A machine learning based approach for the detection of sybil attacks in C-ITS

Badis Hammi*, Mohamed Yacine Idir†, Rida Khatoun‡
*EPITA Engineering School, France
badis.hammi@epita.fr
†Université Gustave Eiffel, France
myacine.idir@uge.fr
‡Institut Mines Telecom Paris, France
rida.khatoun@telecom-paris.fr

Abstract—**Cooperative Intelligent Transportation Systems (C-ITS) are gaining ground and are almost part of our everyday life. Unfortunately, such environments are increasingly the target to different attacks and sybil attacks are considered to be among the most dangerous ones. In this context, the intrusion detection systems are vital for the sustainability of C-ITS and the detection of sybil attacks are particularly challenging. Therefore, in this work, we propose a novel approach for the detection of sybil attacks in C-ITS environments. We provide an evaluation of our approach using extensive simulations that rely on real traces, showing our detection approach's effectiveness.**

Index Terms—**Certificate, C-ITS, PKI, Privacy, Pseudonym, Security, Sybil attack, VANET**

I. Introduction and problem statement

The Information Communication Technologies (ICT) have revolutionized the lives of people and are now pervasive in almost all fields of life. In this context, the Cooperative Intelligent Transportation Systems (C-ITS) are not an exception because of their capacity to improve the transport of people and goods. Indeed, they facilitate the driver's decision making tasks, and improve the users safety through a plethora of applications [1].

To support the different applications, a large number of messages are exchanged continuously between the stations (called Intelligent Transportation System's Station-Vehicle (ITSS-V) in the C-ITS context)[1] and the infrastructure (Intelligent Transportation System's Station-Road Side Unit (ITSS-R))[2]) in what is called V2X communications. Therefore, the correctness and reliability of the exchanged messages have a direct impact on the efficiency and effectiveness of these applications. Unfortunately, C-ITS applications through their messages can be the target of numerous security attacks and the sybil attack is considered to be among the most dangerous ones [1][2]. Figure 1 shows the scenario of a sybil attack where an attacker node creates different virtual nodes, also called sybil ghosts, in order to have a certain influence on the network's decisions especially in voting based protocols and applications. The creation of the sybil ghosts is performed through the creation of fake messages using different fake identities and different fake locations.

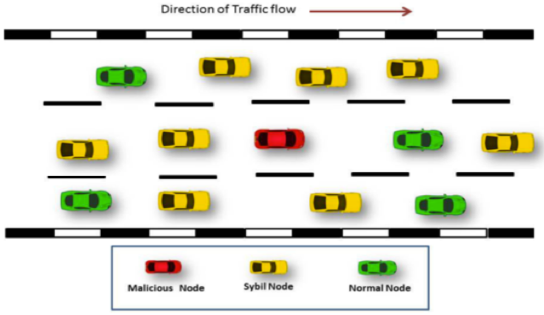


Fig. 1: Sybil attack: traffic congestion

Sybil detection approaches are divided into three classes: (1) position verification, (2) reputation and data-driven systems and (3) resource testing. In the position verification approach, the claimed position of each station is verified via the signal strength or via dedicated radars or sensors. In this context *Xiao et al.* [3], measure the signal strength of beacons received and compare them with the claimed position of a vehicle. These measures are performed by vehicles traveling in the opposite direction to avoid fake measures sent by the attacker. Benkirane *et al.* [4] proposed an approach where they assume that each vehicle on the road is linked to three reliable RSUs at a given time. Thus, when a vehicle broadcasts a message to other vehicles, the three RSUs also receive this message. The detection mechanism involves the collaboration of the RSUs. Indeed, based on the Received Signal Strength Indication (RSSI) measurements made by the three RSUs, the distances that separate the vehicle to each of the three RSUs at a given time is calculated. Since the messages of different sybil nodes are broadcasted by one physical node, each RSU receives the same RSSI values which allows the detection of the sybil nodes. However, due to the optimal positioning of the RSUs, it can be difficult if not impossible that each vehicle is always linked to three RSUs.

Reputation and data-driven systems rely on data collected from stations and generally does not require special hardware.

[1] In the remaining of this paper, we use the terms vehicle, node, and ITSS-V to refer to a connected vehicle.

[2] In the rest of this paper, we use the terms RSU and ITSS-R interchangeably to refer to a connected road side unit.

Bißmeyer *et al.* [5] proposed a central approach in which vehicles send Misbehavior Reports (MRs) to a central entity when detecting overlaps. These MRs contain signed evidence of the overlap and trust statements toward neighbors. The central entity analyzes all received MRs and then decides whether a node is a sybil ghost or not. In [6] Ayaida *et al.* proposed a detection approach whose key idea is that each vehicle monitors its neighborhood in order to detect an eventual sybil attack. This is achieved by comparing the real accurate speed of the vehicle and the one estimated using the Vehicle-to-Vehicle (V2V) communications with vehicles in the vicinity. This estimated speed is obtained using the traffic flow fundamental diagram of the road's portion where the vehicles are moving.

The resource testing approach assumes that physical entities are limited in resources such as computation, storage, and radio channels. Thus, in this approach, a typical puzzle is given to all stations to evaluate their resource availability. If one station is used to create and simulate multiple entities, then, it will be limited in responding to all puzzles. For example, *Raj et al.* [7] proposed a detection method that relies on proofs of work and location. The main goal here is that when a vehicle encounters an RSU, it will be authorized by a timestamped tag which is a concatenation of time of appearance and the anonymous location tag of that RSU. As the vehicle keeps moving, it creates its trajectory by incorporating a set of consecutive authorized timestamped tags that are chronologically chained to each other. This trajectory is used as an anonymous identity of the vehicle. Hence RSUs have the main authority to provide proof of location to vehicles. However, this technique is not suitable for a heterogeneous environment such as C-ITS. Furthermore, an attacker can easily have more computational resources compared to legitimate nodes or have more radio transmitters [1][8].

In our previous work [1] we presented an extensive state-of-the-art review and analysis of the solutions aimed at detecting sybil attacks in C-ITS. Throughout our analysis we showed that most of these works are either outdated or are not adapted to current C-ITS infrastructures and standards and hence proved that sybil attacks still represent an open issue. We also provided a network and attack models as well as the requirements to be considered when proposing a sybil detection approach. Finally, we provided one dataset for an urban scenario and another dataset for a highway scenario that can be used by researchers in future works. The work that we present in this paper is the continuation of the work cited above [1]. Indeed, we propose a novel approach to detect sybil attacks in C-ITS environments. Our approach meets the requirements identified and the performance results obtained demonstrate its efficiency and effectiveness.

II. Proposed approach

A. Network model

We consider a C-ITS environment having a set of ITSS offering and using different ITS services in a centralized or a distributed architecture. Each ITSS communicates with a large number of other ITSSs. The communication network used is unreliable and potentially lossy (e.g., 802.11p or ITS-G5). We assume that all entities on the network are not trustworthy. Indeed, the high number of stations in the network increases the risk of including compromised ones. The network function only forwards packets and does not provide any security guarantee such as integrity or authentication. Thus, a malicious user can read, modify, drop or inject network messages.

According to IEEE [9] and ETSI [10] standards, the network relies on a C-ITS Public Key Infrastructure (PKI) to ensure security management in the network. The C-ITS PKI comprises a Long Term Certificate Authority and a Pseudonym Certificate Authority that supply ITSSs with certificates. The ITSSs never use the Long Term Certificate (LTC) for communication but only to authenticate to the PKI in order to request new Pseudonym Certificates (PC). The PCs are continuously used because each packet must be signed by a private key associated with a public key certified by a PC. To comply with the privacy (and non-tracking) requirements, each ITSS must change its PC as well as all the network identifiers (e.g., IP address, MAC address, station ID, and so on) multiple times during a trip. The European standard ETSI TS 102 867 recommends that pseudonyms are changed every five minutes, whereas the American standard SAE J2735 recommends that this is done every 120 seconds or 1 km, whichever occurs last. The PCs of a given ITSS can only be linked by dedicated authorities (e.g., the Linkage Authority) and cannot be linked by other stations.

B. Detection system operation

In our approach, the RSUs ensure the detection task and the execution of the detection algorithm. Such a choice is considered because (1) the RSUs can have a large computation and storage capacity and (2) to remove the burden of the detection algorithm from the constrained devices that are the vehicles. Indeed, each RSU provides a local detection at its level. Then, the RSUs within a given region collaborate together to provide a global detection. In this work, we describe the local detection process at a given RSU level.

An RSU receives the data transmitted by the stations within its coverage zone. Each station is identified by a pseudonym certificate. Therefore, if a station changes its pseudonym certificate, the RSU will consider the data received as transmitted from two different stations.

The first step of the detection process, consists in data monitoring and collection where the RSU collects data from the different stations. We recall that each vehicle broadcasts continuously messages (e.g., in ETSI based architectures each station sends at least 10 Cooperative Awareness Messages (CAM) messages per second [11]). For each station, the RSU creates a data matrix that we note D_t^s (the matrix that an RSU creates using the data collected from a station s at a given time t). D_t^s contains the different information related to the station. For example, in our evaluation presented in Section III we collect the following data: time, vehicle's identifier, latitude, longitude, speed, and acceleration. However, other parameters

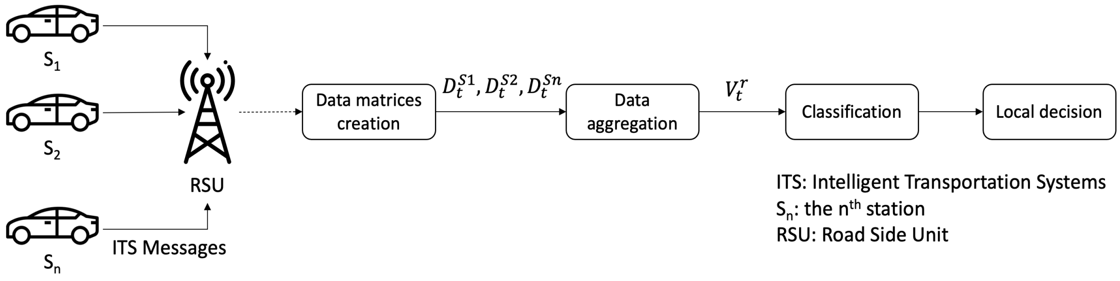


Fig. 2: Detection process steps

can be considered. The exhaustive list of the different metrics to consider is available in IEEE BSM [12] and ETSI standards [11].

The second step consists in data aggregation. Indeed, each data matrix D_t^s is transformed into a vector noted V_t^s (the vector of data aggregated and relative to the station s at a given time t). For the quantitative metrics like the speed and the acceleration, the average and variance are computed. We compute another metric called the jerk which describe how an object's acceleration changes with respect to time. V_t^s comprises the average and variance of the jerk of a given station. Moreover, the amount of distance traveled within the capture time is computed relying on geolocation metrics of the first and last received messages of a given station when passing by an RSU. The total traveling time related to this distance is also computed. Finally, all the vectors relative to the set of all the stations monitored are grouped into a matrix that we note as A_t^r (the matrix of data aggregated by the RSU r at a given time t) as presented below:

$$A_t^r = \begin{bmatrix} ID_{s_1}, T_{s_1}, d_{s_1}, \overline{Sp_{s_1}}, \sigma^2 Sp_{s_1}, \overline{Ac_{s_1}}, \sigma^2 Ac_{s_1}, \overline{J_{s_1}}, \sigma^2 J_{s_1} \\ \vdots \\ ID_{s_n}, T_{s_n}, d_{s_n}, \overline{Sp_{s_n}}, \sigma^2 Sp_{s_n}, \overline{Ac_{s_n}}, \sigma^2 Ac_{s_n}, \overline{J_{s_n}}, \sigma^2 J_{s_n} \end{bmatrix}$$

Where n is the number of the stations monitored. s is the station considered, T is the total time measured (from the first message received to the last one), d is the total distance that a vehicle travels (according to the geolocation information), Sp is the speed, Ac is the acceleration, J is the jerk, $\overline{x}$ is the average of x and $\sigma^2 x$ is the variance of x.

The third step is the classification of the stations' activity where the aggregated data matrix A_t^r is fed to a classifier. The latter decides for each station if its activity is considered as malicious or not. The choice of the classifier is left to the implementer according to his preferences and to the capacity of the available hardware and software. Nonetheless, the choice of the classifier can have consequences on the detection quality and accuracy as we show in Section III-C.

Following this classification, the RSUs of a given road/region collaborate with each other and with the linkage authority (to link the different pseudonym certificates related to the same stations) of the PKI for more investigation. However, we do not describe this phase in this paper. The Figure 2 describes the steps of the detection process at an RSU level.

III. Performance evaluation and discussion

A. Attacker model and evaluation scenarios

The C-ITS environment relies on wireless communications. Therefore, in this work, we assume that an attacker or malicious user has total control over the network used, i.e., the attacker can selectively sniff, drop, replay, reorder and delay messages arbitrarily with negligible delay. We also assume that the attacker has a pool of valid PCs. For instance, the attacker can obtain these PCs by tampering with the storage device of an ITSS. Besides, the attacker can benefit from increased computation power and storage compared to the existing devices.

All the messages are signed. Thus, the attacker cannot modify existing messages. However, since the attacker has a set of valid certificates the attacker can change the signature and modify the fields as needed, or can create new packets. Knowing that the certificates are pseudonym identities and are not linkable, the majority of the receiving entities (stations and services) will not notice that these are packets sent from an attacker.

Within a network, devices can receive unaltered and altered messages. Therefore, we evaluate our detection approach regarding different rates of altered messages ranging from 10% of additional sybil ghosts to 50% [1]. Moreover, we evaluate our detection proposal against different sybil scenarios with different difficulties:

1) Sybil scenario with random values: this scenario represents the case where the attacker forges new packets, with random values in the fields and broadcast them. As described by [13], this scenario can be used to launch a Denial of Service (DoS) attack where the motivation behind such an attack could be to overwhelm the misbehavior detection system of neighboring ITSS or that of the platform or just to disturb the network's communications. Moreover, it is one of the most commonly used scenarios for the evaluation of numerous existing approaches. Thus, to be stealthy than the discussed approaches, we propose that the attacker does not generate any random data that can make the detection easy. Instead, we propose that the attacker uses the same geolocation data as other captured packets. Moreover, for the speed, acceleration and so on, the attacker can use a random value between the maximum and minimum values that he observes on the network during the attack period.

2) Sybil scenario with static values: this scenario represents the case where the attacker simulates a traffic congestion. To be stealthy, the attacker captures some packets in the targeted area, then changes some fields in the packets (e.g., sets the speed and the acceleration to zero and modifies the signature and the timestamp). Next, the attacker broadcasts these modified packets and repeats the process of changing the signature but without changing the coordinates, heading, and so on, until the end of the attack.

3) Sybil scenario with replayed values: in this scenario, the attacker continuously captures traffic packets, changes their signatures, and timestamps, but keeps their movement data such as coordinates, speed, heading, acceleration and so on and broadcasts them. This scenario can be considered as the highest difficulty level for a detection scheme because it uses a realistic traffic model. The detailed description of these attack algorithms can be found in [1].

B. Datasets and evaluation framework

For the evaluation of our approach we use the two datasets provided by [1]. Each dataset represents the data collected within 1 Km^2 of range (range of an RSU) and for 24 hours[3]. The first dataset presents an urban scenario that describes the activity of 62,421 vehicles, and the second presents a highway scenario that describes the activity of 24,326 vehicles.

For the evaluation of our approach, we injected the station's traces described above into a simulator built on the R tool[4]. For each scenario (urban and highway), we tested the three sybil attack scenarios described in Section III-A. In the reminding of this paper we note the sybil scenario with random values as "Scenario 1", the sybil scenario with static values as "Scenario 2" and the sybil scenario with replayed values as "Scenario 3". For each of these attack scenarios and for both environment scenarios (urban and highway), we realized five experimentations where we vary the amount of additional sybil ghosts from 10% to 50% (10, 20, 30, 40 and 50%).

C. Evaluation results

As we explained in Section II-B, the third step of our detection process needs a classifier to classify the data transformed by the second step as malicious or not. In this section we evaluate the impact of four different classifiers. We have chosen common classifiers, with a gradual descriptive power ranging from logistic regression to neural networks, which are often considered as black boxes. More precisely, we used (1) Logistic Regression (LR), (2) Support Vector Machine (SVM), (3) Random Forest (RF), and (4) Artificial Neural Network (ANN). **Logistic regression** is a statistical model for studying the relationships between a set of qualitative variables X_i and a qualitative variable Y. It is just a generalized linear model using a logistic function as a link function. Nevertheless, it has good descriptif power, and significantly good results. **Support vector Machine** are used in a variety of applications (bioinformatics, information retrieval, computer vision, finance, and so on). They can be used for regression problems as well as classification problems like ours. There are default sets of hyperparameters. The latter are very few in number: they are limited to the choice of the regulation technique who serves as a degree of importance that is given to misclassifications. In our study, we only use a linear kernel. **Random forest** is nothing more than a set of decision trees. Each tree is trained on a subset of the dataset and gives a result (malicious or not in our example). The results of all the decision trees are then combined to provide a final answer. Each tree "votes" (yes or no) and the final answer is the one with the majority of votes. In our study, a number of 500 trees was chosen, varying this number didn't provided different results. An **Artificial Neural Network** is conceptually inspired by a biological neuron and its functioning. In a network, a neuron transfers an output according to its inputs, weighted by a synaptic weight evolving during learning (synaptic plasticity), the whole network defining a function of the explanatory variables. We can choose several activation functions, as well as several hidden layers, in this study we chose the logistic activation function, and a hidden layer with 3 neurons.

For the set of all the experimentations provided in this work, we performed a 5 folds cross validation technique to measure the classification efficiency of the different models.

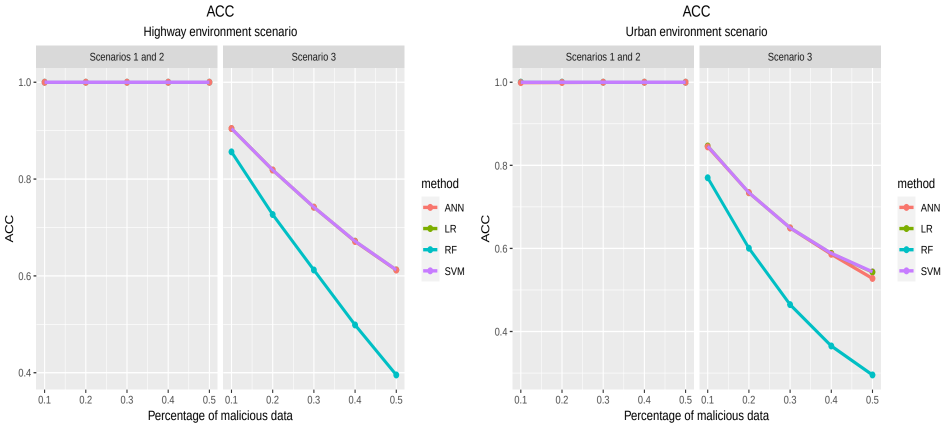
The binary dependent variable Y = "malicious or not" is fed to the classifier with different values of explanatory variables. After a training period, the classifier is given only the explanatory variables, the output of the model is a number between 0 and 100. 0 representing a non-malicious station and 100 a malicious station. A threshold is then chosen (usually 0.5) to obtain the binary result. The retained performance metrics are then computed from the output of the model and the reality.

We have computed the confusion matrix of all the simulations we conducted. A confusion matrix contains information about actual and predicted classifications that a detection system provides. From the latter, we have calculated different statistical indicators such as Receiver Operating Characteristic (ROC) curves, Accuracy, Error rate, Positive and Negative Predictive Values.

The accuracy (ACC) of a measurement system is the degree of closeness of measurements of a quantity to that quantity's true value. It has a value between 0 and 1. Figure 3. describes the accuracy values obtained over the experimentations realized on both environment scenarios with the four classification techniques. For Scenarios 1 and 2, we note that the accuracy we obtained is in the interval $[0.98, 1]$ for (1) both environments; highway and urban, (2) for whatever value of additional sybil ghosts from 10% to 50% and (3) for all the used classification techniques. Therefore, our detection system is very effective and accurate in detecting the attacks generated by the sybil scenario with random values (scenario 1) and the sybil scenario with static values (scenario 2). This is mainly due to the aggregation method we use, which turns out to be very efficient in the discrimination of attack data. We recall that these two attack scenarios are the most used ones when

[3] https://github.com/BadisHammi/C-ITS_Datasets

[4] http://www.r-project.org



(a)

(b)

Fig. 3: Accuracy of the detection algorithm for: (a) Highway scenario; (b) Urban scenario

perpetrating sybil attacks [13][1]. Which makes our detection approach very efficient against the most common form of the sybil attack in C-ITS environment.

For the sybil scenario with replayed values (scenario 3) and considering the experimentations realized on the highway environment data, the accuracy of the detection algorithm varies from 0.9 to 0.6 while varying the amount of sybil ghosts. More precisely, the accuracy values show good detection performance when the amount of additional sybil ghosts is less than or equal to 25% and show average detection performance when the amount of additional sybil ghosts is more than 25%. These results are obtained for the classification methods that are ANN, LR, and SVM. Relying on the same classification techniques, we obtained very similar results for the accuracy of the detection algorithm when using data from the urban scenario. More precisely, the accuracy results vary from 0.84 to 0.55 with a good accuracy when the amount of additional sybil ghosts is less than or equal 22% and an average accuracy performance for the rest. However, the application of the Random Forest (RF) provides less efficient accuracy results than the ones obtained from the other classification techniques. That is, the accuracy results obtained vary from 0.86 to 0.4 for the highway environment and from 0.78 to 0.3 for the urban scenario.

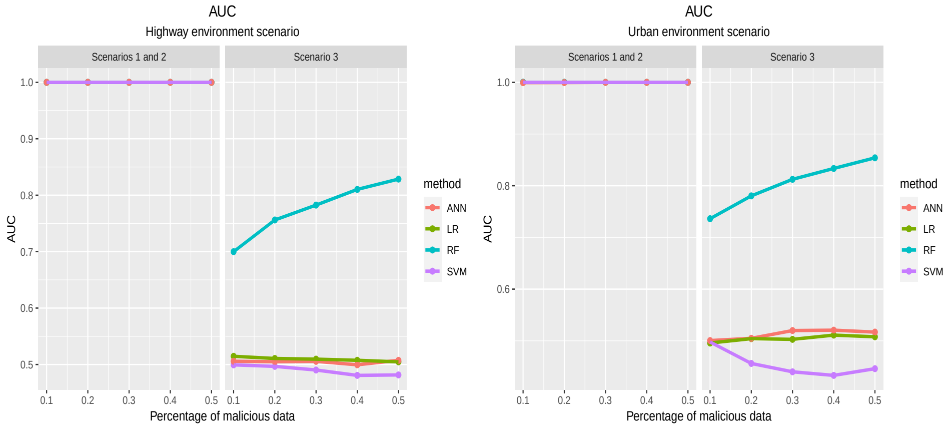
Relying only on the accuracy metric cannot provide a real indication on the performance of a detector. Hence, we analyse the Receiver Operating Characteristic (ROC) curves obtained from the different detections provided. The ROC curve represents a measure of the performance of a binary classifier. Graphically, the latter is represented in the form of a curve which gives the rate of true positives according to the rate of false positives.The Area Under Curve (AUC) value of a ROC curve reflects the detection performance of the evaluated system. Closer is the AUC to 1, better is the detection. The Figure 4 describes the different AUC of ROC curves obtained through all the realized experimentations.

For scenarios 1 and 2, we note an AUC = 1, for all the classification methods used and for both environment scenarios, which confirms the efficiency of our detection algorithm in detecting the most common form of the sybil attack in C-ITS environments. However, for scenario 3, the use of ANN, LR and SVM leads to poor detection performances. Sole RF realizes good AUC results that reach 0.83 for the highway environment scenario and 0.86 for Urban environment scenario.

In fact, SVM, LR, and ANN classifiers fail to distinguish malicious data and tend to classify all data as negative (which maximizes accuracy, since most data is actually negative), resulting in a ROC curve with little discriminative ability, which is somehow logical because malicious data is a copy of the legitimate data. RF in the other hand manages to predict two different classes, resulting in a different ROC curve and a better AUC, that is a better discriminative ability. The performance of RF in AUC is paradoxal with the performance of the same RF in computing the accuracy while we obtained poorer ACC results. Indeed, while RF manages to predict the two different classes, this distinction of the two parts is not a guarantee of good prediction. In case of bad prediction, we will have a worse ACC, but a higher AUC for its distinction ability. Moreover, the overall accuracy is based on a specific cut-off point, while the ROC tries all cut-off points and plots sensitivity and specificity. Hence, when we compare overall accuracy, we are comparing accuracy based on a cutpoint (0.5 in this study). We conclude that RF does predict two different classes but does not manage to do better than a complete prediction of one class.

Discussion

From the validation results, we can observe that the detection approach proposed is very efficient in the detection of the most common form of the sybil attacks in C-ITS (scenarios 1 and 2) with an ACC $\simeq 1$ and an AUC = 1. However, it is less efficient in the detection of the sybil attacks of replayed scenarios (even if the use of random forests classification



(a)

(b)

Fig. 4: AUC for: (a) Highway scenario; (b) Urban scenario

technique guarantees fairly good detection results). Thus, we conclude that at the level of one RSU, we cannot detect efficiently all the forms of the sybil attack. In fact, as described above, in this paper, we describe only the first part of our detection system which is the local operation part at the RSU level. Indeed, to be more efficient, a global detection process that involves the collaboration and cooperation of multiple stations mainly adjacent RSUs is needed.

It is also worth noting that we cannot compare our approach's performances with the state of the art existing detection approaches, because they do not experiment the same sybil attack scenarios. Which can lead to unfair comparisons.

IV. Conclusion and future works

In this paper, we have tackled the problem of sybil attacks in C-ITS environments. To remedy this problem, we proposed a detection approach that comprise four steps and that rely on machine learning classifiers. We validated our approach through simulations that rely on real traces. The results obtained prove the efficiency of our approach to detect the most common form of sybil attacks in C-ITS. However, it is less efficient against complex attack scenarios. Furthermore, the results were obtained through a centralized approach. However, a centralized detection approach cannot scale with a highly distributed and decentralized environment such as C-ITS.

The results presented in this paper represent only a step of our detection system. That is, the local detection at the RSU level. Therefore, our short-term future work will focus on proposing a fully decentralized and distributed approach of our detection algorithm. The latter will ensure a collaboration between the RSUs and provides a method to share the detections performed locally to reach a global decision and to be more efficient in the detection of complex attack scenarios. It will also define the cooperation method with the linkage authority of the PKI.

References

[1] Badis Hammi, Yacine Mohamed Idir, Sherali Zeadally, Rida Khatoun, and Jamel Nebhen. Is It Really Easy to Detect Sybil Attacks in C-ITS Environments: A Position Paper. *IEEE Transactions on Intelligent Transportation Systems*, page 15, 2022.

[2] Fatih Sakiz and Sevil Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad Hoc Networks*, 61:33–50, 2017.

[3] B. Xiao, B. Yu, and C. Gao. Detection and localization of sybil nodes in VANETs. In *DIWANS'06 Proceedings of the 2006 Workshop on Dependability issues in Wireless Ad hoc Networks and Sensor Networks*, pages 1–8, Los Angeles, California, September 2006.

[4] Said Benkirane. Road safety against sybil attacks based on rsu collaboration in vanet environment. In *International Conference on Mobile, Secure, and Programmable Networking*, pages 163–172. Springer, 2019.

[5] N. Bißmeyer, J. Petit, J. Njeukam, and K. M. Bayarou. Central misbehavior evaluation for VANETs based on mobility data plausibility. In *VANET'12 Proceedings of the 9th ACM International Workshop on Vehicular Inter-networking, Systems, and Applications*, pages 73–82, Lake District, UK, June 2012.

[6] Marwane Ayaida, Nadhir Messai, Sameh Najeh, and Kouamé Boris Ndjore. A macroscopic traffic model-based approach for sybil attack detection in vanets. *Ad Hoc Networks*, 90:101845, 2019.

[7] G Anitha F Stephen Raj. Detection of Sybil attack in VANET. *Karpagam JCS*, 14(2), 2020.

[8] Bo Yu, Cheng-Zhong Xu, and Bin Xiao. Detecting sybil attacks in vanets. *Journal of Parallel and Distributed Computing*, 73(6):746–756, 2013.

[9] Intelligent Transportation Systems Committee & others. IEEE Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages. *IEEE Vehicular Technology Society Standard*, 1609.2:1–884, January 2016.

[10] ETSI TS 103 097 V2.1.1: Intelligent Transport Systems (ITS), Security header and certificate formats; Release 2. page 22, October 2021.

[11] 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. November 2014.

[12] DSRC Committee. On-Board System Requirements for V2V Safety Communications. *SAE Standard J*, 2945/1:114, Issued 2016-03, Revised 2020-04.

[13] Joseph Kamel, Farah Haidar, Ines Ben Jemaa, Arnaud Kaiser, Brigitte Lonc, and Pascal Urien. A Misbehavior Authority System for Sybil Attack Detection in C-ITS. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 1117–1123. IEEE, 2019.