Intelligent Misbehavior Detection System for Detecting False Position Attacks in Vehicular Networks

Faisal Hawlader[+], Abdelwahab Boualouache (*)[+], Sébastien Faye[++], and Thomas Engel[+]
[+] SnT, University of Luxembourg, Luxembourg
[++] ITIS, Luxembourg Institute of Science and Technology, Luxembourg
Email: [+]{faisal.hawlader, abdelwahab.boualouache, thomas.engel}@uni.lu
[++] sebastien.faye@list.lu

***Abstract*—Position falsification attacks are one of the most dangerous internal attacks in vehicular networks. Several Machine Learning-based Misbehavior Detection Systems (ML-based MDSs) have recently been proposed to detect these attacks and mitigate their impact. However, existing ML-based MDSs require numerous features, which increases the computational time needed to detect attacks. In this context, this paper introduces a novel ML-based MDS for the early detection of position falsification attacks. Based only on received positions, our system provides real-time and accurate predictions. Our system is intensively trained and tested using a publicly available data set, while its validation is done by simulation. Six conventional classification algorithms are applied to estimate and construct the best model based on supervised learning. The results show that the proposed system can detect position falsification attacks with almost 100% accuracy.**

***Index Terms*—Vehicular Networks; Security; Machine Learning-based Misbehavior Detection Systems;**

I. Introduction

The development of Intelligent Transportation Systems (ITS) has been a major step forward for constructing smart cities, allowing users to be better informed and make their participation safer on the road. The evolution of information technologies, together with the variety of network access mechanisms and service providers, has led to the birth of Cooperative Intelligent Transportation Systems (C-ITS). A C-ITS aims at implementing innovative services associated with transport and traffic management [1].

Vehicular networks have raised a huge interest both in academia and industry. The technologies that are connected to C-ITS are mainly intended to make future transportation systems safer and comfortable via two main categories of applications: safety-related applications such as emergency reporting and collision warning, and non-safety-related applications such as Internet access and location-based services. However, vehicular networks are vulnerable to many internal and external attacks such as message droppings and denial of service that can lead to hazardous situations for drivers and passengers [2]. While external active attacks can easily be avoided using cryptographic solutions, internal attacks are difficult to avoid using these same solutions since internal attackers are already authenticated in the considered network [3]. Alternatively, using Misbehavior Detection Systems (MDSs) is an efficient way to detect internal attacks. Position falsification attack is one of these internal attacks, where attackers broadcast false position information to damage the vehicular system or obtain personal benefits on road traffic. Position falsification attacks have serious risks on Cooperative, Connected, and Automated Mobility (CCAM) applications [4]. Therefore, securing communication against position falsification attacks has become a fundamental requirement. Recently, vehicular networks have benefited from the advances in machine learning in the areas of network security. Indeed, several ML-based Misbehavior Detection Systems (ML-based MDSs) have been proposed for the efficient detection of false position attacks [5–10]. However, existing solutions leverage numerous features which increase the computational complexity and overhead. This is because vehicles have to spend a significant amount of time retrieving relevant information for multiple sources and calculating required features before inferring the machine learning model, which results in considerable delays in detecting attacks.

To address the aforementioned issue, this paper proposes a novel scheme based on machine learning techniques to detect position falsification attacks accurately. We provide a synthesis of the existing work associated with position falsification attacks to identify their limitations. We then propose a novel method for feature extraction based on vehicles' positions and develop an accurate multi-class classifier for detecting different types of position falsification attacks. Finally, we assess the performance of our system for detecting position falsification attacks using several metrics.

The remainder of this paper is organized as follows. Section II describes related works. Section III presents the system and attacker models. Section IV describes our proposed misbehavior detection system. The results of

the performance evaluation are presented in Section V. Section VI concludes the paper.

II. Related work

Numerous approaches have been suggested to detect position falsification attacks over the past year through academic research. Grover et al. [5] proposed an approach utilizing a machine learning technique to expose position falsification attackers. Four features were considered to detect the attack geographical position validation: acceptance field verification, speed variation verification, and received signal strength. Issam Mahmoudi et al. [6] proposed a ML-based global misbehavior detection system to analyze the reported misbehavior sent by vehicles and Roadside Units (RSU). A set of algorithms was trained to assess the detection based on a few selected features: (i) plausibility and consistency check features, (ii) communication kinematic data features, and (iii) generic features. So et al. (1), [7] also proposed mechanisms to detect position falsification attacks using a set of plausibility checks. They designed plausibility metrics with six features: (i) local plausibility check: sender's location is compared with a predicted plausible location and the distribution of average acceleration; (ii) movement plausibility check: this feature check the plausibility of the total displacement with the average velocity during the entire trip and compare with total displacement; (iii) quantitative features: these features are numerical description of the vehicle behavior, which represents the difference between the calculated average velocities based on total displacement, time, and the predicted average velocity. Le and Maple [8] suggested ML approaches to detect misbehavior in vehicular networks based on n- sequence trajectory inspection where a sequence of messages was considered to form a trajectory [11]. Three features were used to extract the data: (i) movement plausibility check: focus where the vehicle is moving but reported as uncharged, by observing a sequence of trajectories; (ii) minimum distance to trajectories: aims to find moving patterns of the vehicle in the legitimate set; and (iii) minimum translation to the trajectories. So et al. (2) [9] recently proposed three novel physical-layer plausibility checks. They used ML models to evaluate their proposed checks based on the VeReMi dataset [12]. Gyawali and Qian [10] proposed a cooperative MDS using ML algorithms, where each vehicle is equipped with MDS. They used the VeReMi data set to get the learning features for position falsification attacks and compared each received beacon with the previous one to measure the distance between the sender and receiver. However, this work focuses on the false alter attacks detection rather than position falsification attack.

As we can notice, existing approaches leverage numerous features to detect false position attacks, which complicates the detection process. This is because vehicles have to spend a significant amount of time retrieving relevant information for multiple sources and calculating required features before inferring the machine learning model, which results in considerable delays in detecting attacks. To overcome this issue, we propose a novel approach that provides accurate and fast detection based only on the position information.

III. System Model

A. System Architecture

As shown in Figure 1, our architecture mainly consists of three types of entities:

1) **Trusted Authority (TA)**: is the only trustworthy control center. All significant transactions such as vehicle registration, key management, verification, and confidential keys allocated to the vehicles are deposited in TA since it has sufficient storage and computation capability.

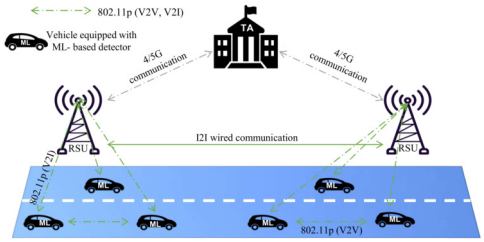

Fig. 1. Vehicular network architecture for the proposed ML-based MDS

2) **Road-Side Unit (RSU)**: one of the fundamental components, equipped with two network interfaces: a wired link to communicate with the neighboring RSUs and a 4/5G to communicate with the TA. We assume all communication links are secure, RSU performs as the intermediaries between TA and vehicles.

3) **Vehicle**: where each vehicle is equipped with an 802.11p network interface to communicate with RSUs (V2I) and other vehicles (V2V). Each vehicle periodically broadcasts a safety message, where each message includes location, time, velocity, etc. We consider each vehicle has the ML-based MDS installed locally, always activated, ready to classify every safety message upon arrival, and any detection should be reported to the TA immediately.

B. System Attacker Model

In this paper, we consider several types of position falsification attacks as defined in [12]:

1) **Constant position attack:** the attacker transmits a fixed, pre-configured constant position.
2) **Constant offset Position attack:** in this case, the attacker generates a fixed offset and adds it to the real positions, which helps the attacker pretend to change the line on the road trip and hide the actual position.

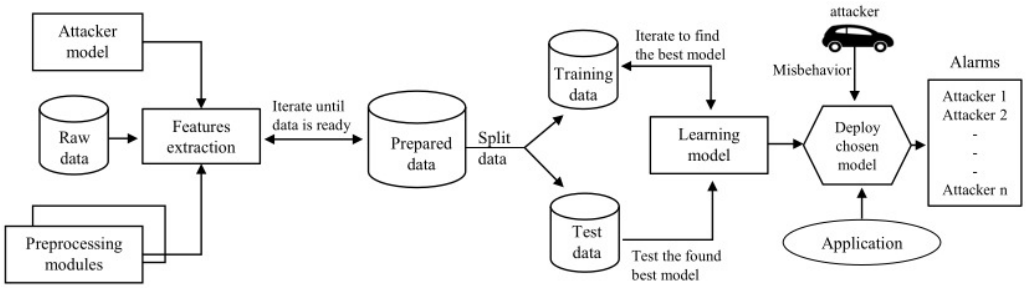

Fig. 2. Machine learning based misbehavior system.

3) **Random position attack:** the attacker transmits a newly generated random position from the vehicular area without considering the real position.
4) **Random offset attack:** this case of attack generates a random offset from a pre-configured rectangular area around the vehicle, which could be considered a close variation of the random attacks. However, in this case, the vehicle chooses a random value that ranged over the rectangle region around the vehicle.
5) **Eventual stop attack:** an attacker behaves normally for a certain time period and then eventually attacks by broadcasting the last position repeatedly (i.e., like it had stopped). This attack can be very harmful as the attacker can pretend not to be on board.

IV. Building a model for ML-based Detection

Our detection mechanism has several stages as shown in the Figure 2. It relies on supervised machine learning techniques, with the aim of building an intelligent MDS model using training data to classify future data as malicious or normal.

A. Data Set

Since supervised ML techniques completely rely on the training dataset, it is crucial to pick the right dataset, which is representative of what might happen in reality. To do so, various strategies could be considered, such as real-world scenario testing, analytical models, and synthetic/simulation-generated datasets. Researchers among the literature are usually considering simulator-generated datasets to validate the system performance due to the lack of real-world scenarios and high-error possibilities of analytical data. Though there are few publicly available data sets, most of them are not an optimal choice due to the scarcity of implemented attacks, messages broadcast standards, and vehicle density. The VeReMi dataset [12], which has been published lately and publicly accessible for research purposes, is a promising one. The VeReMi dataset comprises five position falsification attacks, three vehicle densities (low, medium, and high), three attacker frequencies (10, 20, and 30 %), and each parameter set was replicated five times for randomization, has 225 unique simulations seed. Table I illustrates a brief summary of VeReMi data-set.

TABLE I
Brief summary of VeReMi data-set

VeReMi Data Set				
Time	Traffic Density	Number of Vehicles	Attacker Density	Number of Messages (sent)
3.00	Low	35 to 39	10%, 20%, 30%	908 to 1144
5.00	Medium	97 to 108	10%, 20%, 30%	3996 to 4489
7.00	High	491 to 519	10%, 20%, 30%	20482 to 21878

For a particular scenario, a log file keeps the record of each incoming message from neighboring vehicles (300-meter range) through its complete journey.

B. Feature extraction

For precisely detecting a position falsification attack, we need to extract representative features that describe the various patterns of the attack. Figure 3 illustrates the variation of positions received from normal and suspicious vehicles. Each curve illustrates the variation in position of a given vehicle. The blue curves represent the behaviors of normal vehicles, while the rest represent the behaviors of suspicious vehicles. As we can see, suspicious vehicles can have no variation in the consecutive positions or can have a random variation in position, which can be interpreted as constant and random attacks, respectively.

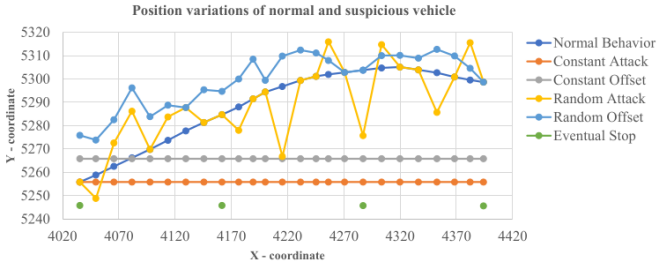

Fig. 3. Variation in position of normal & suspicious vehicles

Thus, the problem becomes how to decide a set of features that can best interpret the variation in position patterns. Our algorithm uses the simple sampling concept from signal processing to convert the variation of positions to a sequence of samples, and each sample will be used as one feature in the learning vector. Figure 4 shows an example of the process of feature extraction from the variation in positions trace. There are two main parameters, sampling length Δ and sampling interval δ, which determine the dimension and attribute value of the feature vector. Specifically, we split the curve into small segments of length δ and calculate the average distance of x_i of each segment. A set of consecutive Δ samples constitute the final feature vector $X = [x_1, x_2, ..., x_\Delta]$.

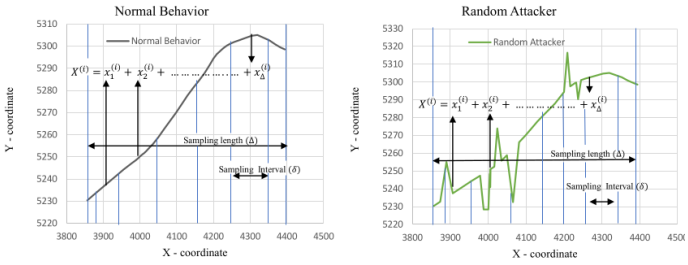

Fig. 4. Sampling process

V. Performance Evaluation

To evaluate our proposed ML-based MDS system's performance, we trained and tested various classification models using the VeReMi dataset. We have also validated the proposed ML model through simulations. This section describes our experiments and the obtained results.

A. Machine Leaning Model Evaluation

As already mentioned, we have used the VeRiMi data set for training and testing classification models. However, before developing a ML model, pre-processing steps were necessary so that the data is fully ready for training and testing. The first stage concentrates on clearing and formatting the source data to prepare an intermediate data set. Indeed, source data contains duplicated, noisy, and additional numbers of features. Therefore, we needed to pick only the required features (position of vehicles). The second stage consists of calculating the distance between two consecutive positions of the vehicles, using formula 1.

$$d_n = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad (1)$$

where d_n represent distance between two positions (x_i, y_i) and (x_{i+1}, y_{i+1}) and n, i are integers. A filtered data set was produced where the features are the distance between the vehicles' two consecutive positions.

In step three, we did shuffle the preprocessed dataset before spilled into training and testing to produce a random situation and avoids having testing data that contains only a dataset of misbehaving cases. Next, we spilled the dataset into two subsets, training (70%) to train the model and testing (30%) for testing the model performance. We considered a set of metrics to evaluate the accuracy of our proposed ML model. The formulas of these metrics are given in Table II, where TP is the true positive, TN is the true negative, FP is the False Positive, and FN is the False Negative. We have built the ML model practicing binary classification (one vs. one) and multi-classification (one vs. rest). Six classification algorithms are used to evaluate and produce the best ML model: Support Vector Machine (SVM), Decision Tree, Random Forest, K-Nearest Neighbor (KNN), Naive Bayes, and Logistic Regression (LR).

TABLE II
Formulas related to the evaluation matrix

Evaluation matrix Formulas	
Metric	Equation
Accuracy:	$\frac{TP+TN}{TP+FP+TN+FN}$
Precision:	$\frac{TP}{TP+FP}$
Recall:	$\frac{TP}{TP+FN}$
F1- score:	$2 * \frac{Precision * Recall}{Precision + Recall}$

*1) **Binary Classification**:* The exact output of binary classification algorithms is a forecast label. The label indicates whether the observation should be classified as normal or misbehaving depending on the patterns of the calculated distances between two incoming consecutive positions. We interpret the label by assigning 0 for normal vehicles and 1 for attacks. The experimental results obtained from a binary classifier model are shown in Figure 5. As

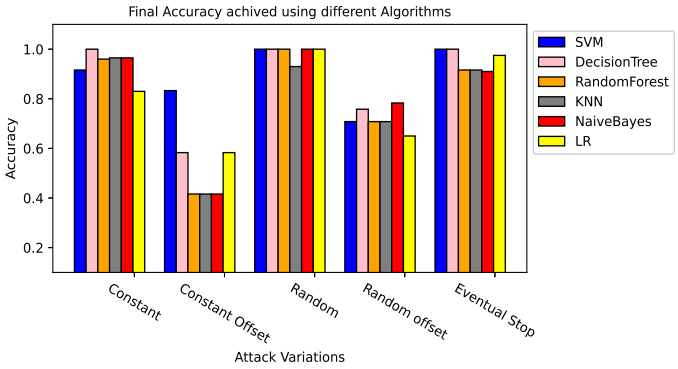

Fig. 5. Accuracy vs attack variations using different algorithms

shown in the Figure, the decision tree algorithm performed well to detect the constant attack and get almost 100% accuracy with the same percents of precision, recall, and F_1- score. However, the rest of the algorithms could get 96.5% accuracy with different adaptations of the estimated

metrics. Besides, the constant offset attack is considered one of the most difficult attacks to detect, as mentioned through the literature [13], but in our case, SVM was able to get 83.3% accuracy, **which is already higher than [10] and [9]**. Moreover, the results also show that all ML algorithms accurately detected the random position attacker, except the KNN algorithm has 83% accuracy.

TABLE III
EXPERIMENTAL RESULTS ACHIEVED BY DIFFERENT CLASSIFIERS FOR MULTI CLASS CLASSIFICATION, BEST RESULTS ARE IN BOLD.

Evaluation Metrics (multi-classification)	ML- model					
Evaluation matrix	SVM	DT	RF	KNN	NB	LR
Accuracy	0.633	0.733	0.696	0.422	0.400	**0.744**
Precision	0.633	0.707	0.696	0.416	0.410	**0.744**
Recall	0.630	0.734	0.696	0.430	0.410	**0.744**
F1-score	0.633	0.733	0.696	0.400	0.400	**0.744**

The binary classification allows us to find a misbehaving vehicle without knowing the types of misbehavior. However, in the real use case, it is also expected to know the type of attack to take action against the attacker. This is why we train a multi-class classifier to distinguish between different types of position falsification attacks.

*2) **Multi-class classification**:* We have also performed a multi-class classification, which takes into account each class of position falsification attacks. Normal vehicles are labeled as 0, whereas attackers are labeled with their corresponding attacks ids as mentioned in [12]. The obtained results are shown in Table III, which summarized the Logistic Regression (LR) has better performance (74% scores) compared to the other algorithms. The multi-class classification provides a more realistic use case scenario since it helps us analyze the corresponding types of attacks. However, as we can see from Table III, our ML model cannot get good accuracy using the VeReMi data set. This is because the VeReMi data set presents some inconsistencies. To evaluate our multi-class classifier, we decided to generate our own data set through simulations, which is consistent and makes a clear difference between position falsification attacks.

B. Simulation

We have carried out a set of simulations to validate the performance of our proposed ML-based MDS. These simulations are conducted using a road traffic simulator, SUMO [14], and a network communication simulator, OMNeT++, linked together thanks to the Veins Simulation Framework [15].

Table IV summarizes the simulation parameters. We considered an urban scenario that models the traffic of the city of Manhattan New York, USA using SUMO. We focused on a region of interest of dimensions 2km x 2km. The vehicles were generated using SUMO to take trips of 5 min duration over the city. We considered 100 vehicles, 30% of them are malicious. In our simulation,

TABLE IV
SIMULATION PARAMETERS

Parameter	**Value**
Simulation duration	300 s
Transmission Range	500 m
Number of vehicles	100
Ratio of misbehaving vehicles	30%
Attacker's number of each type of attack	6

we have considered more advanced position falsification called attack-and-stop attack, which is an extension of eventual-stop attack. In attack-and-stop attack, the attack periodically switches between the attack behavior and the normal behavior.

We have exercised the same experiments on the data set obtained from simulations, as we have previously performed with the VeReMi dataset but excluding the binary classification. The results obtained are shown in Table V, and showing that the DT algorithm performs well with 100% percent scores where the rest of the algorithms have scores higher than 94% percent, which explains the efficiency of the built multi-class classifier.

TABLE V
RESULTS ACHIEVED FOR MULTI CLASS CLASSIFICATION USING SIMULATOR GENERATED DATA SET, THE BEST RESULTS ARE IN BOLD.

Evaluation Metrics (multi-classification)	ML- model					
Evaluation matrix	SVM	DT	RF	KNN	NB	LR
Accuracy	0.983	**1.000**	0.940	0.947	0.982	0.992
Precision	**0.965**	1.000	0.919	0.948	0.771	0.948
Recall	**0.965**	**1.000**	0.947	0.947	0.822	0.991
F1-score	0.965	1.000	0.947	0.944	0.944	0.983

In vehicular networks, the ability of MDS to detect an attack as early as possible is always expected, which demands the lowest computational time. The system requires a minimum number of features for efficient detection, also reduces the computational complexity and saves the resources significantly, most importantly, minimizes the computational time and overhead. From those points of view, we investigated the optimal number of features required to identify the misbehaving (position falsification attack) vehicle in the network. Our experimental results in the Figure 6 show to reach the good accuracy; we should at least have 22 features in the preprocess dataset, which implies that our proposed system only need 23 messages from certain vehicles to classify as normal or an attacker.

C. Results comparison

Figure 7 shows the accuracy of our built multi-class classifier on the VeReMi data set and on the generated data set. We can see that the classification results obtained using the generated data set are better than the results obtained using VeReMi. Indeed, we get almost 100%

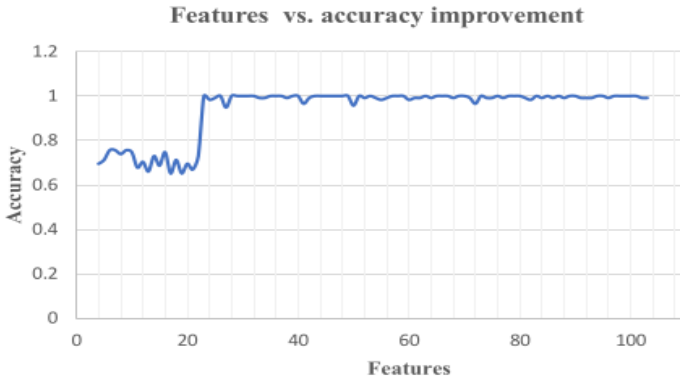

Fig. 6. Accuracy improvement with respect to the features.

accuracy in all cases. As we already mentioned, the low accuracy values obtained using VeReMi are due to the inconsistency of this data set. The results obtained using our generated data set demonstrate the effectiveness of our proposed classifier to detect position falsification attacks.

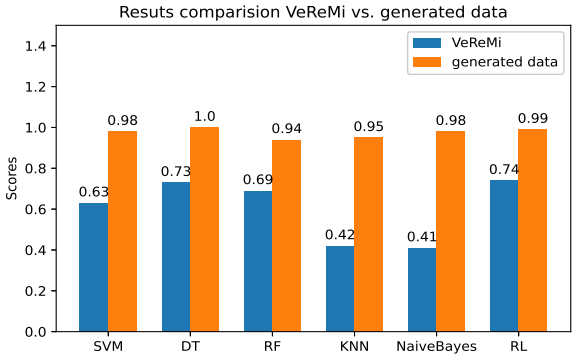

Fig. 7. The final score comparison VeReMi vs our generated data set

VI. Conclusion

This paper proposed an intelligent misbehavior detection system for preventing position falsification attacks in vehicular networks. Leveraging only on position, this system provides accurate and real-time detection. We evaluated the system with a publicly available dataset and using six popular machine learning algorithms. We have also conducted simulations to validate its performance using the Veins Simulation Framework (OMNet++ and SUMO). The results have demonstrated its detection efficiency. As future work, we plan to carry out extensive simulations using large scale realistic scenario to further assess the performance of the proposed system.

Acknowledgment

This work is a part of the 5G-MOBIX project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825496. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains

References

[1] S. Djahel, R. Doolan, G.-M. Muntean, and J. Murphy, "A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 125–151, 2014.

[2] A. Boualouache, R. Soua, and T. Engel, "Sdn-based misbehavior detection system for vehicular networks," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[3] C. A. Kerrache, C. T. Calafate, J.-C. Cano, N. Lagraa, and P. Manzoni, "Trust management for vehicular networks: An adversary-oriented overview," *IEEE Access*, vol. 4, pp. 9293–9307, 2016.

[4] F. Boeira, M. Asplund, and M. P. Barcellos, "Mitigating position falsification attacks in vehicular platooning," in *2018 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2018, pp. 1–4.

[5] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine learning approach for multiple misbehavior detection in VANET," in *International Conference on Advances in Computing and Communications*. Springer, 2011, pp. 644–653.

[6] I. Mahmoudi, J. Kamel, I. Ben-Jemaa, A. Kaiser, and P. Urien, "Towards a reliable machine learning-based global misbehavior detection in c–its: Model evaluation approach," in *Vehicular Ad-hoc Networks for Smart Cities*. Springer, 2020, pp. 73–86.

[7] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 564–571.

[8] A. Le and C. Maple, "Shadows don't lie: n-sequence trajectory inspection for misbehaviour detection and classification in vanets," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. IEEE, 2019, pp. 1–6.

[9] S. So, J. Petit, and D. Starobinski, "Physical layer plausibility checks for misbehavior detection in v2x networks," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, pp. 84–93.

[10] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[11] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1672–1678.

[12] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2018, pp. 318–337.

[13] J. Kamel, A. Kaiser, I. ben Jemaa, P. Cincilla, and P. Urien, "Catch: a confidence range tolerant misbehavior detection approach," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–8.

[14] SUMO, "Simulation of urban mobility," http://sumo.sourceforge.net/, June 2019.

[15] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, "Veins: The open source vehicular network simulation framework," in *Recent Advances in Network Simulation*. Springer, 2019, pp. 215–252.