Physical Layer Plausibility Checks for Misbehavior Detection in V2X Networks

Steven So
Boston University
Boston, Massachusetts, USA
sbso@bu.edu

Jonathan Petit
OnBoard Security
Wilmington, Massachusetts, USA
jpetit@onboardsecurity.com

David Starobinski
Boston University
Boston, Massachusetts, USA
staro@bu.edu

ABSTRACT

Location spoofing is a proven and powerful attack against Vehicle-to-everything (V2X) communication systems that can cause traffic congestion and other safety hazards. Recent work also demonstrates practical spoofing attacks that can circumvent application layer sanity checks. In this paper, we propose three novel physical layer plausibility checks that leverage the received signal strength indicator (RSSI) of basic safety messages (BSMs). These plausibility checks have multi-step mechanisms to improve not only the detection rate, but also to decrease false positives. These checks can be run independently by each vehicle and do not rely on the assumption that the majority of vehicles is honest. We comprehensively evaluate the performance of these plausibility checks using the VeReMi dataset (which we enhance along the way) for several types of attacks. We show that the best performing physical layer plausibility check among the three considered achieves an overall detection rate of 83.73% and a precision of 95.91%, far outperforming recently proposed machine learning-based misbehavior detection methods operating at the application layer.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; **Network security**; • **Networks** → *Cyber-physical networks*;

KEYWORDS

Vehicular Network, Connected Vehicles, Physical Layer, RSSI, Misbehavior Detection, Plausibility Check, Security

ACM Reference Format:
Steven So, Jonathan Petit, and David Starobinski. 2019. Physical Layer Plausibility Checks for, Misbehavior Detection in V2X Networks. In *Proceedings of ACM WiSec Conference (WiSec'19).* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/xx.xxx/xxx_x

1 INTRODUCTION

Vehicle-to-everything (V2X) aims to improve traffic safety and efficiency through timely over-the-air exchange of information between vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I). V2X effectively increases the operator's and vehicle's line-of-sight, creating a safer environment. In V2X, vehicles communicate using Basic Safety Messages (BSMs) that are defined in the SAE J2735 standard. A BSM contains situation data, such as the vehicle's location, speed, acceleration, heading, and brake status.

Yet, concurrently to enhancing safety, V2X also raises security risks. Specifically, attackers may exploit this system of communication to broadcast bogus BSMs containing fake vehicle location or speed in order to create ghost vehicles or imaginary traffic jams.

Data authenticity in V2X is ensured by authentication protocols that are defined in IEEE 1609.2 [9]. The problem is that even if the data is authentic, its correctness is not guaranteed. Indeed, attacks have already been simulated and practically demonstrated against current connected vehicle applications, such as Intelligent Traffic Signal Systems (I-SIG) [1, 5].

There exists, therefore, also a need to ensure the correctness of the data (i.e., data-centric trust [20]), which can be achieved through plausibility checks. This approach resembles that of intrusion detection systems used in computer networks, where detection algorithms monitor the network for unusual or suspicious activity. Likewise, a detection system for V2X should be capable of locally detecting misbehavior and ensuring revocation of credentials in a timely manner [13].

A possible approach for implementing misbehavior detection is to run plausibility checks at the application layer (namely, by checking the plausibility of the contents of the BSMs). Recent work [14] proposes machine learning algorithms to this effect, and the performance of the algorithms are verified against the VeReMi dataset [18]. This dataset, simulated and built on top of the VEINS simulator [15], contains labeled data consisting of well-behaved vehicles along with five different type of position forging attacks (cf. Section 2.2). The work in [14] shows that, overall, a K-Nearest Neighbors (KNN) algorithm yields a detection rate of 61% while a Support Vector Machines (SVM) algorithm achieves a detection rate 65%. In particular, it turns out that these algorithms fail in effectively detecting certain position forging attacks that mimic the movements of a real vehicle. The work of [5] exploits this weakness to mount a practical attack against a real Intelligent Traffic Signal System (I-SIG) for connected vehicles.

These limitations of application layers plausibility checks call for the design of additional, complementary approaches. In this work, we propose, design, and evaluate new plausibility checks that operate at the *physical layer*. While there already exist misbehavior detection algorithms based on physical layer properties such as Angle-of-Arrival (AoA), Direction-of-Arrival (DoA), Doppler-Shift, Time-Difference-of-Arrival (TDoA), and Received Signal Strength Indicator (RSSI) [6, 21, 23], most of these approaches attempt to defend against a specific type of attacks (e.g., a Sybil attack). In contrast, our aim is to design algorithms whose robustness are

WiSec'19, May 15-17, 2019, Miami, FL

ACM ISBN 978-1-4503-5933-7/19/04.
https://doi.org/xx.xxx/xxx_x

tested against a range of possible attacks, including the attack used in [5].

Our approach is based on collecting the RSSI value obtained from each individual BSM and on using this data to determine whether a received BSM is malicious. We propose three plausibility checks to classify the transmitting vehicle as either malicious or normal. These plausibility checks are evaluated against the VeReMi dataset.

The contributions of this paper are five-fold:

(1) We augment the VeReMi dataset with new information (i.e., the location of the receiving vehicles for each BSM), which is necessary for performing physical layer analysis.
(2) We qualitatively validate the RSSI data produced by VeReMi by performing a comparison with actual RSSI data obtained under real-world conditions.
(3) We statistically characterize the RSSI behavior of five different position-forging attacks.
(4) We introduce a physical layer misbehavior detection system in the form of three plausibility checks First-BSM (FBSM), Majority-BSM (MBSM), and Weighted-BSM (WBSM) that are used to detect misbehavior.
(5) We perform an in-depth evaluation of the performance of these three plausibility checks and compare their performance with the application layer plausibility checks of [14]. We show that, overall, FBSM and WBSM achieve much higher detection rates (recall) and precision.

The paper is organized as follows. Section 2 covers the related work of misbehavior detection in V2X and datasets. Section 3 describes the experimental setup. Section 4 introduces our RSSI-based plausibility checks. Section 5 explains our experiments before Section 6 presents our results. Section 7 discusses current limitations of our system. Section 8 concludes the paper.

2 RELATED WORK

2.1 V2X and Misbehavior Detection

A complete V2X system must implement a security mechanism to ensure that the data broadcast is trustworthy. Data-centric misbehavior detection is an important component of such a security mechanism that evaluates the correctness of data contained in BSMs. Several prior papers on data-centric misbehavior detection either require consensus [10] or a central authority [2]. In particular, [2] proposes a scalable central misbehavior evaluation system for On-Board Units (OBUs) and roadside units (RSUs) to detect attackers from within the network. The approach requires that every node within the network forward incident reports to a central authority, which takes the final decision [2]. A major constraint of this method lies in the assumption that the majority of nodes are honest. The plausibility checks discussed in our paper can be run independently by each vehicle and do not rely on the above assumption.

The recent work of [14] proposes and evaluates application layer plausibility checks that relax the assumption that the majority of nodes must be honest. The outputs of the plausibility checks are used in the feature vectors of machine learning algorithms such as K-Nearest Neighbors (KNN) and Support Vector Machines (SVM) to perform misbehavior detection. Nevertheless, the methods of [14] perform poorly in detecting position forging attacks that mimic the movements of a real vehicle. These attacks are indeed difficult to detect at the application layer since an attacking vehicle may be behaving normally based on the BSM contents. In this paper, we show that these types of attacks are more easily detectable using physical layer properties.

The work of [5] exploits the aforementioned vulnerability of application layer plausibility checks to launch a traffic congestion attack against a real Intelligent Traffic Signal System (I-SIG). An I-SIG uses BSMs broadcast by connected vehicles to actuate traffic signals at an intersection and reduce congestion. The work of [5] shows how to spoof the BSM payload in order to imitate a legitimate vehicle, meaning that BSM fields such as acceleration, velocity, and position are all cross-validated and appear to be valid. The attack can impact traffic control signals at intersections, costing drivers several order of magnitude higher travel time than normal.

In [17], a physical layer method is proposed to detect misbehaving vehicles, using Doppler speed and angle-of-arrivals. The method relies on a trust scoring mechanism which is implemented by surveying other vehicles in the network. The system applies Kalman filtering and chi-squared tests to the Doppler speed and the angle-of-arrival data to determine the most trustworthy vehicle within a vehicle's perimeter. The system then uses that vehicle as a reference to test the trustworthiness of other vehicles. This method has not been tested against various position forging attacks, such as the ones later discussed in VeReMi. Another issue is that this system does not perform well under high density traffic conditions where vehicles move slowly, since the Doppler speed of different vehicles is difficult to distinguish. The RSSI metric has the advantage of not depending on the vehicle's speed.

To the best of our knowledge, [22] is the only work that uses the RSSI of BSMs for misbehavior detection, and this for detecting Sybil attacks. A Sybil attacker generates several fake identities with false messages, severely impairing the functions of safety-related protocols. The method of [22] exploits statistical properties of the RSSI of surrounding vehicles to detect Sybil attacks. Vehicles are classified as part of a Sybil attack if their RSSI values are sufficiently similar to those of other vehicles, implying that the BSMs are being emitted from the same source. The limitation of this method is that it does not take into account a malicious vehicle that does not transmit its own location but generates a singular ghost vehicle at a different location. Hence, this method can be exploited by other position forging attacks, such as those contained in VeReMi.

2.2 Data Sets

There exist several datasets that measure RSSI behavior in vehicular environments. For instance, [8] conducts a field testing campaign as part of the iTETRIS European research project, assessing the impact of urban environments and different RSU deployment conditions, to test the quality of IEEE 802.11p Vehicle to Infrastructure (V2I) communications [8]. In this dataset, 22 different RSUs broadcast messages to a vehicle moving in an urban environment. The locations of the RSUs cover different scenarios such as trees and vegetation, bridges and elevation, and Non-Line-of-Sight (NLOS) conditions. The dataset contains the RSSI of the received BSMs for different positions of the vehicle. However, this dataset only includes normal behavior, and does not include attacker data. There exist other current real-time datasets, but they also do not appear

Table 1: VeReMi Attack Types Description

ID:Attack	Detail	Parameters
1: Constant	Attacker transmits a fixed location	x = 5560, y = 5820
2: Constant Offset	Attacker transmits a fixed, offset added to the real position	Δ x = 250, Δ y = -150
4: Random	Attacker sends a random position inside the simulation area	uniformly random in playground
8: Random Offset	Attacker sends a random position in a rectangle around the vehicle	Δ x ,Δ y are uniformly random from [-300,300]
16: Eventual Stop	Attacker behaves normally for some time and then attacks by transmitting the same position repeatedly	Stop probability increases by 0.025 each position update

ideal for evaluating the performance of a misbehavior detection system due to the absence of attackers, the inconsistency in BSMs broadcast rate [12], and the presence of a single vehicle [4].

The VeReMi dataset does contain several different types of attacker data with consistent BSM broadcast rate [18]. This open source dataset is attempting to become the dataset of reference for the evaluation and comparison of misbehavior detection algorithms. VeReMi comprises 5 different position forging attacks, 3 vehicle densities (low, medium and high), 3 attacker densities (10%, 20% and 30%), and at every combination of those 3 parameters the simulation is repeated 5 times for randomization. This dataset is built to test misbehavior detection mechanisms in diverse scenarios. The 5 different types of attacks are shown in Table 1. The dataset contains the message logs of the attacking and legitimate (non-malicious) vehicles. This includes reception time stamp, claimed transmission time, claimed sender, unique message ID, GPS position (x, y, z), RSSI value, position noise and speed noise vector for each receiving vehicle in every scenario. A ground truth file stores the true values of the BSM attributes of both attacking and legitimate vehicles. The attacker type attribute in the ground truth file keeps the label of the attack ID as described in Table 1. This was all parsed by [14]. However, the parsed dataset lacks information about the location of receiving vehicles at the time they receive BSMs. This makes it very difficult, if not impossible, to determine the true distance between receiving and transmitting vehicles at the time that a BSM is received.

To reliably trust any results from VEINS, we need to validate that RSSI values correspond to real world measurements. While there exists some prior work on checking the accuracy of the noise and interference models of the VEINS simulator in real world settings [3], we nevertheless conduct our independent study of comparing RSSI behavior in simulation versus real world data.

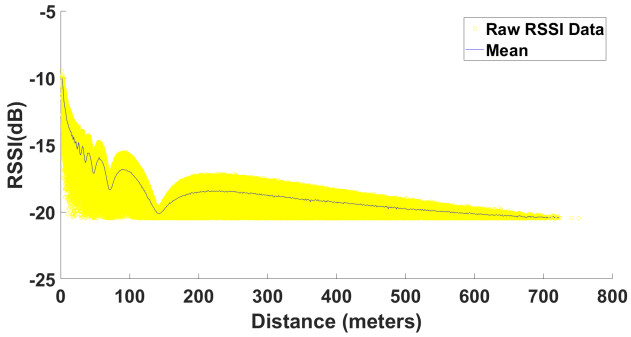

Figure 1: Simulated RSSI behavior according to distance between receiver and transmitter.

3 EXPERIMENTAL SETUP

The first step is to recreate simulations in VeReMi. We change the code in VeReMi to properly record the receiver's location at the time of a received BSMs arrival. Next, we create a script that parses the data and converts the file format from JSON to CSV. The JSON file output of VeReMi's current framework indeed has many objects inside one file formatted in an unorthodox manner. Columns in the CSV file contain information about the vehicle including the label of the vehicle in the last column.

We have published the new generated dataset to facilitate future work on physical layer analysis at [16].

3.1 RSSI Behavior: Simulation vs. Real Data

We first qualitatively validate the RSSI data produced by the simulator. Specifically, we run simulations for high traffic density vehicles and a 30% attacker density. Figure 1 shows the raw simulation BSM data produced using VeReMi. The yellow points represent each individual BSM sent, and the trend line represents the mean value at each distance apart away from the receiving and transmitting vehicle.

We generate Figure 2 from the iTETRIS dataset previously mentioned. We consider 4 different scenarios: (1) direct line-of-sight (LOS) (traces T1 and T2); (2) non-line-of-sight (NLOS) (traces T3 and T4); (3) bridges and elevation (trace T7); and (4) trees and vegetation (traces T8 and T9). Note that we focus on the qualitative behavior of the data, rather than the precise numerical values. As stated above, RSSI values depend on the environment, hence it is important to find out whether the behavior appears consistent for all environments.

From the figures, we observe that (1) the RSSI behavior for real-data does stay consistent across different environments; (2) the RSSI behavior for the simulation data closely resembles the real-world data. These observations justify the use of VeReMi to test the performance of plausibility checks.

3.2 Misbehavior vs. Normal Behavior

To create a misbehavior detection system using RSSI, it is important to understand the RSSI behavior caused by different attacks. In this case, by observing attacks of type 1, 2, 4, 8, and 16 described in Section 2.2, we next show that these attacks exhibit distinct RSSI

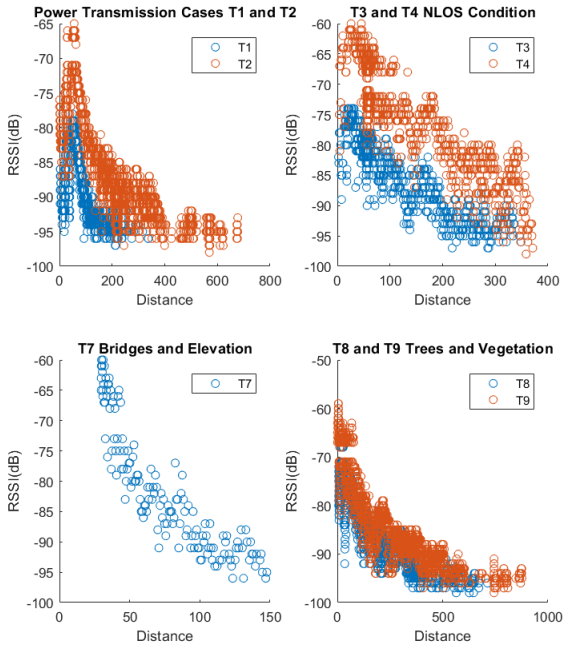

Figure 2: Real RSSI behavior according to distance between receiver and transmitter. Note that traces T_1 and T_2 correspond to line-of-sight, while traces T_3 and T_4 correspond to non-line-of-sight (NLOS).

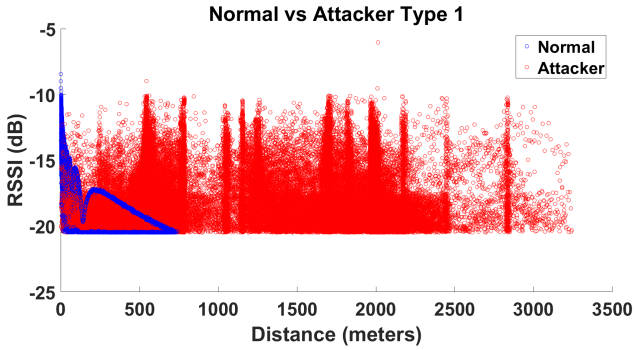

Figure 3: RSSI behavior according to distance between receiver and transmitter for Attack Type 1.

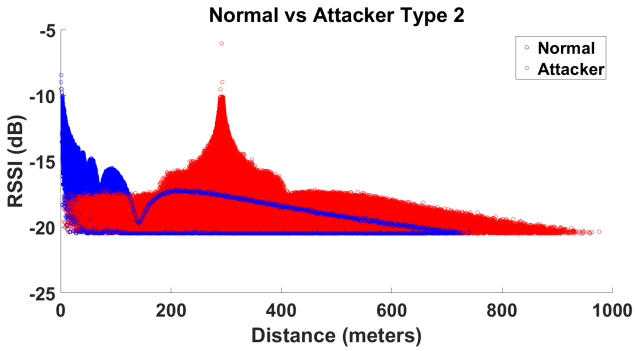

Figure 4: RSSI behavior according to distance between receiver and transmitter for Attack Type 2

vs. distance behavior. Note that for normal behavior, the distance represents the real distance, while for misbehavior, the distance is erroneous due to the spoofed positions advertised by attackers.

In Figures 3 through 7, we compare the RSSI behavior under normal conditions and under attacks. When the results overlap, we only show the maximum and minimum values observed for the normal behavior. Notice the different scales of the x-axis on the different figures. The maximum communication range of vehicles is 800m, hence the normal RSSI behavior does not extend beyond that distance.

Attack Type 1: Constant Position. The first case is when attackers advertise a constant position, while they are actually moving. Figure 3 shows spikes. These spikes are primarily due to stationary receiver vehicles receiving constant position BSMs from attackers. Hence, these stationary receivers receive many messages corresponding to the same (fake) distance. Since the attackers are moving and the real distance is changing, the range of RSSI corresponding to the (fake) fixed distance varies significantly behind normal. Another important observation is that the real communication range for normal vehicles is at most 800m, but many fake distances induced by the BSMs of constant position attackers far exceed that communication range.

Attack Type 2: Constant Offset. The RSSI vs. distance curve is shown in Figure 4. We note that this curve is symmetrical. The right half of the attacker RSSI vs. distance curve seems to have shifted in distance in comparison to the normal curve, and this is because the constant offset makes a misbehaving vehicle appear farther away than its true location. The left half of the attacker RSSI vs. distance curve corresponds to cases where the constant offset makes a misbehaving vehicle appear closer to the receiving vehicle.

Attack Type 4: Random Position. Attack type 4's behavior is shown in Figure 5. Since attack type 4 is a random position attack, we can see that the RSSI vs. distance curve is random, similar to a uniform distribution.

Attack Type 8: Random Offset. This attack has the behavior most similar to normal. Attack type 8 is a random offset attack, which means that sometimes the offset can be very small. Therefore, some of the RSSI data will behave normally. Nevertheless. similar to the constant offset (attack type 2), there is still a pattern which resembles a shifted normal RSSI vs. distance curve.

Attack Type 16: Eventual Stop. The behavior of attack type 16 is shown in Figure 7. Not surprisingly, this sudden stop attack closely resembles the constant position attack (Attack type 1) since the sudden stop attack turns into a constant position attack.

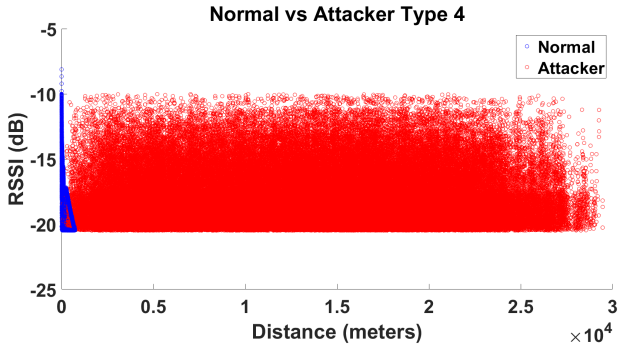

Figure 5: RSSI behavior according to distance between receiver and transmitter for Attack Type 4

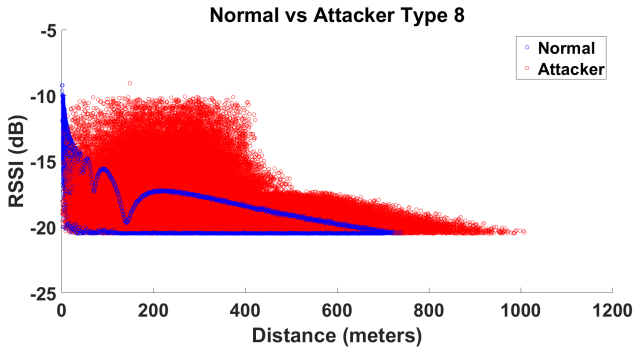

Figure 6: RSSI behavior according to distance between receiver and transmitter for Attack Type 8

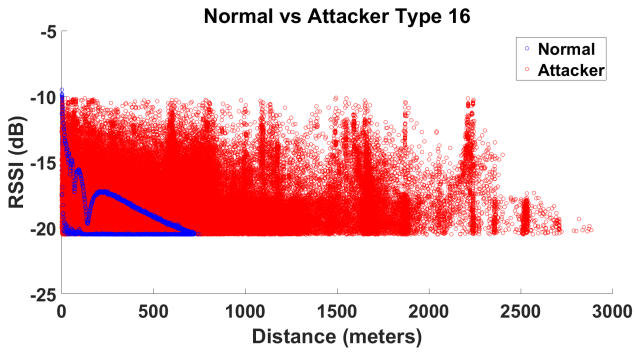

Figure 7: RSSI behavior according to distance between receiver and transmitter for Attack Type 16

4 PLAUSIBILITY CHECKS

4.1 Identifying Anomalous BSMs

The framework for the vehicular misbehavior detection system is shown in Figure 8. When a vehicle enters a new area it must know the RSSI versus distance distribution, either by downloading it from a trustworthy source such as a roadside-unit (RSU) (see Section 7 for further discussion) or predefined. This distribution will be used to classify individual BSMs.

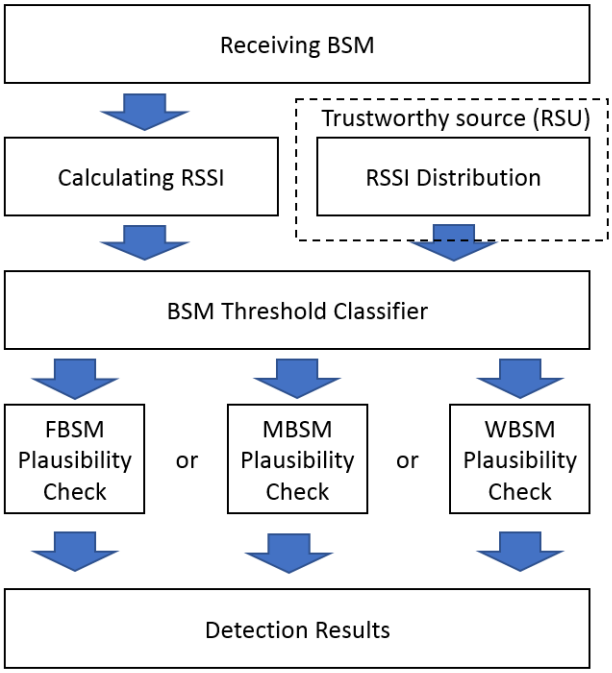

Figure 8: Block diagram of detection misbehavior detection framework

At every reception, the RSSI is computed by the receiver. Using the RSSI distribution and the RSSI computed by the vehicle, the BSM will be classified as normal or as anomalous. The class of the BSM will be used as an input to the three different plausibility checks described in this Section. The output of these plausibility checks classify the sender as normal or misbehaving.

In order to check if a vehicle is misbehaving, the RSSI and location of each BSM must be cross-validated. Our approach, shown in Figure 9, is to generate confidence intervals for normal RSSI values based on the BSM dataset for normal behavior. For each BSM, we record the GPS coordinates from the transmitting vehicle and the receiving vehicle to measure the difference in distance between the two vehicles. Then, we group together BSMs with similar distances between transmitting and receiving vehicles, and we compute the confidence interval for each group based on the mean and the variance. The resolution of the grouping is 1 meter, meaning that each group contains BSMs whose distances between the transmitting and receiving vehicles differ by at most 1 meter. Using the confidence intervals as thresholds, every individual BSM is checked for potential malicious activity. If the RSSI of the BSM at a given distance from the receiver is outside of the confidence interval, then we mark the BSM as anomalous. The thresholds are illustrated in Figure 9.

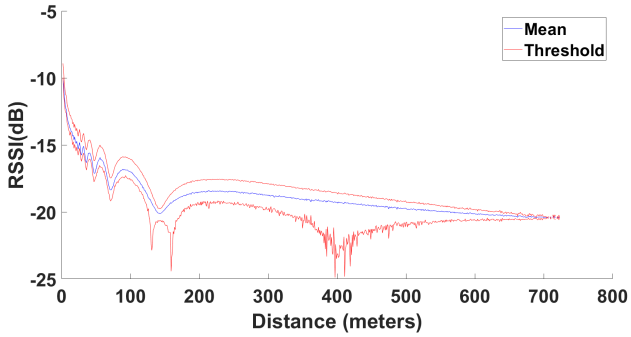

Figure 9: Thresholds on RSSI value per meter

4.2 Misbehavior Detection Algorithms

We propose three vehicular plausibility checks based on the BSM check. The first plausibility check is the "safety first" approach. In this plausibility check, once a BSM is outside of the confidence interval, the transmitting vehicle is immediately classified as a misbehaving vehicle. We refer to this plausibility check as the **First-BSM** or the FBSM approach.

The second plausibility check uses a majority rule. If the majority of the BSMs are classified as malicious, then the vehicle is classified as malicious. This plausibility check will be referred to as the **Majority-BSM** approach or the MBSM approach. There should be a difference in performance between FBSM and MBSM. FBSM should yield more false positives as it is more sensitive, whereas MBSM should have a higher miss-rate.

The third plausibility check is referred to as the **Weighted-BSM** or WBSM plausibility check. This plausibility check assigns a score to each vehicle, and updates this score for every new BSM the vehicle receives from the transmitting vehicle according to a weighted moving average. If the vehicle reaches a score that is below a threshold, which is outside the 99.7% of normal vehicle scores, then the vehicle is classified as an attacker. The formula for calculating the plausibility score using the WBSM test is

$$Plausibility\ Score = [(1-\alpha) * Score_{prev}] + (\alpha * BSM_{score}),$$

where

- $\alpha \in [0;1]$ is the weight given to new information;
- $Score_{prev}$ is the most recent plausibility score of the vehicle;
- BSM_{score} is a Boolean variable equal to 0 if the BSM is classified as *malicious* and equal to 1 if classified as *normal*.

5 EXPERIMENTS

We conduct experiments using Matlab. We evaluate the performance of the plausibility checks in terms of correct-classification rate (CCR), precision and recall, and compare the performance to the application layer plausibility checks of [14]. The correct-classification rate is defined as the fraction of the correct classifications to the total number of classifications. The precision is defined as $\frac{TP}{TP+FP}$; a true positive (TP) is an attacker that is detected as an attacker, and a false positive (FP) is a normal vehicle detected as an attacker. The recall is defined as $\frac{TP}{TP+FN}$, where a false negative (FN) corresponds to an attacker not being detected as an attacker.

Table 2: BSM detection results for different attack type and confidence intervals. The best performing method is highlighted in bold.

Detection Method	CCR	Precision	Recall
Constant Position			
95%	0.9331	0.9043	**0.8857**
99.7%	**0.9583**	**0.9935**	0.8761
Constant Offset			
95%	0.8144	0.8465	**0.5166**
99.7%	**0.8279**	**0.9880**	0.4709
Random Position			
95%	0.9710	0.9254	**0.9987**
99.7%	**0.9999**	**0.9954**	**0.9987**
Random Offset			
95%	0.8293	0.7764	**0.4605**
99.7&	**0.8489**	**0.9826**	0.4122
Eventual Stop			
95%	0.8397	0.8357	**0.5587**
99.7%	**0.8616**	**0.9886**	0.5304

Our first experiment aims to identify which confidence interval, 95% or 99.7%, yields the best thresholds for determining malicious BSMs. Note that 99.7% or 3 standard deviations is commonly used a confidence interval to identify outliers/anomalies [7], hence those are the thresholds chosen to be tested. Toward this end, the data from each individual position forging attack, constant position, constant offset, random position, random offset, and eventual stop, are grouped and tested. Then, an overall test is conducted in which all of the different attack types are merged together. For each test, the confidence interval used for the BSM plausibility check is obtained to ensure optimal results as the mean and variance of each dataset may slightly vary. We compare the precision, recall, and correct-classification rate (CCR) to determine the best confidence interval to use for the plausibility checks. Once the experiment for choosing the most suitable threshold is completed, we test the FBSM and MBSM plausibility checks. For each type of attack, the FBSM and MBSM plausibility checks are implemented and evaluated. Next, we perform tests for the WBSM plausibility check. For this plausibility check, we consider values for α ranging from 0.05 to 0.95, with a step size of 0.05. We ultimately select the value of α that yields the maximum CCR, along with the corresponding precision and recall.

Note that initially, for each attack type, we use a different dataset to evaluate the performance of the plausibility checks. For each different attack type, we collect over $4 * 10^5$ BSMs over a span of 360 seconds in the VEINs simulation. Then, in Section 6.7, we merge these datasets into a single dataset and test the performance of the plausibility checks over the 5 different types of attacks.

Table 3: Detection results of physical layer vs. machine learning-based application layer plausibility checks of [14]. The best performing method is highlighted in bold.

Detection Method	CCR	Precision	Recall
Constant Position			
KNN [14]	0.9452	0.9521	0.8328
SVM [14]	0.9564	**1.000**	0.8290
FBSM	0.9572	0.9344	**0.9300**
MBSM	**0.9641**	0.9997	0.8868
WBSM	0.9634	0.9545	0.9286
Constant Offset			
KNN [14]	0.7508	0.5613	0.1937
SVM [14]	0.7543	0.5729	0.1788
FBSM	0.8943	0.9180	**0.7311**
MBSM	0.8150	**0.9993**	0.4154
WBSM	**0.8988**	0.9424	0.7244
Random Position			
KNN [14]	0.9463	0.9506	0.8363
SVM [14]	0.9116	0.8149	0.8860
FBSM	0.9832	0.9537	**1.000**
MBSM	**0.9999**	**0.9997**	**1.000**
WBSM	0.9886	0.9681	**1.000**
Random Offset			
KNN [14]	**0.9471**	0.9627	0.8253
SVM [14]	0.9177	0.8035	**0.8755**
FBSM	0.9307	0.9126	0.8068
MBSM	0.8143	**0.9986**	0.2762
WBSM	0.9368	0.9402	0.8049
Eventual Stop			
KNN [14]	0.8173	0.7143	0.4254
SVM [14]	0.8403	0.8162	0.4636
FBSM	0.8949	0.9129	**0.7120**
MBSM	0.8366	**0.9998**	0.4471
WBSM	**0.9005**	0.9404	0.7081

6 RESULTS

6.1 Confidence Intervals

Table 2 shows results for 95% and 99.7% confidence intervals, where the best results are highlighted. From Table 2, we note that the precision for the 99.7% confidence interval is much higher than the precision for the 95% confidence interval. Yet, the difference between the recall between the two confidence interval is marginal. Therefore, for the plausibility checks, a 99.7% confidence threshold appears more suitable.

6.2 Attack Type 1: Constant Position

The results for the constant position attacks are shown in Table 3, and the best results are highlighted. The table shows that CCR and recall for FBSM, MBSM, and WBSM are higher than for the SVM and KNN application layer plausibility checks of [14]. While the precision of SVM is perfect and the precision for KNN is higher than FBSM, it is important to note that [14] uses a plausibility check specifically targeting a constant position attack. This same reasoning strengthens the need for the proposed physical layer plausibility checks. Indeed, the recall for the three physical layer checks ends up being higher because a receiving vehicle sometimes receives only one BSM from a transmitter. The application layer has no way of determining if the first BSM carries any inconsistent information. On the other hand, the physical layer checks can readily check if the RSSI is plausible based on the location advertised in that BSM.

We also observe that the detection rates for the FBSM approach are better than the detection rates for MBSM. However, in terms of precision and CCR, the results for the MBSM approach are better. Indeed, for attack type 1, it may be possible that an attacker vehicle is within a plausible distance away from the receiver vehicle the majority of the time, hence causing a lower detection rate compared to the FBSM method. The precision of the FBSM method is lower because of its high sensitivity. Indeed, it only takes 1 bad BSM, which can be due to unusually high interference, to classify the entire vehicle as an attacker.

The WBSM method performs better than the FBSM, but worse than the MBSM in terms of precision, and better than MBSM and worse than FBSM in terms of recall. However, the recall of the WBSM method is within 1% to the recall of the FBSM method, but outperforms the FBSM method by over 2% in terms of precision. In this case, the WBSM method seems to strike the best balance between recall and precision.

6.3 Attack Type 2: Constant Offset

Table 3 shows the drastic performance performance improvement of physical layer plausibility checks over machine learning based application layer plausibility checks for constant offset attacks. MBSM beats both KNN and SVM by over 20% and FBSM and WBSM beats KNN and SVM by over 50% in terms of recall. The reason for this significant difference is that the constant offset attack on the application layer behaves like a normal vehicle. The position, velocity, and acceleration all behave normally, hence it is virtually undetectable.

Table 3 also shows a significant difference in recall between FBSM and MBSM. The recall of the FBSM method beats the MBSM method by over 30%. MBSM's advantage in precision is 8%, which is marginal compared to the recall advantage of the FBSM plausibility check. The overall CCR of the FBSM method is about 8% higher than the MBSM method, which implies that the FBSM method is better than the MBSM method in detecting attack type 2. The WBSM method performs similar to the FBSM method, with a slightly lower recall but about 3% higher precision than the FBSM method. Hence, the WBSM method again can be viewed as the best performing method.

6.4 Attack Type 4: Random Position

The results for attack type 4 show that all of the plausibility checks perform perfectly in terms of recall as seen in Table 3. This shows that once again the three physical layer plausibility checks outperform application layer-based machine learning. The main difference is that the physical layer checks need only one BSM to reach a conclusion whether a vehicle is malicious or not. Other the other hand, tn the application layer, a receiving vehicle needs at least two BSMs to reach a conclusion.

The main difference between FBSM, MBSM, and WBSM is the precision performance. The MBSM method is the best choice for detecting attack type 4 since it has the highest precision value.

6.5 Attack Type 8: Random Offset

Interestingly, random offset is the only attack in this dataset, where the physical layer methods do not outperform SVM and KNN. This is most likely because random offset is seen as noise, from the perspective of the physical layer. However, SVM and KNN are only marginally better than FBSM and WBSM.

Similar to the results for type 2, there is a significant advantage for using FBSM over MBSM in terms of recall. As seen in Table 3, FBSM's recall is over 50% better than MBSM's recall, making the 8% difference in precision negligible. The reason for this significant difference in performance is that, since the offset is randomly placed, the majority of the malicious BSMs might are within the threshold, hence causing the MBSM plausibility check to incorrectly classify the vehicle. The performance of the WBSM method is much closer to the FBSM method, and similarly to all the results, WBSM has a higher precision with only a slightly lower recall.

6.6 Attack Type 16: Eventual Stop

For this attack type, FBSM and WBSM outperforms SVM and KNN by 25% in detection rate, and over 10% in precision. The detection rate of MSBM is similar to that of SVM and KNN, however the precision is over 17% better.

We also note a significant performance gap between FBSM and MBSM. FBSM performs 26% better than MBSM in terms of recall. In terms of precision MBSM performs better by 8%. The reason for the drastic difference in recall is that in some cases the labeled attacker vehicle does not misbehave until the end of the communication transaction. This means that for more than 50% of the transactions, the attacking vehicle is communicating with the receiver normally; hence the MBSM check fails. The WBSM plausibility check shows an improvement in precision and a slight decrease in recall, indicating that WBSM is the best method here.

6.7 Overall Detection

The overall detection results from the combined dataset using all 5 attacks for the three physical layer plausibility checks and two application layer plausibility checks are shown in Table 4. With the exception of MBSM, FBSM and WBSM perform close to 20% better in terms of detection rate than SVM and KNN; in addition to the significant improvement in detection rate, the precision for all three physical layer plausibility checks are much better than SVM and KNN. This shows overall that even without machine learning, detection of position spoofing at the physical layer is more effective than at the application layer (although the two approaches could certainly be used in conjunction, as discussed in the sequel).

Table 4: Overall detection results. Best performing method is highlighted in bold.

Detection Method	CCR	Precision	Recall
KNN [14]	0.8788	0.8879	0.6166
SVM [14]	0.8838	0.8716	0.6515
FBSM	0.9312	0.9294	**0.8421**
MBSM	0.8860	**0.9996**	0.6280
WBSM	**0.9376**	0.9591	0.8373

We observe that overall the MBSM approach yields the highest precision (i.e., the lowest false-positive rate). However, the FBSM and WBSM approaches outperform the MBSM approach by 20% in terms of detection rate, and have over a 5% higher overall classification rate. This shows that the FBSM and WBSM tests are more suitable approaches to support a misbehavior detection system. The WBSM test can be viewed as more robust, since the detection rate is consistently at most 1% lower than with the extremely sensitive FBSM method, while the precision is higher by 3%.

7 DISCUSSION

7.1 Real-time Detection: Access to Local RSSI Distribution

In a real-time detection system, concerns may arise on how vehicles will be able to get the data in order to produce the thresholds. One solution could reside in using road-side units (RSU). Since RSSI confidence intervals are sensitive to the environment, the RSUs can be used to store data and produce the thresholds for their surrounding environments. The RSUs can then relay this information to the vehicles entering the communication range of the RSU, using WAVE Service Advertisement (WSA) messages for example. However, the robustness and reliability of these plausibility checks rely on the trustworthiness and accuracy of the source that provides the RSSI distribution (the RSUs in the case described earlier). If that source is compromised, then the vehicle will not be able to rely on this detection method alone. A fail-safe could be implemented where a vehicle can survey and build the RSSI distribution on its own or in collaboration with neighboring vehicles.

7.2 Stronger Attacker Model

A more advanced attacker model could include an attacker that adjusts the transmission power in order to fool receiver(s). However, with our detection system, this attack can only fool one vehicle. Since the other vehicles will be in a different relative location from the attack, those vehicles will still be able to detect an attack and report the vehicle. To fool multiple spaced receivers, a even stronger attacker is required, e.g. one that could perform multi-antenna beamforming. Beam forming allows an attacker to only communicate with one vehicle without being detected by the others [11, 19].

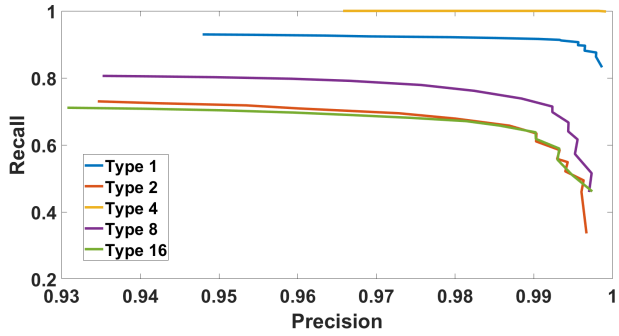

Figure 10: Precision-Recall curves of WBSM for each attack type. These curves are obtained by varying α from 0.05 to 0.95, with step increments of 0.05.

7.3 Assumption on Majority of Honest Neighbors

The plausibility checks proposed do not rely on other vehicles' data in order to detect misbehaviors, since the vehicle will determine the RSSI locally within its system. The only prior information vehicles need is the distribution of RSSI vs. distance, which can be provided by a trustworthy source such as an RSU. In doing so, this relaxes the constraint that the majority of the vehicles in an area must be honest to detect misbehavior.

7.4 Precision vs Recall

The precision vs. recall trade-off is frequently mentioned in this paper. From a safety point of view, a high recall is most desirable, which implies that FBSM would be the best approach. However, due to the very marginal loss in recall, versus the large increase in precision, we believe that WBSM is the best among the three methods. In practice, the claim of which plausibility check is the best should be based upon the priorities of the operator. However, we note that these plausibility checks could be used together as well. FBSM could be used as the initial detection method, and MBSM and WBSM can later be used to further investigate the vehicle to confirm the claim that the vehicle is misbehaving.

7.5 Detection Latency

Another important factor to take into consideration is the latency of detection. The FBSM approach may have the highest false positive rate, but it also has the lowest latency in term of detection. The MBSM approach has the highest precision but also the highest latency. This is true for an attack such as the sudden stop attack. Since the vehicle builds trust inside the system by behaving normally, it will need to misbehave for a long period of time in order to be classified as a misbehavior, explaining why WBSM is the more balanced solution. This method assigns more weights to recent BSMs, thus, decreasing the latency of MBSM, yet keeping a higher precision than FBSM. Results for different settings of the weighting parameter α are shown in Figure 10. The choice of α can heavily influence latency; if α is close to 0 then it will simply behave like MBSM, and if the value of α is close to 1 then it will behave similar to FBSM. Hence, α must be tuned in such a way that it is reasonable for the operator to precisely, accurately, and quickly detect misbehaving vehicles in the system. However, the problem of balancing trust versus latency, and precision versus recall will always result in a trade-off between the proposed methods.

7.6 Complementing Application Layer Misbehavior Detection

One should note that attack type 2, i.e. the constant offset attack, is a very difficult attack to detect on the application layer. This is because the ghost vehicle generated by an attacker is a shadow of a real vehicle. All of the contents inside the BSM cross-validate each other, leading to no inconsistencies between any of the fields. Using physical layer properties such as RSSI, we showed that this attack can be detected. However, we believe that these plausibility checks should be used as a subset of a larger misbehavior detection system that includes application layer plausibility checks in order to increase the overall detection performance.

8 CONCLUSION

In this paper, we propose three different plausibility checks to build a physical layer misbehavior detection framework. Using VeReMi, we evaluate plausibility checks for five different types of attacks, before evaluating their overall performance in aggregate. We find that overall WBSM achieves the best performance, yielding a 83.73% recall with a 95.91% precision rate. Comparing the results to application layer plausibility checks [14], we find that WBSM and FBSM perform much better in every performance metric, by around 5% in CCR, 5% in precision, and 20% in recall.

Our paper makes other important contributions, including (i) generating and publishing an enhanced VeReMi dataset [16] that should facilitate the evaluation of physical layer misbehavior algorithms, and

(ii) validating the RSSI data produced by VeReMi with actual RSSI data traces. Our work also characterizes the RSSI signature of the five forging position attacks simulated in VeReMi.

We conclude by observing that physical layer detection is an effective way of detecting position spoofing attacks, including powerful traffic congestion attacks. As future work, it would be interesting to evaluate our approach against other attacks, such as Sybil attacks. One could also investigate stronger attackers that try to obfuscate their physical layer properties. Finally, one could leverage other physical layer properties, such as channel state information (CSI), to improve the detection and robustness of the system.

Acknowledgment

This research was supported in part by NSF under grant CNS-1409053.

REFERENCES

[1] Kyoungho Ahn, Hesham Rakha, and David K Hale. [n. d.]. *Multi-modal intelligent traffic signal systems (MMITSS) impacts assessment.* Technical Report. No. FHWA-JPO-15-238.

[2] Norbert Bißmeyer, Joël Njeukam, Jonathan Petit, and Kpatcha M Bayarou. 2012. Central misbehavior evaluation for vanets based on mobility data plausibility. In *9th ACM international workshop on Vehicular inter-networking, systems, and applications.*

[3] Bastian Bloessl, Florian Klingler, Fabian Missbrenner, and Christoph Sommer. 2017. A systematic study on the impact of noise and OFDM interference on IEEE 802.11 p. In *Vehicular Networking Conference (VNC), 2017 IEEE*. IEEE, 287–290.

[4] Caocsar. 2019. UMTRI Connected Vehicle Dataset. https://github.com/caocscar/ConnectedVehicleDocs.

[5] Qi Alfred Chen, Yucheng Yin, Yiheng Feng, Z Morley Mao, and Henry X Liu. 2018. Exposing Congestion Attack on Emerging Connected Vehicle based Traffic Signal Control. In *Network and Distributed Systems Security (NDSS) Symposium 2018*.

[6] Sushanta Das and Mounita Saha. 2015. Autonomous vehicle positioning system for misbehavior detection. US Patent 8,954,261.

[7] WJ Dixon. 1953. Processing data for outliers. *Biometrics* 9, 1 (1953), 74–89.

[8] Javier Gozálvez, Miguel Sepulcre, and Ramon Bauza. 2012. IEEE 802.11 p vehicle to infrastructure communications in urban environments. *IEEE Communications Magazine* 50, 5 (2012).

[9] IEEE 1609 Working Group et al. 2016. IEEE Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages. *IEEE Std* (2016), 1609–2.

[10] Jyoti Grover, Nitesh Kumar Prajapati, Vijay Laxmi, and Manoj Singh Gaur. 2011. Machine learning approach for multiple misbehavior detection in VANET. In *International Conference on Advances in Computing and Communications*. Springer.

[11] Liang Liu, Rui Zhang, and Kee-Chaing Chua. 2013. Secrecy wireless information and power transfer with MISO beamforming. In *Global communications conference (GLOBECOM), 2013 IEEE*. IEEE, 1831–1836.

[12] James O'Hara. 2019. Wyoming CV Pilot. https://goo.gl/FqvJHn.

[13] Sushmita Ruj, Marcos A Cavenaghi, Zhen Huang, Amiya Nayak, and Ivan Stojmenovic. 2011. On data-centric misbehavior detection in VANETs. In *IEEE Vehicular technology conference*.

[14] Steven So, Prinkle Sharma, and Jonathan Petit. 2018. Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET. In *IEEE 17th International Conference on Machine Learning and Applications ICMLA 2018*. IEEE.

[15] Christoph Sommer, Reinhard German, and Falko Dressler. 2011. Bidirectionally coupled network and road traffic simulation for improved IVC analysis. *IEEE Transactions on Mobile Computing* 10, 1 (2011).

[16] So Steven. 2019. Modified Veremi Dataset. https://github.com/stevenso8/WiSec_DataModifiedVeremi_Dataset

[17] Mingshun Sun, Ming Li, and Ryan Gerdes. 2017. A data trust framework for VANETs enabling false data detection and secure vehicle tracking. In *Communications and Network Security (CNS), 2017 IEEE Conference on*. IEEE, 1–9.

[18] Rens W van der Heijden, Thomas Lukaseder, and Frank Kargl. 2018. VeReMi: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. *arXiv preprint arXiv:1804.06701* (2018).

[19] Barry D Van Veen and Kevin M Buckley. 1988. Beamforming: A versatile approach to spatial filtering. *IEEE ASSP Magazine* 5, 2 (1988), 4–24.

[20] Aifeng Wu, Jianqing Ma, and Shiyong Zhang. 2011. RATE: a RSU-aided scheme for data-centric trust establishment in VANETs. In *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*. IEEE, 1–6.

[21] Gongjun Yan, Stephan Olariu, and Michele C Weigle. 2008. Providing VANET security through active position detection. *Computer communications* 31, 12 (2008), 2883–2897.

[22] Yuan Yao, Bin Xiao, Gaofei Wu, Xue Liu, Zhiwen Yu, Kailong Zhang, and Xingshe Zhou. 2017. Voiceprint: A Novel Sybil Attack Detection Method Based on RSSI for VANETs. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*. IEEE, 591–602.

[23] Bo Yu, Cheng-Zhong Xu, and Bin Xiao. 2013. Detecting sybil attacks in VANETs. *J. Parallel and Distrib. Comput.* 73, 6 (2013), 746–756.