

A Practical Evaluation Method for Misbehavior Detection in the Presence of Selfish Attackers

Marek Wehmer^a and Ingmar Baumgart

FZI Research Center for Information Technology, Karlsruhe, Germany

Keywords: Evaluation Metric, Misbehavior Detection, VANET, Vehicular Network, Cyber Physical Systems Simulation.


Abstract: With recent deployment activities of Vehicle-to-X systems, the need for practical misbehavior detection is growing. The academic discussion on related topics has progressed in the last years and delivered new evaluation approaches. Most research however concentrates on evaluations based on the confusion matrix of packet classifications, such as the precision-recall graph. We show that this approach has fundamental limitations and does not allow to derive valid statements about the real-world impact of a scheme. After reviewing the state of the art, we show that the physical manifestation of attacks must be considered when evaluating misbehaviour detection systems. We propose a shift of perspective towards an attacker-oriented evaluation and contribute a new metric for selfish attackers based on the physical impact of the attack. We further present a simulation framework to practically evaluate misbehavior detection systems.

1 INTRODUCTION

The roll-out of Vehicle-to-X communications is gathering momentum. As a means of greatly reducing traffic accident numbers and efficiency, expectations are high. Inter-vehicular communication will impact safety relevant control systems and must therefore be considered a critical resource in vehicles. Security considerations play an important role in the design and implementation of such systems: Attacks and mitigation strategies have been a research topic for the last decade. The decentralized nature and latency requirements imply particular challenges for the design of security mechanisms. Messages conform to a small set of fully specified formats and convey a claim about the physical world. Moreover, all messages are signed by authorized participants, but not encrypted. The signature implies a certain degree of trust in the received message, but cannot be fully relied upon: As in any public key infrastructure, secret keys can leak and be abused by malicious actors to send misleading information. But even bugs and faulty sensors can lead to wrong claims being signed and sent out by legitimate participants of the network. In this setting, misbehavior detection fulfills a critical task: incorrect information must be filtered out, while correct information must be trusted to prevent accidents and increase traffic efficiency. One building block of

maintaining safety properties is to filter out packets with untruthful information. Misbehavior detection systems are designed to recognize such attacks and mitigate their impact.

Evaluation of misbehavior detection schemes has been a tedious task and proposed schemes have usually been assessed with specific simulation scenarios or analytically, because real-world evaluation is difficult and infeasible in many cases. Many contributions use the false-positive rate (FPR) and the false-negative rate (FNR) of classifications of received packets to evaluate the quality of schemes. Recent advances in evaluation methodology have improved the comparability between schemes by using a common reference dataset and also set the path towards better metrics. Those metrics are however still based on the confusion matrix, and therefore refer to the binary classification performance of all received packets during a simulation run. This approach has fundamental limitations in some cases. For example, the relevance of misclassifications is not necessarily distributed equally among participants. For some vehicles, a specific packet sent by the attacker might just not have any effect, while it can be critical for the safety of another. Some attackers may also try to send a great amount of packets to convince a specific vehicle of one misinformation. For their success, it does not necessarily matter if their target accepts all the packets or just one.

^a  <https://orcid.org/0000-0001-5155-8934>

In order to understand why it is impossible to comprehensively assess misbehavior detection with the confusion matrix, a new perspective on misbehavior is needed. Attackers sending misleading or incorrect information about the state of the world are not trying to get as many packets accepted as possible—their goal is to alter the state of the physical world in their favor. The success of their attack and in consequence the ability of any misbehavior detection system (MDS) to mitigate this attack can thus only be measured by considering the effect of the communication on the physical world. This aspect has rarely been considered and requires new tools to interactively simulate cooperative intelligent transport systems (C-ITS) in the presence of attackers.

2 RELATED WORK

Attacks and misbehavior detection in vehicular networks have been studied extensively in the last years (van der Heijden et al., 2019; van der Heijden, 2018). Many approaches of detecting attacks have been published, but most focus on detecting very specific attacks. As such, results are difficult to compare and usually authors do not share a common evaluation approach or even metrics that allow for comparison. Moreover, the implementation is often not available, which makes reproducing results difficult. Recently, the VeReMi dataset containing communication traces of different attacks being conducted in the LuST (Codeca et al., 2015) traffic scenario has been published and extended to improve evaluation quality in the field and provide a basis for comparison (van der Heijden et al., 2018; Kamel et al., 2020). The simulation code is based on the Veins vehicular network simulator (Sommer et al., 2011) and publicly available.

The choice of metrics for evaluating the quality of mechanisms is a related topic that is being discussed in the community. The authors of VeReMi propose to use the precision / recall graph of all reception events (van der Heijden et al., 2018) instead of the FPR / FNR metrics that have been used by many works before. While such metrics are easy to obtain in simulations, the numbers are difficult to interpret and even though they seem applicable for most data-centric approaches and suggest comparison to some extent, not all mechanisms can be measured accurately. The evaluation depends on further subtleties, such as the aggregation method and a definition of when a message is to be considered as malicious (van der Heijden and Kargl, 2017), which further weakens comparability between mechanisms. To

take the dispersion of errors in detection performance between participants into account, the gini-index of the FPRs / FNRs of different vehicles has been proposed (van der Heijden et al., 2018). While differences in classification performance between vehicles can be described with the gini-index, the underlying question of the individual importance of packets and vehicles remains open. The similarity between C-ITS and cyber physical systems (CPS) has been noticed and discussed before in the context of misbehavior detection (van der Heijden et al., 2016). However, the discussion did not consider the physical part and mostly separated both domains. Application behavior metrics have been used to analyze the physical impact of attacks for specific applications like cooperative adaptive cruise control (CACC) (van der Heijden, 2018), but have not been explored further due to concerns about dependencies on specific implementations (van der Heijden and Kargl, 2017). This concept is related to our approach, and we generalize its application and systematically show why these metrics are crucial for the assessment of MDS.

3 A NEW SECURITY MODEL FOR C-ITS

In the literature, several attacker models are used and attackers are usually described by their intention and their capabilities. The intentions and capabilities of possible attackers differ greatly, but a single attacker can perform multiple attacks according to his capabilities. While the individual misbehavior detection mechanisms mostly focus on detecting specific attacks, the discussion about evaluation metrics for misbehavior detection tries to find a generic set of metrics that are independent of the mechanism but also the attacker model. One property common to most (but not all) attackers is that they send packets, and this is what recent evaluation metrics are based on: the classification of incoming packets into legitimate and malicious. Metrics based on the confusion matrix of the classification have to assume the quality of a detection mechanism is related to the number of correct classifications in some form and that mechanisms with a better classification are better at detecting attacks.

We argue that this approach is too broad and while such metrics are applicable to most relevant attacks and detection mechanisms, the underlying assumption does not hold in many cases and therefore the effectiveness of mechanisms is not sufficiently reflected. This is substantiated when thinking about the success of attacks: an attack is successful if the attacker's goal is achieved. The hypothesis that the

degree of which the attacker's goal is achieved is strongly correlated to a function of the classification of received packets by other participants is at least not trivially justified. We call this hypothesis H_{cm} . On the contrary, we already provided some indications of why H_{cm} does not hold in practice in Section 1. In the remainder of this section, we present a new model to discern attackers for which H_{cm} holds and attackers for which it does not.

3.1 Limitations of the Confusion Matrix

Confusion-matrix based metrics such as the FPR and FNR have been used to evaluate network intrusion detection systems (IDS). Traditional computer networks and C-ITS share a number of attack vectors, i.e. attackers might try to gain code execution privileges or crash systems by sending packets that exploit bugs in the parsing code. They might also try to exfiltrate information or deny operation to the network itself by attacking flaws in application logic or network infrastructure. We refer to these attackers as *network attackers*. Because traditional computer networks involve many different applications running on general-purpose computers communicating using many different protocols, assuming specific intentions or goals of attackers is unhelpful and therefore not included in the attacker model. In classic networks based on the internet protocol stack, a one-to-one-communication is possible, meaning the attacker only communicates directly with his targets. In this setting, the relation between a received network packet (or a series of received network packets) and an attack is sufficiently close and IDS performing better in the confusion matrix are therefore likely to be better at detecting attacks. For network attackers, we can assume H_{cm} .

In C-ITS, network communication has a well-defined function. Network packets are broadcast over geographic areas and use a very small set of publicly specified protocols to distribute information about the physical world. In contrast to classical networks, the meaning of each packet is known. The implementation of involved applications may be proprietary, but their purpose is well-defined and as such assumptions about the effect of received packets can be made. In this setting, we face a different type of attacker that we call a *physical attacker*. The physical attacker is characterized by his intention of altering the state of the world in some aspect. To reach his goal, he tries to convince other vehicles to behave in a certain way by sending packets that conform to the specification, but may contain false information. We argue that H_{cm} does not hold for the physical attacker for two reasons.

1. The broadcast of packets in C-ITS implies that irrelevant participants will receive the packet. These participants do not contribute to the realization of the attacker's intention and may not even act on the new information they receive, because it only affects the intended target of the attacker. Even though classification performance for those vehicles is irrelevant for both the recipient and the attacker, it contributes to the confusion matrix.
2. When the same packet with false information is received by a target of the attacker, the resulting impact can be different. Suppose an attacker tries to provoke an accident by faking an end-of-queue warning. Two vehicles erroneously classify the warning as legitimate, where one of them is near the alleged end-of-queue situation and the other is not. The vehicle nearby the fake end-of-queue will be more likely to trigger a dangerous emergency break maneuver than the vehicle further away. Both packet reception events have the same influence on the confusion matrix and both belong to the same attack.

There is no obvious relation between the success of an attack and the number packets received by other participants. When considering physical attackers, we require a new set of metrics that do not depend on H_{cm} and instead reflect the ability of a MDS to mitigate an attack.

3.2 Embracing the Physical World

Metrics based on the confusion-matrix still have a valid application in C-ITS, since classical network attackers can still be a threat. The main application domain of MDS however is the detection of attacks from physical attackers and the discussion of MDS often considers network attackers out-of-scope.

The attacker's intent and the undesirable consequences of the attack are both situated in the physical world. We therefore reiterate on the description of C-ITS as cyber physical system. C-ITS satisfy the definition of a CPS in that vehicles are networked software systems that control a physical process. We detail our model in the next section and refer to the set of applications as software systems and the physical world as the physical process in the remainder of this section. The idea of performing misbehavior detection over intrusion detection is rooted in the realization that C-ITS are CPS. An important property of CPS is that the physical process and the networked software are interdependent and cannot be modeled separately. Many detection schemes indeed exploit the properties of the physical world to identify misbehavior on the network layer. To build new metrics

without assuming H_{cm} , we propose to equally consider the properties of the physical world when assessing the effectiveness of MDS. We identify two problems that need to be solved.

1. The physical process is complex. Finding the right abstraction is difficult and problem-specific. One contribution evaluated MDS by analyzing a specific application and comparing the effects of one selected attack (van der Heijden et al., 2016). While this leads to very useful conclusions, the comparability between mechanisms is limited when varying the attack.
2. No evaluation framework is available. VeReMi has provided a common dataset for non-interactive evaluations, but cannot be used to analyze effects that MDS have on the physical world.

In the following section, we address the first issue by suggesting a metric for the *selfish attacker* proposed in (Samara et al., 2010; van der Heijden et al., 2016). The selfish attacker is a physical attacker with the goal of gaining an advantage on the road. By focusing on the attacker's goal, we can provide a metric that is applicable to all attacks and MDS. We address the second issue with our simulation framework that we present in Section 5.

4 A METRIC FOR SELFISH ATTACKERS

The selfish attacker adheres to specified message formats, but does not necessarily follow all protocol specifications, e.g. he can send messages more often than allowed or suppress packets he should transmit or forward. We deem this attacker type highly relevant, because most traffic participants have a motivation for this kind of attack and it seems plausible that such an attack could be performed by individuals with little costs, e.g. using methods of modifying the firmware of certain vehicle models. In the scope of this work, we do not consider *cooperative* attackers, i.e. we assume that the attacker only controls a single vehicle and does not cooperate with others when performing his attacks. We assume that the attacker's goal is to reach his destination as fast as possible.

We previously discussed the limitations of current metrics based on the confusion matrix and further note there is another fundamental shortcoming of such metrics in the context of selfish attackers: In real-world traffic scenarios, maliciousness is not a property of a message and cannot be captured by a metric based on the confusion matrix of individual packet classifications. Measuring FPRs and FNRs

thus is a difficult task by itself, because there is no inherent definition of packet maliciousness. In some cases, the decision is obvious: if a vehicle sends a warning for an accident that did not happen, the information is clearly incorrect and the packet should be treated as malicious. But this is not a function of just the packet itself: the correctness depends on the state of the world at this point in time, i.e. whether an accident happened or not. Apart from this, the classification is gradual, especially when considering sensor noise and inaccuracies. A message containing a vehicle's location will always have some offset due to GNSS and integration errors. There is no natural distance at which the message is clearly incorrect and should be classified as malicious. A workaround used in some evaluations is to define attacker vehicles and assume all packets sent by them are malicious and should be classified as incorrect. We note that in the case of selfish attackers, the maliciousness of a packet also depends on the intention behind the transmission of the packet, not on the packet's content, which gives further weight to our argument that metrics should be closely coupled with the attackers intention.

4.1 Modeling the Cyber Physical System

In light of these observations, we propose a CPS-based MDS-metric for selfish attackers. The set of applications running inside all vehicles forms the software system interacting with the physical process. We view the traffic flow itself as the physical process and argue that this is what needs to be protected in the presence of selfish attackers: an attacker that is unable to change the traffic flow will not gain an advantage over the same scenario without the attack.

The MDS is part of the software system, i.e. the set of components that form driving decisions inside the individual vehicles, and therefore also influences the traffic. This model can take most influences into account that are part of the real traffic system, including other components running inside the vehicle's software system.

4.2 Attacker's Advantage

In order to measure the influence of the MDS on the traffic system, we first formulate a metric for the degree to which an attacker achieves his goal. We assume that the attacker controls a single vehicle with a start position p_s and a destination p_d . The vehicle passes the driving distance s_a between p_s and p_d in time t_a . To measure the advantage the attacker achieves with his attack, we measure the time t_a and

the time $t_{a,\text{fair}}$, which is the driving time of the same vehicle when behaving like a legitimate network participant, i.e. not performing any attacks. We call this the attacker's advantage

$$q_a = 1 - \frac{t_a}{t_{a,\text{fair}}}. \quad (1)$$

In our model, this is purely a property of the physical process, i.e. an effect that the attackers' communication has on the physical world. q_a can admit negative values if the attack leads to a longer travel time for the attacker.

We now propose to base the quality assessment of MDS on q_a : an optimal detection mechanism achieves $q_a \leq 0$, i.e. the attacker is unable to gain an advantage with his attack. In realistic scenarios, this is not always possible, for example if the detection scheme is not running on all vehicles. It is thus desirable to obtain a lower bound to which a specific scheme can be compared in order to assess its quality. We later show how a lower bound for q_a can be estimated.

Because q_a is a property of the physical process, it can be measured independently of the performed attacks or detection mechanisms. To measure q_a , only two conditions are necessary:

1. the physical process must be observable, i.e. t_a can be determined, and
2. the scenario must be repeatable

Condition 2 is necessary to measure $t_{a,\text{fair}}$ under the exact same conditions but without the attack. Since we see the MDS as a distributed system running inside the software system, evaluation decisions such as aggregation methods are not needed for determining q_a . The specific deployment parameters of the mechanism, such as the penetration rate, must however be specified and are seen as a parameter of the mechanism to be analyzed.

While t_a can be measured universally, it does depend on all components that the physical process itself depends on, especially

- the traffic scenario, and
- the implementation of vehicles' applications influencing the driving decisions.

When comparing different MDS, these influences should be minimized, e.g. the measurements should be taken in the same traffic scenario and with the same implementations. We expect that q_a is a useful metric to assess detection performance, even when comparing mechanisms with different traffic scenarios and implementations, as the value of q_a specifies what an attacker can achieve in the presence of a MDS.

5 EVALUATION FRAMEWORK

Our metric q_a cannot be determined using precomputed datasets such as VeReMi, since measuring q_a depends on observing the attack's impact on the physical process. The simulation of interactions between vehicles' software and the traffic system requires that the simulator allows coupling a road traffic simulation and the networking and software simulation. This is usually called an *interactive* simulation. This requirement is necessary to allow the misbehavior detection mechanism to impact the physical process, i.e. the traffic simulation. Our framework uses the artery simulation module (Riebl et al., 2019). We use this basis to provide an extensible framework for the analysis of misbehavior detection mechanisms. Our framework can be used to analyze attacks and detection mechanisms for several attacker types, but we limit our description in this work to the analysis of detection of selfish attacks and the measurement of q_a . We extend artery to facilitate the simulative analysis of misbehavior detection mechanisms and attack scenarios.

Artery is an interactive simulation framework, originally developed for the testing of applications in the European ITS-G5 vehicular communications protocol stack. Artery is implemented as package for the OMNeT++ Network Simulator (Varga, 2001) and couples the networking simulation modules with the SUMO road traffic simulator (Lopez et al., 2018). The communication stack is implemented by the vanetza project (Riebl et al., 2017), a standalone open source implementation of the ITS-G5 standards.

5.1 Detection Model

To measure the impact of MDS on the physical process, we assume a data-centric misbehavior detection model, i.e. the MDS acts on individual packets. Each received packet is passed to the misbehavior detection algorithm and classified. Our framework, by default, discards all packets that are classified as malicious. Other packets are passed up in the stack and handled by the application. Figure 1 shows the processing of received packets inside a vehicle's simulation module. We use this simple model to decouple the misbehavior detection from the application implementation. Detection mechanisms can be implemented independently of application behavior. This approach removes flexibility from the application but allows to keep the implementation minimal and does not incur unrealistic limitations in our view.

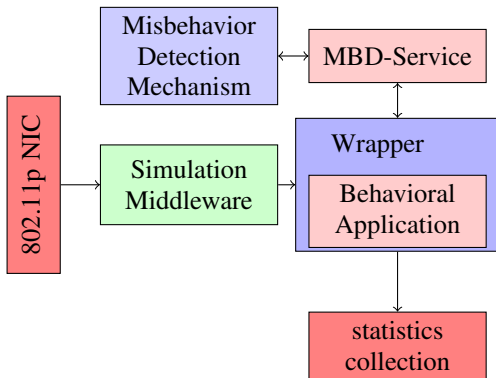


Figure 1: Overview over the processing of messages in a simulated vehicle.

5.2 Simulation Scenarios

A simulation scenario $s = (s_t, B, A_{\text{mbds}}, p_{\text{mbds}}, A, R)$ in our framework is defined by

1. a traffic scenario s_t ,
2. the set of behavioral applications B defining the vehicles' driving decisions,
3. a misbehavior detection mechanism A_{mbds} ,
4. the penetration rate $0 \leq p_{\text{mbds}} \leq 1$ defining the share of vehicles equipped with a MDS,
5. the attacker definition A and
6. the set of random seeds $R = \{r_1, \dots, r_n\}$

In order to analyze a detection mechanism, A_{mbds} and A must be implemented in our framework. We designed the interfaces with a focus on extensibility to make analysis feasible for most data-centric mechanisms and expect that most implementations can be integrated without much effort. The parameter p_{mbds} is likely to be varied as part of the analysis.

The VeReMi-dataset provides a substantial benefit for the academic discussion by providing a common reference against which new detection mechanisms can be evaluated. We see the need of comparing similar detection methods directly and likewise aim to provide a shared evaluation platform that is useful for future research. To maximize comparability of q_a measurements, we hope to find a common fixed set of parameters s_t and R that can be used across publications for many mechanisms.

5.3 Traffic Scenarios

We selected two traffic scenarios *highway* and *urban* to facilitate an effective evaluation. The design of traffic scenarios is not trivial, as it represents a trade-off between execution time, universality and attacker-



Figure 2: Highway segment scenario.

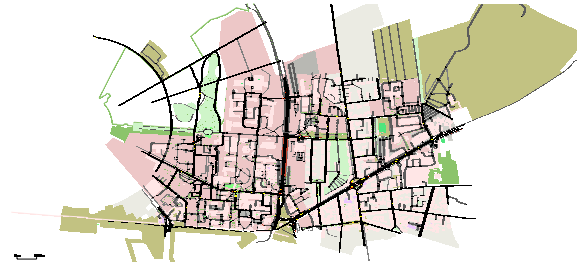


Figure 3: Urban scenario.

specificity. A large-scale scenario such as the LuST-Simulation (Codeca et al., 2015) used by VeReMi contains numerous traffic situations and is suited for many attacks, but is very expensive in terms of run time costs. For this reason, we built two separate traffic scenarios, a simple and artificial highway segment and a medium-scale urban road network in the city of Karlsruhe, Germany. Figures 2 and 3 show the road network of both scenarios.

The *highway* scenario simple and contains few junctions. Traffic flows from left to right and all vehicles try to reach the last segment of the main road. The main road is not speed-limited, other road segments are limited to $13,89\text{ms}^{-1}$. Without external interference, no vehicle chooses to leave the main road. The traffic is composed of mostly passenger cars and busses with a maximum speed of 20ms^{-1} but also includes 2.6% sports cars with a maximum speed of 20ms^{-1} . The attackers maximum speed is set to 100ms^{-1} to allow him to create an advantage out of a beneficial traffic situation. The total amount of vehicles that are driving simultaneously varies between 112 and 117.

The *urban* scenario is more complex and models a realistic urban traffic system with traffic lights and many junctions. Vehicles start at a random starting position and follow a randomly selected route. In this traffic scenario, speed is limited and the traffic is dense. Without performing an attack, the route takes 38 minutes on average. Maximum speeds of vehicles do not influence the driving time.

5.4 Behavioral Applications

The physical process measured by our approach is highly dependent on the driving decisions made by each individual traffic participant. Many attacks aim to influence the driving decisions of other participants by sending misleading messages.

For our analysis, we keep the assumptions about the implementation minimal. We assume all non-attacker participants act rationally and always take the route that appears to be the fastest towards their destination. Once the application receives new information about the world, it recalculates the path with the shortest travel time to the destination and changes the route if the new path is shorter than the current route.

5.5 Random Seeds

Individual simulation runs in our framework are deterministic and reproducible, if the implementations of A_{mbds} and A support it. All randomized elements in the scenario, such as the selection of vehicles equipped with a misbehavior detection mechanism according to p_{mbds} , depend only on a single seed value. In order to obtain a stable value for q_a , multiple simulation runs are executed with different seed values. In this case, t_a^i and $t_{a, \text{fair}}^i$ are measured for each simulation run i and q_a can be calculated as

$$q_a = 1 - \frac{1}{N} \sum_{i=0}^{N-1} \frac{t_a^i}{t_{a, \text{fair}}^i}, \quad (2)$$

where N is the total number of simulation runs. For large N , the influence of the choice of $R = \{r_i\}_{0 \leq i < N}$ diminishes. Individual traffic situations can however change substantially with r_i and therefore q_a can be different for deviating seed values.

6 EVALUATION OF EXISTING SCHEMES

We analyze a basic variant of the detection mechanism proposed by Petit, Feiri and Kargl (Petit et al., 2011) using our evaluation metric for different penetration rates. The implemented scheme tries to create a consensus about dangerous events reported by other vehicles. As long as the vehicle is not forced to make a decision, the mechanism waits for further messages sent by other neighbors about this event. Events are only classified as legitimate if the number of warnings sent by different senders exceeds a dynamic threshold.

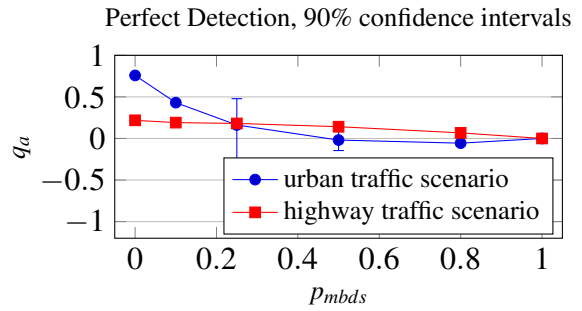


Figure 4: Estimation of a lower bound for q_a for both traffic scenarios.

6.1 Attacker Implementation

We implement a simple fake message injection attack A_{mi} . The attacker's vehicle sends warnings for nonexistent accidents on the attacker's route every four seconds to convince other vehicles of avoiding the route and reduce traffic on his path. The attacker's vehicle follows a fixed route defined by the scenario. While this is a very simple strategy, our results show that it is very effective if no misbehavior detection is performed.

6.2 Simulation Parameters

The attack and detection mechanism are usable in the urban traffic scenario as well as in the highway traffic scenario. For our evaluation, we use both traffic scenarios. We use the implementation of behavioral applications as described in Section 5.4. We define a total of twelve simulation scenarios by using six different values for p_{mbds} in both traffic scenarios. We use $R = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ in all scenarios, which results in a total of 100 simulation runs.

6.3 A Lower Bound for q_a

Our framework can be used to estimate the limits of what misbehavior detection can achieve. We implemented a detection mechanism A_{perfect} that has access to the internal attacker state and always classifies packets correctly.

Figure 4 shows the resulting q_a for A_{perfect} with different penetration rates. We see that this attack is very successful in the urban traffic scenario, as the attacker is able to achieve a high q_a value only if no detection mechanism is used. We conclude that even low penetration rates of effective misbehavior detection systems can reduce the selfish attacker's advantage substantially. For $p_{mbds} \geq 0.5$, the attacker gains no noticeable advantage by performing the attack. In some simulation runs, we measure a $q_a < 0$, which

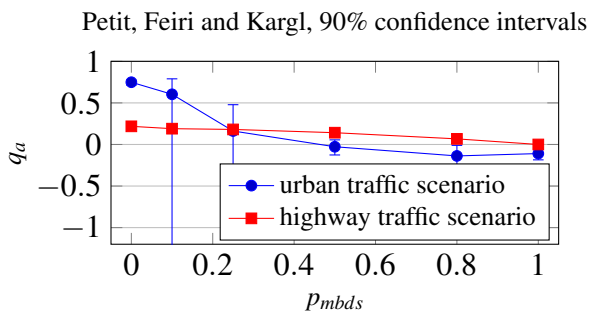


Figure 5: Evaluation result for the mechanism of Petit, Feiri and Kargl in both scenarios.

means the attack created a disadvantage for the attacker. This is not caused by the detection of the attack, but instead by the simplicity of the attack. The false warnings sent by the attacker lead to an increase in travel time in some constellations.

In the highway traffic scenario, the attack is less effective for lower penetration rates, but the attacker still achieves some advantage with higher penetration rates of misbehavior detection.

6.4 Results

We evaluated the scenarios with the mechanism from Petit, Feiri and Kargl and show the results below. Figure 5 shows the values of q_a for both traffic scenarios with different penetration rates. The measurements show that the mechanism can effectively reduce the attacker's advantage. In the highway traffic scenario, we note that the mechanism's q_a values are very close to the perfect detection mechanism.

The execution times of individual simulation runs depend on the traffic scenario and the penetration rate p_{mbds} . We executed the simulations on 8 cores with a 2,6 GHz clock rate and 16 GB of RAM. Eight simulation runs were executed in parallel. Simulation runs of the highway scenario were finished in under 6 minutes each, while the complex urban traffic scenario took between 1.5 hours and over ten hours to complete. Table 1 lists the execution times for some scenarios. The execution times were measured using the perfect detection mechanism. We suspect that the variations in the urban scenario are caused by the reduced number of invocations of the traffic simulation for larger p_{mbds} . Our measurements surprisingly show that even removing a small percentage of vehicles from the attacker's influence can have a substantial impact on his advantage in some situations.

Table 1: Results and Performance.

		$s_t = \text{highway}$	
p_{mbds}	q_a	Execution Time (s)	
0.00	0.75	356	
0.25	0.16	354	
0.80	0.07	361	
1.00	0.00	351	
		$s_t = \text{urban}$	
p_{mbds}	q_a	Execution Time (s)	
0.00	0.22	37 232	
0.25	0.18	22 220	
0.80	-0.13	9877	
1.00	-0.11	5501	

7 CONCLUSION

In this work, we propose a new metric for assessing the quality of misbehavior detection mechanisms in the presence of selfish attackers. To improve over the state-of-the-art metrics based on the confusion matrix of packet classifications, we base our metric on measuring the attacker's success on achieving his goal.

We further present a simulation framework to measure q_a and propose two scenarios for testing attacks and detection mechanisms. We analyze a simplified variant of the detection mechanism proposed by Petit, Feiri and Kargl in this framework and estimate a lower bound for q_a by using a perfect detection mechanism. We discuss run-time performance and conclude that our metric can be used to evaluate practical detection mechanisms in realistic traffic scenarios.

Our metric q_a allows a reliable assessment of MDS and gives a realistic indication of the effectiveness of an MDS, which is an improvement over metrics based on the confusion matrix, such as the precision-recall-graph. Our proposal is able to assess MDS in the presence of selfish attackers. The first application of our framework showed promising results and we hope to extend this analysis to other detection schemes and more sophisticated attacks. We show that metrics based on measuring the physical process can be used to successfully evaluate MDS. We observed that the interpretation of q_a could benefit from additional traffic measurements to better assess negative effects that the misbehavior detection has on desired effects.

ACKNOWLEDGEMENTS

This work was supported by the Competence Center for Applied Security Technology (KASTEL).

REFERENCES

- Codeca, L., Frank, R., and Engel, T. (2015). Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research. In *2015 IEEE Vehicular Networking Conference (VNC)*, pages 1–8.
- Kamel, J., Wolf, M., van der Hei, R. W., Kaiser, A., Urien, P., and Kargl, F. (2020). Veremi extension: A dataset for comparable evaluation of misbehavior detection in vanets. In *IEEE International Conference on Communications (ICC)*, pages 1–6.
- Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., and Wießner, E. (2018). Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*, pages 2575–2582. IEEE.
- Petit, J., Feiri, M., and Kargl, F. (2011). Spoofed data detection in vanets using dynamic thresholds.
- Riebl, R., Obermaier, C., and Günther, H.-J. (2019). Artery: Large scale simulation environment for its applications. In *Recent Advances in Network Simulation*, pages 365–406. Springer.
- Riebl, R., Obermaier, C., Neumeier, S., and Facchi, C. (2017). Vanetza: Boosting research on inter-vehicle communication. In *Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC 2017)*, pages 37–40.
- Samara, G., Al-Salihy, W. A., and Sures, R. (2010). Security issues and challenges of vehicular ad hoc networks (vanet). In *4th International Conference on New Trends in Information Science and Service Science*, pages 393–398. IEEE.
- Sommer, C., German, R., and Dressler, F. (2011). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis. In *IEEE Transactions on Mobile Computing (TMC)*, volume 10, pages 3–15. IEEE.
- van der Heijden, R. W. (2018). *Misbehavior Detection in Cooperative Intelligent Transport Systems*. PhD thesis, Ulm, Germany.
- van der Heijden, R. W., Dietzel, S., Leinmüller, T., and Kargl, F. (2016). Survey on misbehavior detection in cooperative intelligent transportation systems.
- van der Heijden, R. W., Dietzel, S., Leinmüller, T., and Kargl, F. (2019). Survey on misbehavior detection in cooperative intelligent transportation systems. In *IEEE Communications Surveys & Tutorials*, volume 21, pages 779–811.
- van der Heijden, R. W. and Kargl, F. (2017). Evaluating misbehavior detection for vehicular networks. In *5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication*, page 5.
- van der Heijden, R. W., Lukaseder, T., and Kargl, F. (2018). Veremi: A dataset for comparable evaluation of misbehavior detection in vanets. In Beyah, R., Chang, B., Li, Y., and Zhu, S., editors, *Security and Privacy in Communication Networks*, pages 318–337, Cham. Springer International Publishing.
- Varga, A. (2001). Discrete event simulation system. In *Proceedings of the European Simulation Multiconference (ESM'2001)*, pages 1–7.