CARS: Context Aware Reputation Systems to Evaluate Vehicles' Behaviour

G. Costantino, F. Martinelli, I. Matteucci
IIT - CNR, Pisa,Italy
Email:firstname.lastname@iit.cnr.it

A. Bertolino, A. Calabrò, E. Marchetti
ISTI -CNR, Pisa,Italy
Email:firstname.lastname@isti.cnr.it

***Abstract*—The introduction of new generation ICT systems into vehicles makes them highly connected with the external World. As drawback, vehicle becomes potentially vulnerable to security attacks. Here, we consider a scenario in which Vehicular Networks and a Urban Network work together to realize a defence mechanism based on Reputation Systems. In this way, we are able to identify and isolate possible malicious vehicles acting that could send messages with the aim of reducing the availability of the network. We propose Context Aware Reputation Systems, CARS, able to identify insider attackers and isolate them taking into account contextual conditions derived from sensors spread along the entire urban network. Then, we experimentally evaluate CARS on a real data-set of mobility traces of taxis in Rome to compare the proposed systems with existing ones that do not consider contextual conditions. The preliminary results obtained are promising and show the feasibility and potentiality of CARS.**

***Index Terms*—Context-Aware Reputation Systems, Automotive, Vehicular-Security.**

I. Introduction

ICT technologies are pervasive in automotive systems. Vehicles can request assistance, consult the Internet and interact one another via mobile communication, Wi-Fi connections, or other communication protocols. Indeed, both *vehicle-to-infrastructure* (V2X) and *vehicle-to-vehicle* (V2V) communications mainly rely on WiFi connections. Due to the peculiarities of such connections, one of the most common attacks is the *Denial of Service* (DoS) attack [1]. The DoS attack is an attempt to make a communication network resource unavailable to its intended users, such as to temporarily or indefinitely interrupt services of vehicles connected with other vehicles or with the infrastructure.

The main goal of the proposed algorithms is to provide a method for vehicles to rapidly identify and eventually isolate malicious vehicles, taking into account also contextual conditions that may influence their behaviour. Our approach is designed to detect and overcome DoS attacks, whose aim is to reduce the availability of the system by sending a huge amount of messages on the network. However, it could also be the case that, to alert other vehicles about, for instance, a traffic jam, or some other issues that may occur on the road, a vehicle starts to send several alert messages. In terms of availability of the network the result is the same but, obviously, the intentions are completely different. Indeed, there are two main key aspects to consider in traffic analysis and consequently in reputation estimation: the *recurring traffic congestions*, which appear in the same area during the same time in specific (working/holiday) days; the *exceptional events* that can cause a sudden traffic volume increase. In the Denial of Service attack scenario, such contextual, real-time traffic information needs to be taken in consideration so to avoid from one side behavioural misinterpretations and from the other isolate the malicious vehicle. Thus, to model this situation, we propose here *Context Aware Reputation Systems*, CARS, to combine both the value of reputation calculated according to the number of forwarded and generated messages and the *criticality value* of the area in which a vehicle starts to generate messages to evaluate the behaviour of such a vehicle not only according to the number of generated messages [2] but also considering the contextual conditions. The final goal is to reduce *false positive* in the identification of possible malicious vehicles: messages generated by malicious vehicles are then dropped by other vehicles in the network, while the one generated by good guys are forwarded to spread useful information. Consequently, it allows drivers to improve traffic safety and road efficiency and, by guaranteeing the availability of communications network and road efficiency, it leads to a reduction of pollution, positively impacting on the environment.

The paper is structured as follows: next section presents the considered model of attacker's behaviour; §III describes the approach we propose to model and evaluate the criticality of different areas of a city, referring to the city of Rome; §IV presents CARS to identify and isolate a Denial of Service attackers and §V reports our experimental results. §VI discusses on related work in the area and, finally, §VII draws the conclusions and provides some ideas for future work.

II. Modelling the Attacker's Behaviour

Automotive domain is subject to several attacks that may affect both the privacy of the driver as well as the security of the vehicle also in a way that may impact on the safety of the driver itself and all the other passengers or pedestrians in a roadway. Among the others, one of the most common attacks that affects communication system as the one in place in the automotive scenario, we consider the *Denial of Service* attack (DoS) [1]. The aim of DoS is to weaken the availability of the communication network by transmitting a huge number of messages that overwhelm the network communication channels. In particular, we assume to have three infrastructures: the roadside infrastructure that communicate with vehicles through

Vehicle to Infrastructure communication (V2X), the *Vehicle to Vehicle* communications (V2V) infrastructure, and the *Urban Network*, a typical WSN-based Urban Traffic Management System (W-UTMS) [3]. These three infrastructures talk one another to exchange useful information about, for instance, the traffic, the weather conditions, and other information that may improve the quality of driving. In the considered scenario, a DoS attacker is going to prevent vehicles to communicate each other and with the Roadside communication network affecting the safety of system. To model the behaviour of an attacker, we consider the number of messages transmitting on the communication network as a key aspect of the DoS attack. For each vehicle circulating on the roadside, we distinguish two main sets of messages: *forwarded* messages received from a different vehicle and are only forwarded to neighbours; *generated* ones sent for the first time by the considered vehicle.

Hence, according to the *principle of collaboration*, a vehicle that sends generated messages more than forwarded ones is *suspected* to be an attacker. A *collaborative* vehicle behaves in such a way that the functionalities of the network are maintained [4]. According to the BUG threat model [5], a vehicle can assume three different behaviour:

- **Bad behaviour:** the vehicle is essentially malicious, acting as selfish to damage the network intentionally by neglecting or limiting communications.
- **Ugly behaviour:** the vehicle assumes an opportunistic behaviour according to its cost/benefit analysis.
- **Good behaviour:** the vehicle uses its resources to provide V2V communication nodes without any direct interest.

III. Evaluation of Context Attributes

As surveyed in [3], a typical W-UTMS involves four different activities: (i) information collection; (ii) data diffusion; (iii) processing of data to plan the required activities; and (iv) implementation of the suitable actions. At different degree of implementation, the UTMS involved different physical devices such as wireless sensors, Traffic Management Centres (TMC), Road Side Units (RSU), and On Board Units (OBUs) on vehicles. The possibility of the mobility of sensor nodes, the ability to withstand harsh environmental conditions, node failures, low power and scalability makes the W-UTMS adaptable and applicable to many different situation and environments. In this paper we mainly focus on the W-UTMS aspects concerning the sensors collecting real-time traffic information, like vehicle density, type of vehicle, average waiting time and pollution and the traffic data to the RSU [3].

Considering the recurring traffic congestions in literature there are different proposals for the monitoring of urban traffic such, for instance, that mentioned in [6]. In this paper, we do not want to rely on any specific urban traffic monitoring system implementation; we would like to provide a generic approach that can be exploited independently from specific infrastructure. Thus, considering the urban area divided into predefined sectors (for instance, each one of one kilometre square), we assume that we can on-line compute the traffic congestion for each of them and make overall mean value at specific time slot (for instance, every half an hour). These data can be used both for having and actual picture of the traffic congestion in the different sectors and for updating the database responsible for keeping the long terms statistics of the traffic congestion of monitored urban area.

Considering instead the exceptional events, according to US Federal Highway Administration report [7] there are mainly six sources of exceptional congestion: *bottlenecks* mainly due to roadway narrows; *traffic incidents*, caused by vehicle crashes and stalls; *work area* caused by road repairs, building of new roads and maintenance activities and so on; *inclement weather* due, for instance, to excessive rains, snowfall, fog and wind; *poor signal timing* that can occur for instance in case of faulty traffic light controllers, *rare events*, e.g., strikes, footraces or cycling races.

The traffic congestion and the exceptional events are impact factors for the computation of the reputation estimation. In particular, considering the urban area divided in sectors $S_{(x,y)}$, where x and y represent the GPS coordinates of the top left corner of the considered sector, the *traffic congestion* (TC) and *exceptional events* (EE) at fixed interval M_h in the 24 hours, the level of criticality in each sector $S_{(x,y)}$ at the interval M_h can be computed as:

$$C_{((x,y),h)} = TC_{((x,y),h)} + EE_{((x,y),h)} \tag{1}$$

Where $TC_{((x,y),h)}$ and $EE_{((x,y),h)}$ are the traffic congestion and exceptional events computed for the sector $S_{(x,y)}$ at the interval M_h.

In $TC_{((x,y),h)}$ is derived as the current variation of the traffic congestion with the respect to its mean value calculated for the same sector at the same time over specific reference period:

$$TC_{((x,y),h)} = (1 - \overline{\Delta TC_{((x,y),h)}})/2 \tag{2}$$

where $\overline{\Delta TC_{((x,y),h)}}$ is average value of computed instant deviation of the on-line value of traffic congestion from the statistically computed one ($\Delta TC_{((x,y),h)}$). Specifically, $\Delta TC_{((x,y),h)}$, and consequently $\overline{\Delta TC_{((x,y),h)}}$, varies in the range [-1,1], where 1 represents an increment of more than 100% of the traffic congestion, 0 is the neutral value, and -1 indicates a decrement of more than 100% in the traffic congestion. Consequently $TC_{((x,y),h)}$ varies in the range [0,1] with 0.5 as neutral value.

Considering instead the value $EE_{((x,y),h)}$, it is derived as the sum of exceptional events occurring in the sector $S_{(x,y)}$ at the interval M_h. It is therefore computed as

$$EE_{((x,y),h)} = \sum_{w=1}^{k} I_{(((x,y),h),i)}/k \tag{3}$$

where $I_{(((x,y),h),i)}$ is 1 if the i-th exceptional event for the sector $S_{(x,y)}$ at the interval M_h is verified and 0, otherwise, and k is the total number of exceptional events.

The criticality values for the sector $S_{(x,y)}$ at M_h varies in the interval [0,2]. The neutral value is 0.5, i.e., no deviation from the traffic congestion and no exceptional events.

In Figure 1 the overall set up is represented. In this case the different events (traffic congestion and exceptional events) are

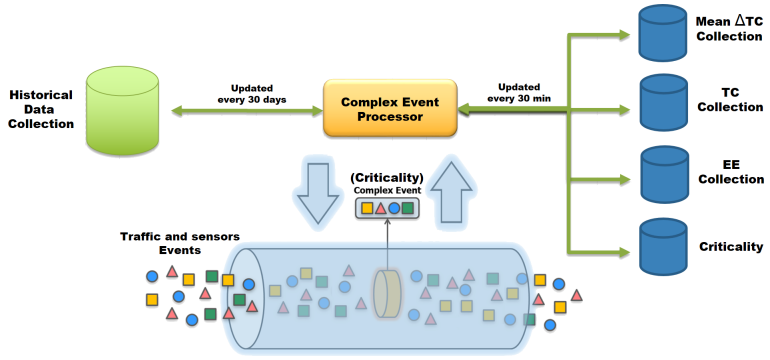


Fig. 1. Architecture set up.

collected for each sectors at each prefixed interval (30 min.). The data are used for:

- Deriving the on-line values of the $\Delta TC_{((x,y),h)}$ and computing at each interval M_h (30 minutes), for each sector $S_{(x,y)}$, the mean value of the data collected, i.e., $\overline{\Delta TC_{((x,y),h)}}$. The $\Delta TC_{((x,y),h)}$ is computed by the deviation of the on-line collected value and the corresponding one in the Historical Data Collection.
- Deriving, at each interval M_h (30 min.), for each sector $S_{(x,y)}$, the values of the traffic congestion $TC_{((x,y),h)}$, the exceptional events $EE_{((x,y),h)}$, and the criticality $C_{((x,y),h)}$ according to previous formulas.
- Updating at longer time intervals (e.g., every 30 days) the Historical Data Collection containing the mean traffic congestion of each specific sector. The data are derived as an average of the values collected for $\overline{\Delta TC_{((x,y),h)}}$ over period longer enough to avoid bias due to local and contingent situations.

IV. CARS: Context Aware Reputation Systems

Starting from the work in [8], [2], we propose *Context Aware Reputation Systems*, CARS, as an enhanced version of a reputation-based approach to identify and isolate a DoS attacker in a automotive scenario, taking also into account contextual attributes, as the one introduced in §III.

We consider the reputation as main information to understand the behaviour of vehicles, and more specifically, it is used as parameter to identify malicious vehicles. In the algorithm we propose, vehicles are able to observe the behaviour of other vehicles that they meet during the journey. In particular, we make the following assumptions: i) the roadside infrastructure acts as a trusted third party and it is able to communicate with the W-UTMS, ii) vehicles are able to observe the behaviour of the other that they meet during the journey and to communicate with the infrastructure, and iii) all communications preserve the integrity of transmitted data (both messages and local observations). The algorithm is composed of four steps: as a first step, each vehicle performs a *direct observation* of neighbours and evaluates their behaviour by registering the number of messages they *forward* and *generate*, as well as, in case of generated messages, taking also trace of where, in terms of latitude and longitude coordinates, and when, in terms of the time-stamp, each message is generated. Then (second step), at fixed time, each vehicle transmits the collected values to a central server belonging to the roadside infrastructure by using V2X communications. The central server calculates a single value of reputation for each vehicle that is travelling in the considered roadway by composing all observations received by vehicles and the evaluation of context attributes provided by the W-UTMS infrastructure. Finally, in the fourth step, the complete set of reputation values is sent to all vehicles that will receive updated reputation values of other vehicles they are able to communicate with. Once the DoS attackers have been identified as the ones having a Bad behaviour, *i.e.*, vehicles with a low reputation value in the broadcasting *Vehicles Global Observation* (VGO), the other vehicles in the network act as firewall by dropping its generated messages.

A. Step 1: Vehicles Local Observation

Each vehicle performs a *direct observation* of *close* vehicles and evaluates their behaviour by comparing the number of *generated* and *forwarded* messages. Vehicles are considered close when the distance between them is a single hop and there is a physical connection between their network interfaces. When a vehicle receives a message, it checks if it has been generated or forwarded by the sender vehicle and stores the information into its *Vehicles Local Observation* (VLO) table.

An example of VLO calculated by vehicle C is illustrated in the Table I, where, V_c says that the table belongs to C, while G_m and F_m show the number of Generated and Forwarded messages of met vehicles, such us, B, G, and others. Note that each vehicle can be uniquely identified by the other, for instance, through its license plate. In the column *Generated_Message_Info*, we collect all the information related to each Generated message: the ID of the message, the Latitude and Longitude in which it has been generated, the time-stamp of its generation, and the number of times it has been sent to, in this case, C.

B. Step 2: Collecting the VLO tables

Once VLO tables are populated and updated, each vehicle sends via V2X communication its VLO to the *Reputation*

TABLE I
EXAMPLE OF VEHICLES LOCAL OBSERVATION TABLE.

V_c	G_m	F_m	Generated_Message_Info				
B	5	19	Id	*Timestamp*	*Latitude*	Longitude	#
			ID_1	2014-02-01 00:00:01	41.9285433333333	12.4690366666667	30
			ID_2	2014-02-01 00:00:02	41.8910686119733	12.4927045625339	40
			...	...	...	...	
			ID_5	2014-02-03 00:00:00	41.7931766914244	12.4321219603157	70
G	5	16	Id	*Timestamp*	*Latitude*	Longitude	#
			ID_1	2014-02-01 00:00:01	41.9285533333333	12.4690366666667	30
			ID_7	2014-02-01 00:00:03	41.8910676119733	12.4927055625339	60
			...	...	...	...	
			ID_8	2014-02-03 00:00:00	41.7931776914244	12.4321119603157	50
...	...	...					

Server, RS in Figure 2, that belongs to the roadway infrastructure. We assume that, at fixed time, for instance every *30* minutes, vehicles send their VLO tables to the server, which collects them. We point out that communications with the server are not performed all at the same time, but they depend on the moment when a vehicles entered in the roadway sector under observation.

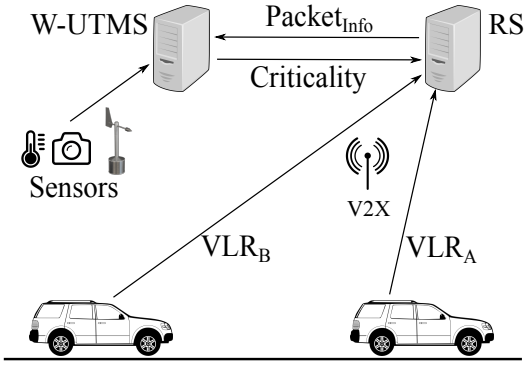


Fig. 2. Sending VLOs to the central server.

In Figure 2, we pictorially show the phase in which vehicles share their VLO with the server. When, the server receives VLOs, it populates its table that contains all G_m and F_m aggregated for each vehicle. It also use information of generated messages collected into VLOs to retrieve the criticality value of each generated message from the W-UTMS. Indeed, the criticality c_{m_i} of a message m_i is calculated as in Eq. 1, where x and y are latitude and longitude, and h is the time-stamp of the taxi when it generates the message. We call this table VGO (*e.g.*, Tab. II).

The strategy to populate the VGO table works in the following way. If the server receives a VLO that contains a new vehicle identity, i.e., information about a vehicle that is not already present in the VGO, the server just creates a new entry in the table and appends its values of G_m and F_m. On the contrary, if the server already knows a vehicle, the it updates the value of forwarded messages for a vehicle j to store in the VGO. So, the formula is the following:

$$F^j_{m_{new}} = F^j_{m_{old}} + F^j_{m_{rcv}} \tag{4}$$

where $F^j_{m_{old}}$ is the value of forwarded messages stored in the VGO, $F^j_{m_{rcv}}$ is the number of forwarded messages calculated locally, and $F^j_{m_{new}}$ represents the aggregation of the previous two values.

Regarding the aggregation of generated messages, the server takes also into account the context in which messages are generated by considering the messages' criticality, c_{m_i}, for message m_i, for each generated messages by the vehicle j. Then it updates those values in the following way:

$$G^j_{m_{new}} = G^j_{m_{old}} + \sum_{w=1}^{k} \sum_{i=1}^{t} e^{-c_{m_i}} * n \tag{5}$$

where $G^j_{m_{old}}$ represents the number of messages generated stored in the VGO for the vehicle j, k are the number of vehicles that send a local observation of the vehicle j, t represents the number of messages generated stored into the VLO for the vehicle j, n is the number of time the message m_i with criticality cm_i is sent by j, and $G^j_{m_{new}}$ is the new value for the number of messages generated by j.

C. Step 3: Vehicles' Reputation

As we presented above, the VGO contains an overall observation of the behaviour of each vehicle travelling in the roadway. Such a behaviour expresses the reputation taken by the vehicle up to the moment it has been established. Once the VGO table is completed, it is possible to calculate the reputation of each vehicle. It is worth noting that, according to different reputation systems, it is possible to differently evaluate the reputation of a vehicle.

Starting from the work in [8], [2], we enhance the two reputation functions by considering contextual attributes.

1) Context Aware Rational Reputation System: It is based on the percentage of the forwarded messages with respect to the totality of the sent messages [8]. According to the principle of collaboration, the reputation of each vehicle is calculated as the ratio of F_m and $F_m + G_m$. The formula that calculates the reputation of a generic vehicle j is:

$$Rep^j = \frac{F^j_m}{F^j_m + G^j_m} \tag{6}$$

TABLE II
EXAMPLE OF VEHICLES GLOBAL OBSERVATION TABLE WITH REPUTATION VALUES CALCULATED BY A REPUTATION SYSTEM S.

V	G_m	F_m	Generated_Message_Info				Rep(S)
			Id	*Timestamp*	*Latitude*	Longitude	#	
B	35	190	ID_1	2014-02-01 00:00:01	41.9285433333333	12.4690366666667	30	Rep_B
			ID_2	2014-02-01 00:00:02	41.8910686119733	12.4927045625339	40	
			...	...	...	...		
			ID_35	2014-02-03 00:00:00	41.7931766914244	12.4321219603157	70	
			Id	*Timestamp*	*Latitude*	Longitude	#	
G	50	300	ID_1	2014-02-01 00:00:01	41.9285533333333	12.4690366666667	30	Rep_G
			ID_7	2014-02-01 00:00:03	41.8910676119733	12.4927055625339	60	
			...	...	...	...		
			ID_50	2014-02-03 00:00:00	41.7931776914244	12.4321119603157	50	
...	...	...	...					

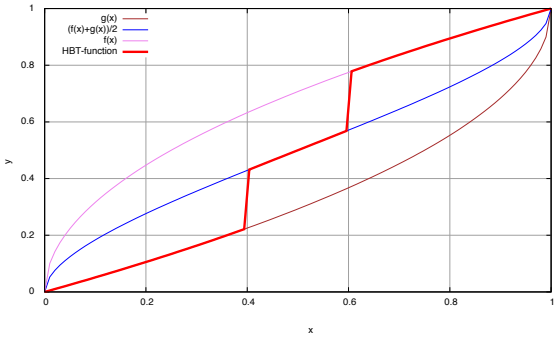


Fig. 3. Context Aware HBT Reputation Function.

Let F_m^j and G_m^j the aggregated values of forwarded and generated messages of the vehicle j stored in the VGO. Note that, even though Formula 6 is the same we have in [2], [8], the value G_m^j is calculated differently taking into account also contextual conditions (Formula 5).

Note that, even thought the reputation is calculated considering both newer and older message, our reputation algorithm does not imply that a malicious vehicle will be always tagged as malicious. In fact, by reverting to a collaborative behaviour, that vehicle will improve its reputation.

2) Context Aware History-based Trend Reputation System: It is based on the history of the behaviour of a vehicle. The intuition follows two main directions: i) increase and decrease the reputation of vehicle in accordance to its behavioural trend (slowly increase if the reputation is high and slowly increase if the reputation is low), and ii) dynamically characterize the behaviours of the BUG model.

To establish the reputation of an unknown vehicle, *i.e.*, a vehicle that for the first time interacts with some other vehicle in the roadside network, we consider a function that starts from a neutral value, *i.e.*, 0.5 in the range $[0,1]$. The reputation of a vehicle increases or decreases according to the following function:

$$HBT(x) = \begin{cases} g(x) & \text{if} \quad x \leq 0.4 \\ z(x) = (f(x)+g(x))/2 & \text{if} \quad 0.4 < x < 0.6 \\ f(x) & \text{if} \quad x \geq 0.6 \end{cases} \tag{7}$$

The server collects all the local observations and for each couple of vehicles T_i and T_W calculates the value $LO_j^{T_i,T_W} = \frac{F_j^{T_i}}{F_j^{T_i}+G_j^{T_i}}$ that represents the local reputation of T_i with respect to T_W at the time j and $G_j^{T_i}$ is calculated as in Eq. 5.

The global reputation at time j is calculated as follows:

$$GO_j^A = \frac{\sum_{i=1}^{n_1} g(LO_j^{A,i}) + \sum_{w=1}^{n_2} f(LO_j^{A,w}) + \sum_{v=1}^{n_3} z(LO_j^{A,v})}{n_1+n_2+n_3}$$

where $n_1+n_2+n_3$ is the total number of vehicles that met A. Vehicles that send local values greater than 0.6 contribute to the summation of $f(x)$, the ones that send local values lower than 0.4 contribute to the summation of $g(x)$, and local values between 0.4 and 0.6 are summed with $z(x)$.

The Context Aware History Based Global reputation is calculated as the average between the old value of the global reputation and the new one at time j.

D. Step 4: Identification of the DoS Attacker

We use the global reputation value of each vehicle in the network to characterize its behaviour with respect to the BUG model. Different reputation systems produce different characterizations. According to the Context Aware Rational Reputation system, we statically fix the reputation thresholds: we say that a $Rep^j \leq 0.3$ is an indication of an anomalous and Bad behaviour, for instance a *DoS* attack; $0.3 < Rep^j < 0.7$ is an Ugly vehicle; a reputation $Rep^j \geq 0.7$ for a Good behaviour.

Exploiting the Context Aware HBT Reputation system, in addition to reputation, we also calculate a global value denoting whether a vehicle in a certain instant in time t_i, according to the current observations should be considered collaborative or not. This is specified as a flag that assumes value 1 if the vehicles is not collaborative and 0 if it is collaborative. It is worth noting that, this information is strictly related to the criticality of the area in which the message has been generated, that is dependent on the time-slot too. A vehicle is not collaborative if:

1) at time t_i, the following formula holds

$$\frac{F_m^j}{F_m^j + \sum_{w=1}^{k}\sum_{i=1}^{h} e^{-c_{m_i}} * n} < 1 \tag{8}$$

2) considering the history of forwarded and generated messages, the sum of the differences between the number of forwarded messages at two consecutive observation instants is lower than or equal to the sum of the difference between the number of generated messages in the same interval of time:

$$\sum_{i=0}(F_{i+1}-F_i) \leq \sum_{i=0}(G_{i+1}-G_i)$$

where both G_{i+1} and G_i are calculated as in Eq. 5.

Hence, a vehicle has a Bad behaviour when the boolean value is set to 1 and the global reputation is less than or equal to 0.4. If one of this two conditions does not hold, *i.e.*, the flag is set to 1 but the reputation is greater than 0.6 or the flag is 0 and the reputation is less than 0.4, we have an Ugly vehicle. If the flag is set to "0" and the reputation is greater than 0.6, then the vehicle is a Good one.

V. Simulations

We developed a simulator to evaluate CARS and made a comparison with the reputation systems described in [2]. We used real mobility trajectories of vehicles coming from the dataset [9] whose trajectories were generated in a roadside network generated in Rome during February 2014. The original dataset is composed by 320 mobility traces of taxis collected for 30 days. We extrapolated from it a subsets of 199 taxis observed in a period of 48 *hours*, from *2014-02-01 00:00:01* to *2014-02-03 00:00:00*. This choice is motivated by the strong density of the traces that made our simulations ending on reasonable time considering, however, a significant number of taxi location-points[1]; in the simulation of 48h we got 557.611 location-points of taxi.

We assume that all taxis are able to communicate with the road infrastructure via the nearest access point, *e.g.*, a traffic light, which is already connected with the network infrastructure. In addition, vehicles exchange messages when their are close each other and exists a physical connection between their network interfaces.

A. Details on our Simulator

We wrote our simulator using Java[2] 8 and taxi-mobility traces were stored into a MySQL[3] database. The original mobility trace was downloaded from the *crawdad*[4] web-site as text file with the structure represented in Table III.

We wrote a Java function to store the entire dataset into a MySQL table to allow our simulator to seamlessly work with the mobility traces. When we run a new simulation, we start from the first item, or location-point, of the table, which corresponds to the first time-stamp, and we verify if there are contacts among taxis, *i.e.*, there are two close taxis. Two taxis circulating on the road are *close* if their distance is minor than *200* meters, which means that the two taxis may exchange messages. The distance[5] between two taxis is calculated using the latitude and longitude position of both taxis considering the earth radius equal to *6.378.800* meters.

0	1	2	3	4
ID	Hop Counter	Protocol Name	Time-To Live	Forwarder List

Fig. 4. Message fields.

Each time that two taxis are close, they forward all messages they have in their messages-buffer[6]. Messages exchanged are identified as: *forwarded* or *generated*. A taxi recognizes a message as *generated* when it verifies that the ID of the taxis, which sent the message, is the same ID of the taxi that generated the message. Instead, if the messages is received from an ID that is different from the one that has generated the message, then the message is labelled as *forwarded*. This distinction is used by the taxis to calculate the local observation (§IV-C). Also, every time that a taxi meets another one, the taxi that receives a message increases its counter by one whether the messages is generated or forwarded. Nevertheless, messages already present in the buffer are discarded but the counter is increased anyway.

Figure 4 shows the composition fields of messages transmitted during simulations: **ID** represents the message-ID, *i.e.*, a unique number for each message. Vehicles refer to this ID to check if the received message is already stored in their buffer; **Hop Counter** is the counter: it is increased by one every time that the message is received by a vehicle; **Protocol Name** is the way to select the reputation system to use during simulations; **Time-To Live** is expressed as time-stamp and it is the moment when the message is generated plus one day; **Forwarder List** saves vehicles ID that received the message during the forwarding. A message generator appears as the first vehicle in this list.

The upload of VLO tables to the Reputation Server, RS, is a relevant part of our simulations to calculate the VGO table. In our simulator, this phase is triggered every 30 minutes, moment in which each taxi offloads its VLO table to the RS. It calculates the global reputation of all taxis that generated or forwarded messages. As final step, the RS broadcasts every 30 minutes the VGO table to all taxis in such a way that they are able to establish a taxi's reputation. RS broadcasting tables is achieved using the *push*-model, i.e., the RS pushes the VGO to a vehicle by the active-connection established in the VLO upload phase.

To simulate the traffic congestion in the Rome context, we first derive the different sectors $S_{(x,y)}$ of the formulas in §III. In particular, we divided the considered geographic area into squares of one kilometre and we identified the coordinate (x,y) of the top left corner of each square.

Then, considering some already available studies such as [7], we derived the values representing the traffic congestion

[1]A location-point is an item of the dataset indicating an ID, longitude, latitude and time-stamp of a taxi's position

[2]http://www.oracle.com/technetwork/java/javase/overview/java8-2100321.html

[3]https://www.mysql.com/

[4]www.crawdad.org

[5]http://www.meridianworlddata.com/distance-calculation/

[6]In our simulator, we consider that the buffer can contain an unlimited number of messages.

TABLE III
TAXIS MOBILITY TRACES.

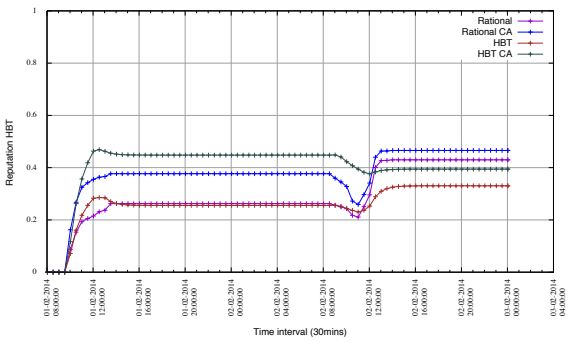
ID	*Timestamp*	*Latitude*	Longitude.
2	2014-02-01 00:00:01	41.9285433333333	12.4690366666667
7	2014-02-01 00:00:02	41.8910686119733	12.4927045625339
...	...	...	...
361	2014-02-03 00:00:00	41.7931766914244	12.4321219603157

for each sector $S_{(x,y)}$ at each interval M_h in the 24 hours, i.e., the data of the Historical Data Collection.

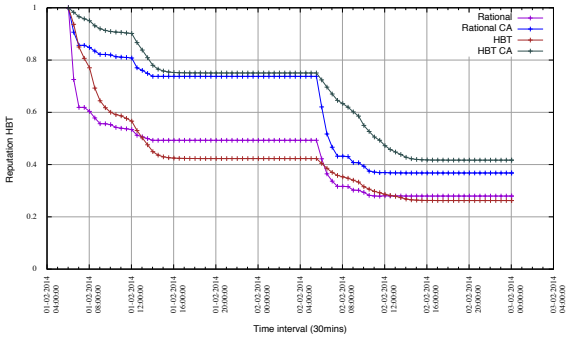
In particular, from the available documentation we considered that, a part from specific exceptions, usually in the early hours, i.e., from 1am to 5am, vehicle density is at lowest level because most people are sleeping. From this time being, the vehicle density starts (exponentially) increasing and tends to reach its maximum value at 8am, when most of the people go to the office, school and college and so on. Then, from 10am and 12am vehicle density decreases gradually for increasing again from 1pm to 2pm due to school and shopping closure. From 3pm, the density starts again to (exponentially) increase till its maximum value at around 5pm when most of people return back home. Then vehicle density decreases gradually and till an almost stable situation from 8pm and 12pm. Therefore, the peak hours are usually between 8am to 9am and 5pm to 6pm. Most of the congestion occurs at this time and is recursive almost for each working day.

For aim of simplicity, in this experiment we only consider traffic congestion during the working day. Similar analysis can be done considering non working days.

Using a random generator, we simulated different values of traffic congestion, to compute each 30 minutes for each sector $S_{(x,y)}$, the $\Delta TC_{((x,y),h)}$, and, consequently the $\overline{\Delta TC_{((x,y),h)}}$. Similarly, we simulated the exceptional events realization. Hence, we derive values of the traffic congestion $TC_{((x,y),h)}$, the exceptional events $EE_{((x,y),h)}$, and the criticality $C_{((x,y),h)}$.



(a) Homogeneous criticality.



(b) Variable criticality High reputation.

Fig. 5. Comparison among several reputation systems.

B. Results

To validate and compare our reputation systems with the ones presented in [2], we run several simulations that lasted 48 hours of time-trace. All taxi movements followed the trajectories written on the mobility traces and taxis generator were randomly selected from a list that we compose.

The results of the comparison among different reputation systems are graphically represented in Fig. 5. In particular, we consider a scenario in which three taxis act as generator (taxi with Id 2, 11, and 31). This assumption does not affect the simulation results, it just allows us to evaluate our reputation systems in recognising possible malicious behaviour. Then, we focus on taxi with ID 11 and compare CARS with reputation systems that do not consider context conditions. As a first study, represented in 5(a), we compare the four reputation systems (Rational, Rational_CA, HBT, and HBT_CA) by considering an homogeneous distribution of neutral criticality in both time and space (criticality equal 0.5). We note that, both context aware reputation systems reports highest reputation value with respect to the reputation system that do not consider contextual information, i.e., criticality value is equal to 0. This is in line with our intuition: according to the principle of collaboration, if something happens, taxis are encouraged to send an alert without being considered as potential malicious ones. Context aware reputation systems reward taxis that generate messages when the criticality value is high.

Secondly, we consider the same taxi (ID 11) that generates several messages and some of them, in particular the ones with Id 0, 2, and 4 have different criticality values (MessageCritical_ID0 = 1.3, MessageCritical_ID2 = 0.2, MessageCritical_ID4 = 0.7). In this case, we obtain four different curves Figure 5(b). In the context aware case, the history-based reputation is always higher than the rational one, while in case of criticality equal to 0, the two curves behaves differently. Hence the combination of historical behaviours and contextual conditions may help to better understand the real behaviour of a vehicle in a network, reducing the possibility of false positive. The advantage is that, vehicles that generate when is needed, i.e., when the criticality is high, are characterize as Good ones, so their messages are not discarded.

VI. Related Work

Most of the existing works are survey on reputation systems in Vehicular Networks or applications of existing solution to asses security in such network. In [10], the authors presented a systematic review of existing papers about trust models in VANETs from 2005 and 2014. All reviewed reputation systems assume that there is not a central entities able to collect local observations and acts as a trusted third parties. Also [11] presents a reputation system for Vehicular Ad-Hoc Network based on direct and indirect observation of the neighbours' behaviour. It does not consider a trusted third party involved in the system, hence indirect experiences are weighted according to the local trust of the forwarder node. The authors discussed on the impact of the reputation system on security aspect but they did not analyse a particular attack. In our approach, we assume that the roadside infrastructure acts as a central server able to compose local observations to obtain a global reputation of all vehicles in the system. In [12] the authors proposed a novel reputation management framework in automotive VANETS that integrates entity-centric and event-centric mechanism. As in our approach, they consider the roadside infrastructure as a central point to collect vehicles observations. The approach takes advantages from the knowledge of the mobility traces of most of the vehicles. In our approach, the history of each vehicle actively contributes to determine its reputation. We also discuss our approach about a specific security attack. The work in [13] presents a fully decentralized approach to compute the reputation of peers based on the traffic between a node and its peers, independently of these peers willingness to cooperate in calculation of their reputation. Apart from the different network communication, the main difference between this work and the one we propose is that it looks for the optimal peer for the communication, while we want to identify and isolate a possible attacker. Many protection mechanisms and frameworks have been developed to enforce security properties in the Vehicular network. In [14], the authors survey about security issues in VANET and sketch some possible solutions for some of them. For instance, to mitigate the DoS attacks, they proposed to switch between different channels or even communication technologies when one of them is brought down. However, this could be not feasible in case vehicles have not all the necessary technologies. Our approach overcomes this issue by identifying a possible attacker and dropping his messages (as a firewall).

VII. Conclusion and Future Work

In this paper, we enhanced two existing reputation systems, *Rational* and *History-Based* reputation systems, to consider also *contextual conditions* as a metric to evaluate the behaviour of vehicle in a vehicular network. The ultimate aim is to increase the precision in detecting, and eventually mitigating, a possible Denial of Service attacker, in such a way to reduce the probability of discarding important messages coming from a false positive malicious vehicle.

We have developed a simulator to compare CARS with reputation systems without contextual conditions and we have experimental evaluated them on a real dataset of mobility traces coming from taxis trajectories in Rome. Findings show that the context aware reputation systems increase the reputation of the vehicle when there is a real emergency situation (criticality value close to 2).

As future work, we aim to refine our results by considering more features retrieved by sensors. Moreover, since we are considering urban area of 1km square, we aim to use smaller urban squares or dynamically calculating the local criticality of each vehicle by considering its exact position in the map. Considering the problem of secure communication in VANETS, we also aim to propose a solution to preserve the integrity of the messages exchanged among vehicles, like VLO and VGO tables.

Acknowledgement

This work has been partially supported by the GAUSS (MIUR, PRIN 2015, Contract 2015KWREMX) and by the H2020 EU funded NeCS (GA 675320).

References

[1] M. Razzaque, A. Salehi, and S. M. Cheraghi, "Security and privacy in vehicular ad-hoc networks: survey and the road ahead," in *Wireless Networks and Security*. Springer, 2013, pp. 107–132.

[2] G. Costantino, F. Martinelli, and I. Matteucci, "Reputation systems to mitigate dos attack in vehicular network," in *Proc. of CRITIS2017*, October, 2017, p. to be published.

[3] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016. [Online]. Available: https://doi.org/10.3390/s16020157

[4] G. Bella, G. Costantino, and S. Riccobene, "Managing reputation over manets," in *Proc. of the 4th Int. Conf. on IAS 2008*, September, 2008, pp. 255–260.

[5] G. Bella, S. Bistarelli, and F. Massacci, "Retaliation: Can we live with flaws?" in *WORKSHOP ON INFORMATION SECURITY ASSURANCE AND SECURITY*, 2005.

[6] R. Du, C. Chen, B. Yang, N. Lu, X. Guan, and X. Shen, "Effective urban traffic monitoring by vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 273–286, 2015.

[7] Status of the nations highways, bridges, and transit: Conditions and performance; us federal highwayadministration report. [Online]. Available: https://goo.gl/CW1xsC Last access: 10 November 2017.

[8] G. Costantino, F. Martinelli, and I. Matteucci, "Exploiting vehicles' reputation to mitigate dos attack," in *Proc. of AMARETTO@MODELSWARD*, February, 2016, pp. 75–82.

[9] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "CRAWDAD dataset roma/taxi (v. 2014-07-17)," Downloaded from http://crawdad.org/roma/taxi/20140717, Jul. 2014.

[10] S. A. Soleymani, A. H. Abdullah, W. H. Hassan, M. H. Anisi, S. Goudarzi, M. A. Rezazadeh Baee, and S. Mandala, "Trust management in vehicular ad hoc network: a systematic review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 146, 2015.

[11] F. Dotzer, L. Fischer, and P. Magiera, "VARS: A vehicle ad-hoc network reputation system," in *Proc. of Int. Conf. WOWMOM*, June 2005, pp. 454–456.

[12] Q. Ding, X. Li, M. Jiang, and X. Zhou, "A novel reputation management framework for vehicular ad hoc networks," *Int. Jour. of Multimedia Technology*, vol. 3, no. 2, pp. 62–66, 2013.

[13] N. Stakhanova, S. Ferrero, J. S. Wong, and Y. Cai, "A reputation-based trust management in peer-to-peer network systems," in *Proc. of the 17th ISCA*, September 2004, pp. 510–515.

[14] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proceedings of the 3rd ACM workshop SASN*. ACM, 2005, pp. 11–21.