performance of several data-centric misbehavior detection mechanisms. These detection mechanisms check the plausibility and consistency of several parameters such as the transmitter radio range, the traveled distance between two transmitted beacons, its potential position conflict with the geographical map and its sudden appearance within the radio range of the vehicle. Based on their results, they recommend to combine these checks to increase the effectiveness of the misbehavior detection.

The author of [8], propose VEBAS, a misbehavior detection framework which combines the already cited verification using a weighting value. The framework analyses then the neighbor behavior locally and establish a specific reputation rate of its neighbors.

[9] proposes a cooperative misbehavior approach which suggests the use of an enhanced Acceptance Range Threshold where the acceptance range is similar to a Gaussian curve instead of a fixed threshold. Moreover, it uses the proactive exchange of neighbor tables which relies on broadcasting the neighbor list in the beacons in addition to the vehicles' positions. The vehicle then use subjective logic to check the consistency of the neighbor's with the position reported by other neighbors.

In this work, we were mainly inspired by local data-centric checks that we explored in our previous work [4]. These local plausibility checks are also considered as necessary checks on all incoming messages by the National Highway Traffic Safety Administration (NHTSA) and thus been included in the Federal Motor Vehicle Safety Standards for V2V Communications [10]. In addition to the misbehavior detection checks, we propose the use of additional relevant verification based on the mobility information received from the beacons. However, the above mentioned misbehavior checks ignore the mandatory included sensor data errors, focusing only on the absolute value of the transmitted information. For instance, according to the experiments of [11], GPS positioning error is estimated to 5 meters when used in communications between vehicles and road users. Integrating the positioning error information in the beacon messages is required to increase the accuracy of the ITS safety applications. In this work, we tackle misbehavior due to the transmission of false mobility information in the beacons. Existing misbehavior detection mechanisms would not distinguish between intentional and non-intentional transmission of erroneous mobility information. We believe that utilizing the mandatory transmitted uncertainty information in the plausibility calculation would have a significant impact on the output results of existing misbehavior detection mechanisms.

III. Impact of mobility data uncertainty on misbehavior detection

A. The local misbehavior detectors

In this work, we consider plausibility and consistency verification of the content of the received beacon message. Note that a *plausibility* check requires only one received message whereas a *consistency* check requires two successive messages coming from the same source.

For instance, when performing a plausibility check on the radio transmission range, a vehicle would judge that a neighbor which sends a position that is not in its communication range is suspicious. A typical example of consistency checks is when a vehicle receives two beacons from its neighbor at different times indicating speed information which are conflicting with the travelled distance information.

A complete set of plausibility detectors for CAM is detailed in [12]. However in this paper we only consider the following checks:

- **Range plausibility:** Check if the position of the sending Intelligent Transport Systems (ITS) Station (ITS–S) is outside of the ego ITS–S maximum range (predefined value mapped on the ego ITS–S maximum radio coverage).
- **Position plausibility:** Check if the position of the sending ITS–S is in a coherent place (e.g. it is on a road, it does not overlaps physical obstacles, etc.).
- **Speed plausibility:** Check if the speed advertised by the sending ITS–S is less than a predefined threshold.
- **Position consistency:** Check if two consecutive Beacons coming from a same ITS–S have plausible separating distance.
- **Speed consistency:** Check if two consecutive Beacons coming from a same ITS–S have plausible acceleration or deceleration.
- **Position speed consistency:** Check if two consecutive Beacons coming from a same ITS–S have consistent speed and separating distance.
- **Position heading consistency:** Check if the positions in two consecutives Beacons coming from a same ITS–S corresponds to the travel heading advertised by that ITS–S.
- **Intersection check:** Check if two Beacons coming from two different ITS–S have overlapping locations (i.e. both ITS–S overlap each others).
- **Sudden appearance:** Check if an ITS–S suddenly appeared within a certain range. Note that we do not consider such ITS–S as misbehaving when this check is the only one that failed.

B. Misbehavior Detection Uncertainty Assessment

In this work, we propose a novel solution, *CaTch*, which takes into consideration the mobility information uncertainty while performing the basic misbehavior checks. In the following sections, we detail how we integrate the confidence range of the beacon message contents in the basic checks calculation.

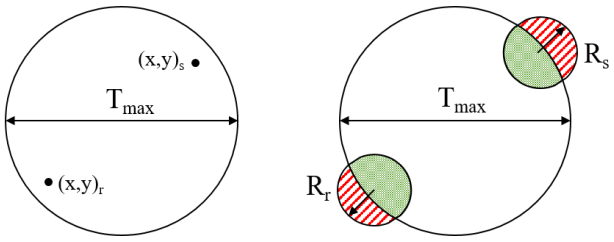
For each plausibility and consistency test, we compare our calculation (CaTch version) with the state-of-the-art one (legacy version). We calculate the uncertainty factor f for each of the misbehavior checks. The uncertainty factor f is a real number between 0 and 1, with 0 being certainly malicious and 1 having no signs of misbehavior. We assign a value of f to the nodes that are partially implausible.

Notice that the confidence range is illustrated by green for the plausible section and red for the implausible one. We use the following common notations as depicted in Table I.

1) Range Plausibility Check: We assume that the communication range is based on a disk model. The initial method to detect an out of transmission range node is done by simply checking if the distance between the sender and the receiver is less than the maximum plausible range as illustrated in figure 1a. In our approach, we take into consideration the

TABLE I: Common Notations

R_x	$\triangleq$	Position confidence range in beacon x
V_x	$\triangleq$	Claimed speed in beacon x
C_x	$\triangleq$	Speed confidence range in beacon x
D_x	$\triangleq$	Claimed heading in beacon x
Δt_{ij}	$\triangleq$	Time separating beacons i and j
d_{ij}	$\triangleq$	Distance separating beacons i and j
A_x	$=$	πR_x^2



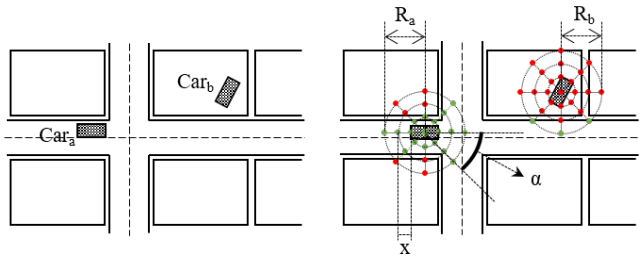
(a) Legacy Version (b) CaTCh Version

Fig. 1: Range Plausibility Check

confidence range of the sender's position broadcasted in the beacon R_s and the confidence range assessed internally by the receiver R_r. We calculate first the position confidence range area of the sender and the receiver, respectively A_s and A_r. We consider T_{max} as the maximum diameter in which the sender and the receiver could communicate. T_{max} coincides with the transmission range of the sender and the receiver. Accordingly, $A_{T_{max}}$, is the area of the circle formed by T_{max}. Notice that any node which is within $A_{T_{max}}$ could communicate with both the sender and the receiver. Second, we calculate the intersection area a_s and a_r for the sender and the receiver respectively with $A_{T_{max}}$ (the green shaded zone as illustrated in figure 1b). Finally, we compute the uncertainty factor of this check as follows:

T_{max}	$\triangleq$	Communication Diameter
$A_{T_{max}}$	$=$	$\frac{\pi T_{max}^2}{4}$
a_r	$=$	$A_{T_{max}} \cap A_r$
a_s	$=$	$A_{T_{max}} \cap A_s$

$$f = (a_r + a_s)/(A_r + A_s) \quad (1)$$



(a) Legacy Version (b) CaTCh Version

Fig. 2: Position Plausibility Check

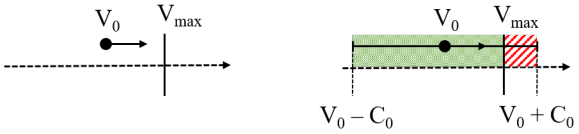
2) Position Plausibility Check: The position plausibility is determined using a straightforward check between the claimed position and the geographic map of the environment. For instance, in figure 2a, Car_b broadcasts a position that is overlapping with a building. This makes its position definitely non-plausible. In CaTch , the position plausibility is calculated for multiple points that are within the position confidence ranges, R_a and R_b of Car_a and Car_b respectively, as depicted in figure 2b. Every segment is separated by an angle α and every point is separated by a distance x. These points are plotted on multiple segments along the radius of the confidence range. The green points in the figure are the possible plausible positions for both Car_a and Car_b and the red points are the non plausible ones. Let n be the number of green points in the figure and let N be the total number of the considered points.

The uncertainty factor of this check is computed as follows:

n	$\triangleq$	Plausible Tested Points
N	$\triangleq$	All Tested Points

$$f = n/N \quad (2)$$

Note that if a vehicle is positioned outside of a road and its advertised velocity is zero, the check does not fail (i,e., we consider the car as parked). However, if the advertised velocity is different from zero, the test fails.



(a) Legacy Version (b) CaTCh Version

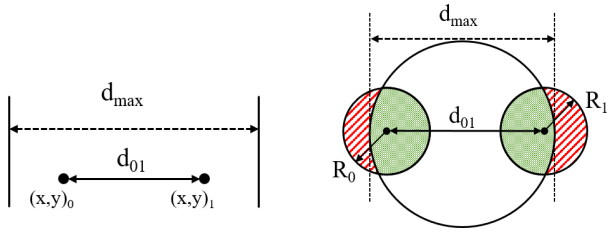
Fig. 3: Speed Plausibility Check

3) Speed Plausibility Check: The speed plausibility is obtained by comparing the claimed speed V_0 with the maximum predefined speed V_{max} as shown in figure 3a. To integrate the speed confidence range C_0 into the calculation, the CaTch version checks for the plausible partition of the claimed speed confidence range compared to the maximum speed (fig 3b). The uncertainty factor f of this check is computed as follows:

V_{max}	$\triangleq$	Max Plausible Speed

$$f = \frac{(V_{max} - V_0 + C_0)}{2C_0} \quad (3)$$

4) Position Consistency Check: The position consistency is computed by comparing the distance between two consecutive broadcasted positions d_{01} with the maximum plausible Euclidean distance d_{max} calculated taking into account the time of the reception of the 1^{st} beacon and the time of the reception of the 2^{nd} beacon on the road as illustrated in figure 4a. The maximum plausible distance is computed based on the position, speed and heading information received in the $beacon_0$ message. CaTch integrates the position confidence range of two successive received beacons, respectively R_0 and R_1 by calculating the position consistency similarly to the



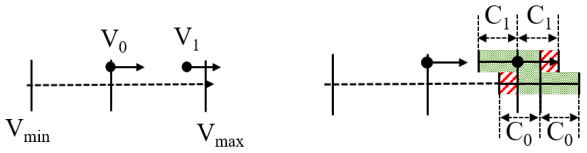
(a) Legacy Version (b) CaTch Version

Fig. 4: Position Consistency Check

range plausibility check. CaTch uses the intersection between the area $A_{d_{max}}$ whose diameter is the maximum possible distance d_{max} (figure 4b) and the areas A_0 and A_1 which are respectively the areas of the of the position confidence of the 1^{st} received beacon and the 2^{st} received beacon. The uncertainty factor f of this check is computed as follows:

d_{max}	$\triangleq$	Maximum plausible distance
$A_{d_{max}}$	$\triangleq$	Area of the maximum plausible distance
a_n	$=$	$d_{max} \cap R_n$

$$f = (a_0 + a_1)/(A_0 + A_1) \tag{4}$$



(a) Legacy Version (b) CaTch Version

Fig. 5: Speed Consistency Check

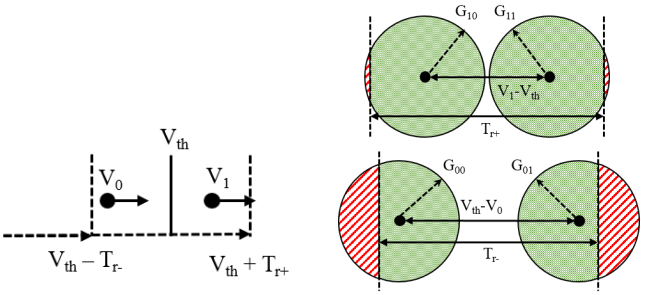
5) Speed Consistency Check: The speed consistency is obtained by checking if the speeds V_0 and V_1 from two consecutive beacons are consistent with the maximum possible acceleration or deceleration, as illustrated in figure 5a. Instead of a direct comparison, CaTch takes into account the confidence range C_0 and C_1 of the speeds of the consecutive beacons as illustrated in figure 5b. The uncertainty factor f is calculated as the overlap of the maximum speed V_{max} and the minimum speed V_{min} with the broadcasted speeds ranges. The uncertainty factor if thus calculated as follows:

V_{min}	$\triangleq$	Minimum plausible speed when the vehicle decelerates
V_{max}	$\triangleq$	Maximum plausible speed when the vehicle accelerates

$$f_{max} = \frac{V_{max}-V_1+C_0}{4C_0} + \frac{V_{max}-V_1+C_1}{4C_1}$$

$$f_{min} = \frac{V_1-V_{min}+C_0}{4C_0} + \frac{V_1-V_{min}+C_1}{4C_1}$$

$$f = \left\{ \begin{array}{ll} f_{min}, & \text{when } V_1 \leq V_0 \\ f_{max}, & \text{when } V_1 > V_0 \end{array} \right\} \tag{5}$$



(a) Legacy Version (b) CaTch Version

Fig. 6: Position Speed Consistency Check

6) Position Speed Consistency Check: The time and distance separating two beacons result in a theoretical speed V_{th} (computed by considering the Euclidean distance). The claimed speed is consistent with the distance separating two beacons if it falls within a range denoted here as T_{r+} and T_{r-} around this theoretical speed as depicted in figure 6a. This check becomes tricky if we consider the confidence ranges on both the position and the speed. To this end, we calculate a new theoretical range G_x formed with a combination of the speed and the position confidence. Next we find the maximum and minimum between the speed of the first and the second beacon. We then check the plausibility of the claimed speed and the tolerance range with respect of this theoretical confidence range (fig 6b). The calculation of the uncertainty factor is as follows:

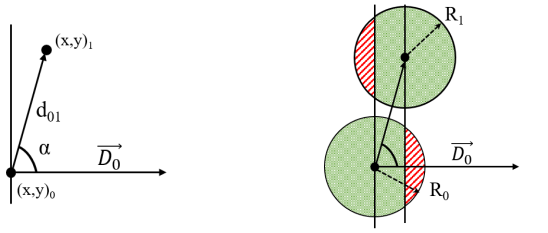
V_{th}	$\triangleq$	Theoretical speed based on separating distance
V_{min}	$\triangleq$	Minimum advertised speed between 1^{st} & 2^{nd} beacon
V_{max}	$\triangleq$	Maximum advertised speed between 1^{st} & 2^{nd} beacon
T_{r+}	$\triangleq$	Tolerance range on excess speed
T_{r-}	$\triangleq$	Tolerance range on dearth speed
Δt	$\triangleq$	Time separating first & second beacon
G_{min0}	$=$	$C_{min} + R_0/\Delta t$
G_{min1}	$=$	$C_{min} + R_1/\Delta t$
G_{max0}	$=$	$C_{max} + R_0/\Delta t$
G_{max1}	$=$	$C_{max} + R_1/\Delta t$
lb_{min}	$=$	$V_{th}/2 - V_{min}/2 - T_{t-}$
lb_{max}	$=$	$-V_{th}/2 + V_{max}/2 - T_{t+}$

$$f_{min} = \frac{2\int_{lb_{min}}^{G_{min0}} \sqrt{G_{min0}^2 - x^2}dx + 2\int_{-G_{min1}}^{-lb_{min}} \sqrt{G_{min1}^2 - x^2}dx}{A_0 + A_1}$$

$$f_{max} = \frac{2\int_{lb_{max}}^{G_{max1}} \sqrt{G_{max1}^2 - x^2}dx + 2\int_{-G_{max0}}^{-lb_{max}} \sqrt{G_{max0}^2 - x^2}dx}{A_0 + A_1}$$

$$f = \left\{ \begin{array}{ll} f_{min}, & \text{when } f_{min} > f_{max} \\ f_{max}, & \text{when } f_{min} \leq f_{max} \end{array} \right\} \tag{6}$$

7) Position Heading Consistency Check: To check the consistency of the heading, we calculate the angle between the advertised heading vector $\vec{D_0}$ and the real heading in the next position. This angle should be less than a predefined threshold. We set this threshold to $\pi/2$ assuming that this is a universal limit, non-dependent on vehicle specific characteristics (figure 7a). R_0 and R_1 are respectively the position confidence range of the 1^{st} and 2^{nd} beacon. The confidence range however could heavily affect this angle. CaTch calculates the plausible



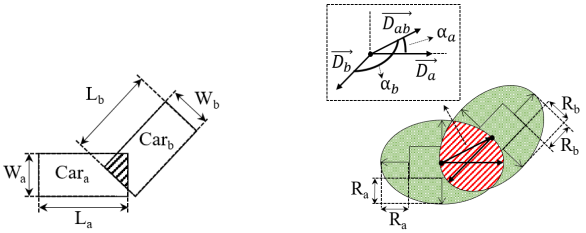
(a) Legacy Version (b) CaTch Version

Fig. 7: Position Heading Consistency Check

portions of the confidence range on both the 1^{st} and 2^{nd} beacon (the shaded green area in figure 7b). A position of a beacon is considered plausible if it forms an angle less than $\pi/2$ with the center of the other beacon. Thus, the calculation of the uncertainty factor goes as follows:

$\vec{D_{01}}$	$\triangleq$	Vector formed by the centers of the 1^{st} & 2^{nd} beacons
α	$\triangleq$	Angle between $\vec{D_0}$ & $\vec{D_{01}}$

$$f = \frac{2\int_{d_{01}\cos\alpha}^{R_0}\sqrt{R_0^2 - x^2}dx + 2\int_{-R_1}^{-d_{01}\cos\alpha}\sqrt{R_1^2 - x^2}dx}{A_0 + A_1} \tag{7}$$



(a) Legacy Version (b) CaTch Version

Fig. 8: Intersection Check

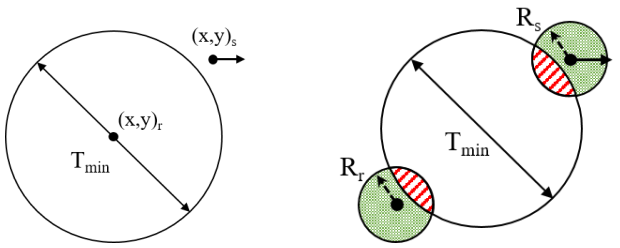
8) Intersection Check: The usual intersection check, models each vehicle as a rectangle. It uses the broadcasted length and width of the vehicle in the beacons as illustrated in figure 8a. CaTch models each vehicle as an ellipse with an increased width and length according to the confidence range of the position (fig 8b). To calculate the uncertainty factor, first the intersection area between the two ellipses is calculated (fig 8b). However this factor alone is not enough since the severity of the intersection depends greatly on the location it occurred and the rotation of the affected vehicles. For example, an intersection on the peripherals is less important than an intersection on the center even for the same intersection area factor. This phenomenon is numerically added to the equation using an importance factor as calculated below:

$$f_a = Ae_{ab}/(Ae_a + Ae_b - Ae_{ab})$$

$$f_i = ol_{ab}/(I_a + I_b - ol_{ab})$$

$$f = f_a f_i \tag{8}$$

L_n	$\triangleq$	Car_n length
W_n	$\triangleq$	Car_n width
Ae_n	$=$	$\pi(R_n + L_n)(R_n + W_n)$
E_n	$\triangleq$	$\frac{x^2}{(R_n+L_n)^2} + \frac{y^2}{(R_n+W_n)^2} = 0$
Ae_{ab}	$=$	$E_a \cap E_b$
d_{ab}	$\triangleq$	Distance between the centers of Car_a & Car_b
$\vec{D_{ab}}$	$\triangleq$	Vector formed by the centers of Car_a & Car_b
α_a	$\triangleq$	Angle between $\vec{D_{ab}}$ & $\vec{D_a}$
α_b	$\triangleq$	Angle between $\vec{D_{ab}}$ & $\vec{D_b}$
I_a	$=$	$(R_a + L_a)\sin\alpha_a + (R_a + W_a)\cos\alpha_a$
I_b	$=$	$(R_b + L_b)\sin\alpha_b + (R_b + W_b)\cos\alpha_b$
ol_{ab}	$=$	$max(0, min(I_a/2, I_{ab} + I_b/2) - max(I_a, I_{ab} - I_b/2))$



(a) Legacy Version (b) CaTch Version

Fig. 9: Sudden Appearance Check

9) Sudden Appearance Check: The sudden appearance flag is triggered when a vehicle appears inside a minimum range T_{min}. Accordingly, $A_{T_{min}}$, is the area of the circle formed by T_{min}. The idea is that a new vehicle detected in this range T_{min} is not entering the edge of our reception range and is thus suddenly appearing. In order to integrate the error range, CaTch uses the inverse of the method used for the range plausibility (see III-B1). The system finds the impossible positions by calculating the intersection of $A_{T_{min}}$ with the confidence range of each vehicle. Therefore, the plausibility factor is calculated as follows:

T_{min}	$\triangleq$	Minimum acceptable range for sudden appearance
$A_{T_{min}}$	$=$	$\frac{\pi T_{min}^2}{4}$
a_r	$=$	$A_{T_{min}} \cap R_r$
a_s	$=$	$A_{T_{min}} \cap R_s$

$$f = (R_r + R_s - a_r - a_s)/(R_r + R_s) \tag{9}$$

IV. Misbehavior Detection Applications

CaTch generates results that are contentious instead of binary, thus contains more information. Providing more information should translate into at least similar or better detection results, depending on the detection application.

1) Detection Applications: In order to evaluate the impact of CaTch results, we run two local misbehavior detection applications:

- **Simple App** This application consists of a simple threshold. Using this application, all messages with uncertainty factor f less than 0.5 for any detector are reported. This application does not take advantage of the additional information provided by CaTch and theoretically both versions of the detectors should perform similarly.

- **Advanced App** This application is based on Machine Learning (ML). Using data from the simulation output, we trained a neural network based on Multilayer Perceptrons (MLPs). The output values of the detectors are placed in an array and used as the MLPs features. The training is done for both versions of the detectors (i.e., with and without CaTch), then tested using new simulation data. Theoretically this application should perform better as it uses the additional information provided by CaTch.

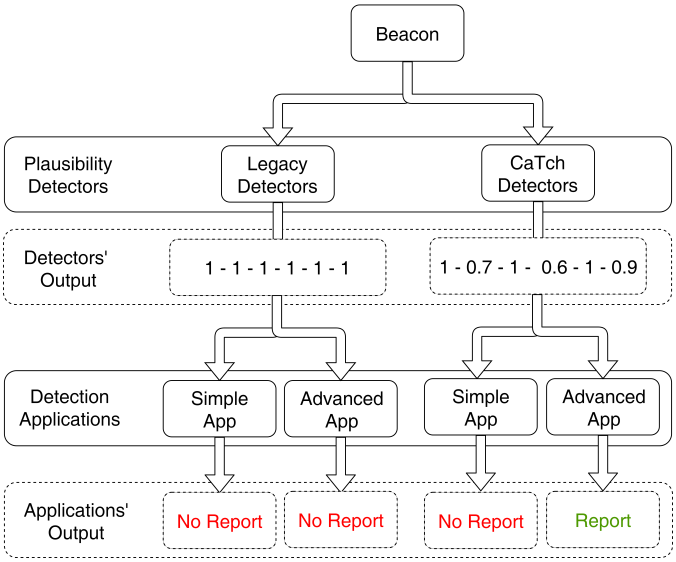


Fig. 10: Example illustrating an advantage of using CaTch

To illustrate how CaTch impacts misbehavior detection, we present the simple example depicted in figure 10.

A beacon message is received. Both detectors check the beacon data (in this example, an arbitrary value of 6 checks is used). The legacy detector outputs binary values only. Here, all 6 checks output the value '1'. CaTch detector outputs real values between 0 and 1 (uncertainty factor f). For checks 1, 3 and 5, CaTch outputs the value '1' like the legacy detector. However for the remaining checks, CaTch does not attribute categorical values (i,e., 0 or 1) as the legacy one and rather, provides floating values.

The detection applications then use these results to decide whether or not a misbehavior report has to be sent. In this example, as the legacy detector generate only '1' values for each check, none of the detection applications trigger the emission of a misbehavior report (remember that the value '1' means the check passed whereas the value '0' means the check failed). However, using the CaTch results, the advanced application triggers a misbehavior report whereas the simple application does not as no CaTch checks are below the threshold value of 0.5.

V. Evaluation

In this section, we show the impact of the measurements uncertainty on the detection results. We evaluate the performance of the legacy basic checks and we compare them with CaTch using multiple detection applications.

A. Experimental Setup

The detectors in the legacy and CaTch versions are implemented as a VEINS extension. VEINS [13] is an open source framework for vehicular network simulations. VEINS is based on OMNeT++ and SUMO, a network simulator and road traffic simulator respectively. As the network, we used a $2.68km^2$ area of Paris-Saclay. A network that combines a suburban-like grid and some organic network properties. The vehicles input flow has been carefully controlled to have $21vehicles$ at any moment which would result in an approximate density of $7.83Vehicle/km^2$. The vehicles send and receive beacon messages with all previously treated information. Every time a vehicle receives a beacon message, all the plausibility detectors are computed and the results are passed to the detection application. For further technical details, the source code of the our VEINS extension along with all the configuration details of the simulated scenario are published on github [14].

1) Considered Attacks: To induce misbehavior, we tested two types of behaviors:

- **The Constant Offset**: in which the misbehaving node emits its sensor location with a fixed offset of $50m$ on the X and Y axis.
- **The Sybil Attack** in which a attacker generates and transmits a virtual grid of vehicles using the plausible data of an existing vehicle in the network. The attacker generates an identity and manages a correct transmission frequency for each ghost vehicle.

The *Constant Offset* is a type of misbehavior often used in this field [7] [9], it is more a simulation of a faulty device than an attack. As for the *Sybil Attack* it is a more complicated type of misbehavior that could cause more disruption to the system. The attacker density was set to 0.1 . The attacks implementation is open source and could be found on github [14].

Furthermore, given the nature of the simulator we did not have measurements errors out of the box. In order to simulate measurements uncertainty, we shifted the simulated location by a random value. The the Global Positioning System (GPS) error and is extracted from normal distribution probability density function with a mean $\mu = 0$ and variance $\sigma = 1$, which is a plausible assumption for location errors [15].

All the vehicles introduced in the simulation are running all of the detectors in both the legacy and the CaTch version. Every time a vehicle received a message it is checked for implausibilities. The results of the plausibility checks are then analyzed by a local misbehavior detection application to determine whether or not to report the subject node.

2) Evaluation metrics: The detection application decide whether or not to report a received message. The output of the detection is arranged according to the classic partition described in Table II.

Using this partition we can calculate, similarly to other studies [7] [16], the following performance metrics:

TABLE II: TP, TN, FP and FN definition

	Genuine	Misbehaving
Reported	FP	TP
Not Reported	TN	FN

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

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

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

$$F_1Score = 2 \times \frac{Recall \times Precision}{Recall + Precision}$$

$$BM = \frac{TP}{TP+FN} + \frac{TN}{TN+FP} - 1$$

$$MCC = \frac{TP+FP}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

Intuitively, the $Recall$ characterizes the system ability to flag all the misbehaving messages, Whereas the $Precision$ characterizes the system ability to not consider misbehaving as genuine messages. The F_1score is the harmonic mean of $Recall$ and $Precision$. It could be used as a single measure to evaluate the system's performance, given that the same importance is applied to both factors, the $Recall$ and the $Precision$. However, more often than not, both factors are not given the same weight, especially with detection mechanisms where $Recall$ is weighed more than $Precision$. The $Accuracy$ is the ratio of the true classified messages over all the classified messages. Unlike the F_1score, the $Accuracy$ takes into account the FNs. The $Bookmaker\ Informedness$ (BM) characterizes the probability of an informed decision. Its score represents how much the system is better than a random guess. The $Matthews\ Correlation\ Coefficient\ (MCC)$ is also a measure of the quality of a binary classification system, however it is especially useful when the measured classes are of very different sizes which is true in our case. This makes it arguably the most suitable metric for our evaluation.

B. Results

Table III shows the results of the previously described simulated scenarios. The scenarios consist of detecting both the Position Offset faulty behavior and the Sybil attack using the simple threshold app and the more advanced Machine Learning app. We run both versions of the detectors simultaneously within each scenario to avoid distorting the results due to randomness. We kept the simulations running until the point of heuristic equilibrium, where the cumulative evaluation metrics stabilized.

The first observation we make is that the detection results depend greatly on the type of misbehavior. The detection results of the *Constant Offset* scenario are better than that of the *Sybil Attack*. Therefore in our scenarios, *Sybil Attack* is much harder to detect than the *Constant Offset*. This is an expected result since that the former have much more plausible features than the latter.

TABLE III: Simulation Results

Scenario App	Detectors	Evaluation Metrics		
		Recall	**Precision**	**Accuracy**
Threshold	**Legacy**	0.7621	0.9233	0.9691
	CaTch	0.7625	0.9207	0.9689
		Δ0.1%	Δ-0.3%	Δ0.0%
		F_1Score	**BM**	**MCC**
	Legacy	0.8350	0.7548	0.8227
	CaTch	0.8342	0.7550	0.8216
		Δ-0.1%	Δ0.0%	Δ-0.1%
Machine Learning		**Recall**	**Precision**	**Accuracy**
	Legacy	0.7642	0.9375	0.9706
	CaTch	0.7498	0.9721	0.9721
		Δ-1.9%	Δ3.7%	Δ0.2%
		F_1Score	**BM**	**MCC**
	Legacy	0.8420	0.7584	0.8312
	CaTch	0.8466	0.7473	0.8400
		Δ0.5%	Δ-1.5%	Δ1.1%

(a) Constant Offset Scenario

Scenario App	Detectors	Evaluation Metrics		
		Recall	**Precision**	**Accuracy**
Threshold	**Legacy**	0.3976	0.9504	0.7468
	CaTch	0.4203	0.9457	0.7546
		Δ5.7%	Δ-0.5%	Δ1.1%
		F_1Score	**BM**	**MCC**
	Legacy	0.5607	0.3834	0.5013
	CaTch	0.5819	0.4038	0.5155
		Δ3.8%	Δ5.3%	Δ2.8%
Machine Learning		**Recall**	**Precision**	**Accuracy**
	Legacy	0.3928	0.9498	0.7446
	CaTch	0.7961	0.9102	0.8852
		Δ102.7%	Δ-4.2%	Δ19.8%
		F_1Score	**BM**	**MCC**
	Legacy	0.5556	0.3783	0.4967
	CaTch	0.8494	0.7424	0.7618
		Δ52.9%	Δ96.2%	Δ53.4%

(b) Sybil Attack Scenario

This leads us to our next point: because the claimed positions are so far from being plausible, the ML app did not find any use for CaTch's uncertainty factor for the detection of faulty nodes. Instead the ML model used CaTch here to better characterize a genuine node. Consequently, the amount of False Positives decreases which lead to a higher Precision (+3.7%). However, this comes at the cost of some of the True Positives and consequently a lower Recall (-1.9%). We notice that all the other evaluation metrics fluctuate accordingly. This constitutes a trade-off between Recall and Precision and we don't find using CaTch in this scenario definitely beneficial. For the case of faulty nodes detection, the Legacy detectors could be sufficient. However, this is not the case for the *Sybil Attack* scenario. Since the messages are more plausible, and the implausibilities are more elusive, the ML app finds a major advantage by using CaTch's additional information. The extra

edge given by CaTch's uncertainty factor enables the ML app to double the Recall (+102.7%) with a small loss in terms of precision (-4.2%). Accordingly, The F_1Score, the BM and the MCC all increase by more than 50%. Therefore, we find that using CaTch in this scenario definitely increases the quality of the detection. Given these points we conclude that the main benefit of CaTch is to detect attacks that are more subtle where the attacker is intelligent and tries to remain within the plausible range. In fact, using CaTch we can train a system to extract a kind of a "fingerprint" of an attack, which is not possible to elaborate when using the binary detectors.

C. Discussion

It is worth noting that the model we chose for the ML App is by far not optimal. It was implemented this way to give a more comprehensive and fairer comparison and to remain consistent with the illustrative example. However, by adding only a notion of node history to the ML model, using only the last few messages of a node instead of only one, we were able to significantly increase the detection quality. In particular, the false positives ceases being much of an issue. Additionally, we found that the fingerprinting of attacks with CaTch over multiple sequential messages could be much more elaborated and complex.

Furthermore, even though CaTch can be give a considerable advantage for detecting subtle misbehavior, it also requires more computational power. And although the simple tests we did on our local unit are optimistic, more research needs to be conducted to evaluate how much computational power is available and needed and if the benefits outweigh the cost.

Additionally, one could assume that an attacker could manipulate the CaTch uncertainty factor to avoid detection. This is in fact possible and is a scenario that should be taken into account when designing the detection application. For example, an application should consider a plausible maximum value for the confidence range depending on the environment. However, CaTch is supposed to be replacing the legacy detectors. Additionally, it is worth noting that it is much easier for an attacker to just manipulate the sensor value to remain within the legacy detectors limits than to manipulate the value and the confidence range to do the same for CaTch.

VI. Conclusion

In this paper we focused on embedded misbehavior detection mechanisms in C–ITS. More precisely, we evaluate the impact of physical measure uncertainty on the plausibility detectors performance. We notice that such uncertainty information should be included in the standard Vehicle–to–X communication (V2X) messages. We theorize that utilizing it may lead to improving the misbehavior detection process.

To benefit from this available information, we propose CaTch, an enhanced version of plausibility detectors that takes into account the physical measure uncertainty. We showed that CaTch provides for each plausibility detector an uncertainty factor that indicates the levels of implausibility. We showed through extensive simulations that CaTch performs better or in the worst case similar than current misbehavior detectors. In particular, the uncertainty information can be used by intelligent misbehavior detection applications to improve the detection quality in the decision making process.

Future works consist of defining a smart misbehavior detection application that takes advantage of the uncertainty factor to decide whether or not reporting an ITS–S. Finally, the definition of decision algorithms at the centralized back-end server is also considered as a next step of this work.

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] B. Lonc and P. Cincilla, "Cooperative its security framework: Standards and implementations progress in europe," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–6.

[2] A. Vinel, N. Lyamin, and P. Isachenkov, "Modeling of v2v communications for c-its safety applications: a cps perspective," *IEEE Communications Letters*, pp. 1–1, 2018.

[3] I. B. Jemaa, A. Kaiser, and B. Lonc, "Study of the impact of pseudonym change mechanisms on vehicular safety," in *2017 IEEE Vehicular Networking Conference (VNC)*, Nov 2017, pp. 259–262.

[4] J. Kamel, A. Kaiser, I. B. Jemaa, P. Cincilla, and P. URIEN, "Feasibility Study of Misbehavior Detection Mechanisms in Cooperative Intelligent Transport Systems (C-ITS)," in *2018 IEEE 87th Vehicular Technology Conference: VTC2018-Spring*, Porto, Portugal, Jun. 2018.

[5] "ETSI EN 302 637-2 V1.3.2: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," *ETSI WG5 Technical Specification*, pp. 1–44, November 2014.

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

[7] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," *CoRR*, vol. abs/1804.06701, 2018.

[8] 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.

[9] R. W. van der Heijden, A. Al-Momani, F. Kargl, and O. M. F. Abu-Sharkh, "Enhanced position verification for vanets using subjective logic," in *2016 IEEE VTC-Fall*, Sept 2016, pp. 1–7.

[10] N. H. T. S. A. DEPARTMENT OF TRANSPORTATION, "Federal Motor Vehicle Safety Standards; V2V Communications A Proposed Rule by the National Highway Traffic Safety Administration," pp. 3854–4019 (166 pages), 2017.

[11] J. J. Anaya, P. Merdrignac, O. Shagdar, F. Nashashibi, and J. E. Naranjo, "Vehicle to Pedestrian Communications for Protection of Vulnerable road Users," in *2014 IEEE Intelligent Vehicles Symposium*, Jun. 2014, pp. 1–6.

[12] J. Kamel, I. B. Jemaa, A. Kaiser, and P. Urien, "Misbehavior Reporting Protocol for C-ITS," in *2018 IEEE Vehicular Networking Conference (VNC)*, Taipei, Taiwan, Dec. 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01917456

[13] 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.

[14] K. Joseph, "Framework for misbehavior detection (f2md)," 2018. [Online]. Available: https://github.com/josephkamel/F2MD

[15] M. Liebner, F. Klanner, and C. Stiller, "Active safety for vulnerable road users based on smartphone position data," *2013 IEEE Intelligent Vehicles Symposium (IV)*, pp. 256–261, 2013.

[16] J.-P. Monteuuis, J. Petit, J. Zhang, H. Labiod, S. Mafrica, and A. Servel, ""my autonomous car is an elephant": A machine learning based detector for implausible dimension," in *SSIC'2018*, 10 2018.