IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 26, 2021, accepted December 15, 2021, date of publication December 20, 2021,
date of current version January 6, 2022.

*Digital Object Identifier 10.1109/ACCESS.2021.3136706*

# Misbehavior Detection for Position Falsification Attacks in VANETs Using Machine Learning

**SECIL ERCAN, MARWANE AYAIDA, AND NADHIR MESSAI, (Member, IEEE)**
CReSTIC EA 3804, Université de Reims Champagne-Ardenne, 51097 Reims, France

Corresponding author: Marwane Ayaida (marwane.ayaida@univ-reims.fr)

This work was supported by the C-Roads France project funded by the European Commission from the CINEA Agency within the 2015 CEF Transport Programme under Grant 2015-FRTM-0378-S. The statements made herein are solely the responsibility of the authors.

**ABSTRACT** Cooperative Intelligent Transport Systems (C-ITS) is an advanced technology for road safety and traffic efficiency over Vehicular Ad Hoc Networks (VANETs) allowing vehicles to communicate with other vehicles or infrastructures. The security of VANETs is one of the main concerns in C-ITS because there may be some attacks in such type of network that may endanger the safety of the passengers. Intrusion Detection Systems (IDS) play an important role to protect the vehicular network by detecting misbehaving vehicles. In general, the works in the literature use the same well-known features in a centralized IDS. In this paper, we propose a Machine Learning (ML) mechanism that takes advantage of three new features, which are mainly related to the sender position, allowing to enhance the performances of IDS for position falsification attacks. Besides, it presents a comparison of two different ML methods for classification, i.e. k-Nearest Neighbor (kNN) and Random Forest (RF) that are used to detect malicious vehicles using these features. Finally, Ensemble Learning (EL) which combines different ML methods, in our case kNN and RF, is also carried out to improve the detection performance. An IDS is constructed allowing vehicles to detect misbehavior in a distributed way, while the detection mechanism is trained centrally. The results demonstrate that the proposed mechanism gives better results, in terms of classification performance indicators and computational time, than the best previous approaches on average.

**INDEX TERMS** misbehavior detection, machine learning, vehicular ad hoc network, intelligent transport systems, dataset.

## I. INTRODUCTION

Cooperative Intelligent Transport Systems (C-ITS), as a part of Intelligent Transport Systems (ITS), allow effective communication through wireless technologies to provide road safety. Vehicular ad hoc networks (VANETs) are mobile networks where mobile nodes are vehicles. They consist of a set of vehicles equipped with an on board unit (OBU) to ensure Vehicle-to-Vehicle (V2V) communications and infrastructures, called roadside units (RSUs), that exchange with OBUs to ensure Vehicle-to-Infrastructure (V2I) communications.

Security is one of the major challenges of VANETs, not only for vehicles but also for passengers' lives. Since VANETs allow the vehicles to exchange data, attackers can exploit them to threaten both the security of the network and the safety of the passengers. Different types of attacks, like Sybil, DoS (denial-of-service), black-hole, and false data injection attacks, may occur on the V2V or V2I communications [1].

These attacks can be dealt with trust/reputation based schemes, data-centric and entity-centric misbehavior detection schemes, and intrusion detection systems (IDS) as depicted in Figure 1.

Trust/reputation based scheme uses the concept of trustworthiness or reputation of each node. Trust/reputation values of each node are calculated by various approaches to detect less trusted vehicles, namely misbehavior vehicles [2]. This scheme is generally implemented to enhance the performance of routing protocols. Less trusted vehicles are detected and can be excluded from the route selection process.

Data-centric and entity-centric misbehavior detection scheme also calculates trustworthiness using three metrics: direct trust, recommendation trust, and comprehensive trust [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-hoon Kim.

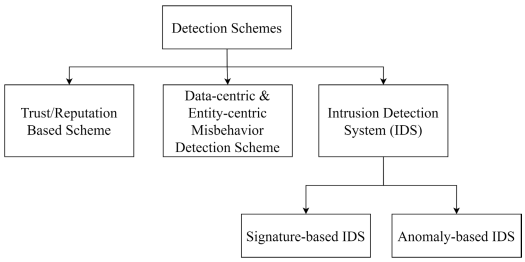



**FIGURE 1.** Classification of detection schemes.

Intrusion detection systems (IDS) are another widely used approach to protect the network. They are generally based on two main detection techniques: signature-based and anomaly-based or eventually a hybridization of these two techniques [4].

Signature-based techniques exploit a database corresponding to the behavior of some known attacks. Then, they compare the collected data with this database in order to detect a behavior that corresponds to a suspicious behavior [5].

Anomaly-based techniques observe the system to catch the deviations (anomalies) and classify them as misbehavior. For anomaly-based IDS on which this paper focuses, several methods have been already proposed [6]–[10], namely watchdog mechanism, machine learning, hashing pseudonym to common values, RSSI-based analysis, honeypot based approach, fuzzy clustering, etc.

IDS in VANETs can be classified based on the decision mechanism as well: Centralized, Decentralized, and Distributed IDS. In general, the decision maker is also the actor that detects the attack. However, there are some cases where the detection is done locally but the final decision is taken by another actor in the network. In centralized IDS, a central station (e.g. trust authority, misbehavior authority, etc.) collects the exchanged messages or misbehavior reports directly from vehicles/RSUs. In the first case, the central station detects the attack, whereas in the second case it evaluates the misbehavior sent reports. The final decision, if there is a launched attack, is taken by the central station and is broadcast to all vehicles. Decentralized IDS is based on the evaluation of attack detection by RSUs or cluster heads. This type of system may have different deployments for detection actors, where it could be done by vehicles, RSUs, or cluster heads. For cluster head based decentralized IDS, the selection of a cluster head is also an additional issue. For distributed IDS, the detection mechanism is deployed within all the vehicles that have to detect the attack and take their own decision/reaction.

In most of the existing misbehavior detection mechanisms in VANETs, the detection is done centrally after combining the data/results of vehicles or infrastructures. Although there are recent works developing distributed intelligent detection models, it still needs to be improved in terms of detection performance by proposing more suitable features and using more adapted learning techniques. Moreover, accurate real-time detection is essential. In this work with an ensemble learning based detection system, training phase will be executed

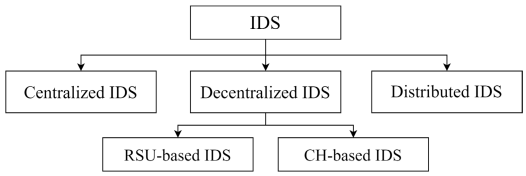



**FIGURE 2.** IDS classification based on the decision mechanism.

centrally. Then, any vehicle will be able to use this trained model to detect rapidly an attack through the new arriving messages/vehicles.

This work proposes a new feature combination related to C-ITS and exploits them to develop a distributed IDS that benefit from machine learning techniques. The performance of the proposed misbehavior detection approach is evaluated by comparing it with other approaches using a specifically built large dataset. The contributions of the work are the following:

- Some new features are introduced to better detect position falsification misbehavior. These position-related features are the estimated angle of arrival, estimated distance between sender and receiver, and the difference between the declared and estimated distance between sender and receiver. These features enhanced well the detection performances.
- A distributed IDS mechanism that exploits the proposed features is suggested. After training is done centrally, each vehicle performs the trained model to decide the misbehaving vehicle whenever it receives a new message.
- An ensemble learning method combining two machine learning techniques is introduced to improve the IDS's performances by aggregating the results of different base learners.
- A comprehensive analysis is done for different traffic densities and attack types to show the efficiency of our proposal.

The paper is organized as follows: the next section consists of a literature review on misbehavior detection. The proposed misbehavior detection mechanism is described in the third section. Section IV presents the used dataset, the results of experiments, and the comparisons. The last section states the conclusion of the paper and suggestions for further studies.

## II. RELATED WORK

VANETs may suffer from several security issues related to authentication, privacy, data non-repudiation, etc. [11]–[14]. To overcome these issues, IDS, which were initially developed for classical computer networks, are more and more used to enhance the security in VANETs. This section aims to present a brief overview of the works related to misbehavior detection in VANETs with a specific focus on machine learning detection techniques. Moreover, on the existing approaches, IDS classification is applied according to the actor that decides whether a vehicle is an attacker or not.

### A. MISBEHAVIOR DETECTION IN VANETs

Several works have studied various intrusion detection mechanisms for VANETs based on a central actor/authority or a decentralized actor such as RSU or cluster head. Malhi and Batra [15] developed a centralized IDS to detect fraudulent warning messages. A trusted central authority collects the data from vehicles and RSUs, and then detects the false information by using an XML dependency tree. Lal and Nair [16] proposed a detection algorithm based on hashing pseudonym to detect Sybil attacks. A centralized station (here, a department of motor vehicle (DMV)), which distributes certified pseudonyms to vehicles, confirms whether an attack has occurred, while checking the coarse-grained hash value of pseudonyms. Another centralized IDS is introduced by Bißmeyer *et al.* [17] for Sybil attack detection, where each vehicle controls a plausibility check to detect attacks and sends reports to a misbehavior evaluation authority. This central authority collects the reports and decides if a node is an attacker or not.

Subba *et al.* [7] presented a multi-layered game theory based and cluster head deployed IDS for selective forwarding, black-hole, DoS, wormhole, and Sybil attacks. It uses a lightweight neural network classifier module in order to detect these attacks. Bouali *et al.* [18] constructed a preventive decentralized mechanism based on Kalman filter to predict the behavior of vehicles for detection of possible DoS, Sybil, false alert, and packet alteration attacks. This decentralized approach where the cluster head monitors vehicles and detects the attack is repeated periodically for a proactive intrusion detection system. Khan *et al.* [19] proposed a decentralized IDS deployed on vehicles and cluster heads for malicious node detection. In this system, detection is done by vehicles considering the load, the distrust value (or reputation), and the distance. Then in case of an attack, vehicles inform the cluster head, which updates the distrust value and broadcasts the decision to other vehicles.

In addition to the works that propose centralized or decentralized detection algorithms, many works developed a distributed intrusion detection mechanism in VANETs. Ghaleb *et al.* [20] introduced a context-aware data-centric misbehavior detection model for local detection of false mobility information. They used consistency and plausibility rules to decide whether a vehicle is suspicious or not. Zhang *et al.* [21] constructed a secure routing protocol using fuzzy logic and Ant Colony Optimization to detect black-hole and flooding attacks. They evaluated the performance of the system in terms of packet delivery ratio, end-to-end delay, and overhead. Ayaida *et al.* [10] developed a mathematical model based on traffic flow theory that allows vehicles to monitor their neighborhood in order to detect an eventual Sybil attack. In fact, they estimate the speed of the neighbors based on the number of vehicles using the fundamental diagram of the road, then they compare it with the declared ones in order to detect the Sybil attack. Yao *et al.* [22] proposed a distributed IDS using a method called Voiceprint, to detect Sybil attacks without any centralized infrastructure. Their method exploits the similarity between RSSI times series and dynamic time warping. Chen *et al.* [23] presented a distributed detection algorithm for Sybil attack, where vehicles use the difference of neighboring nodes' digital signature vectors to detect the attack. Van der Heijden *et al.* [24] generated a public dataset to detect position falsification misbehavior and published it to allow researchers to compare and evaluate different misbehavior detection techniques. The dataset, called Vehicular Reference Misbehavior Dataset (VeReMi), is used as a basis to assess the performances of new detection approaches. Our proposal will be evaluated in this paper using this dataset (i.e. VeReMi) in order to compare its performances with related works that already implemented this dataset.

### B. ML-BASED DETECTION MECHANISMS IN VANETs

One of the main used techniques for misbehavior detection in VANETs is Machine Learning. ML-based techniques learn the behavior of attacks and classify vehicles as normal or misbehavior/attacker. The performance of the approach is evaluated in terms of detection rate e.g. true positive rate, false positive rate, and performance indicators generated by these rates i.e. precision, recall, accuracy, and F1-score. These rates will be discussed in detail in the simulation section.

In this subsection, papers proposing ML-based IDS are studied and also classified based on the decision mechanism starting with the centralized ones. Maglaras [25] presented a centralized detection system for DoS attacks, where detection is done locally by static IDS mounted on RSUs and dynamic IDS mounted on vehicles acting as cluster heads. One-class Support Vector Machine (SVM) classification and K-means clustering methods are combined for detection. Finally, the central system decides if there is an attack after collecting the information from vehicles and RSUs. Zang and Yan [26] introduced an ML-based centralized IDS for DDoS attacks with different vehicle densities, where the process is executed by a centralized collector. Random Forest is used to classify attackers by source IP, destination IP, source port, destination port, protocol, and packet length. Zhang *et al.* [27] implemented SVM to detect false message attacks using, as features, the driving status, speed, acceleration, vehicle type, reputation, and distance. They also studied message suppression attacks dealing with packet drop rate, packet delay forward rate, etc. Their proposal on vehicle trust model requires a central Trust Authority (TA) as well as a local vehicle trust module to combine different assessments. Some other studies were inspired by the public dataset VeReMi and simulated their own scenarios to create new attack types for their approaches [28]–[30]. They also generated several other features (e.g. plausibility, consistency, etc.) as well as kinematic data (e.g. position, speed, acceleration). In these works, a centralized misbehavior authority takes the final decision even if the detection is done by vehicles and RSUs.

Decentralized IDS using ML techniques are also well studied in the literature. Sharma and Kaul [31] proposed a multi cluster head based detection system, where the head is chosen by hybrid fuzzy multi-criteria decision making approaches.

They dealt with packet drop, selective forwarding, and wormhole attacks in this scheme. AOMDV (Ad Hoc On-demand, multipath distance vector) routing protocol is used and dolphin swarm optimization is implemented to select the optimum features and then SVM is performed for separate classes and multi-class as well. Wahab *et al.* [32] defined a cluster head based IDS for detection of packet dropping attacks using an SVM. Ant Colony Optimization is implemented to select the cluster head. Although individual vehicles collect data and find evidence of an attack, the cluster head makes the final decision and broadcasts its decision to the cluster members and other cluster heads. Tian *et al.* [33] proposed a detection mechanism based on bus nodes for DoS attack detection using a Neural network. The bus nodes, which can be considered cluster heads, collect the data and send them to the access points, where the intrusion detection is performed.

In ML-based distributed IDS for VANETs, regardless of whether learning is done by the vehicles, the final decision is generally made by the vehicles using the classifier after the learning phase is complete. Ghaleb *et al.* [34] introduced a feed-forward back-propagation artificial neural network for a distributed misbehavior detection. They derived the necessary features for detecting false mobility information, which are related to misbehavior, environment, and communication status. In [35] and [36], a hybrid context-aware misbehavior detection model is proposed where consistency, plausibility, and behavioral-based features are combined. These two works implemented K-means and RF, respectively, on the before mentioned features to detect the false mobility information. Shams *et al.* [37] presented a distributed detection approach against the misbehaving vehicle that affects packet forwarding performance by dropping or delaying packets. Ad hoc On-demand Distance Vector (AODV) protocol is used as routing protocol and trust aware SVM is implemented for detection. Each vehicle in the packet forwarding route examines the behavior of the next hop to detect any signs of a misbehaving node that could influence the system performance. Kim *et al.* [38] created a public intrusion detection dataset, called KDD CUP 1999, for four attack types: DoS, probing, user to root (U2R), and remote to local (R2L) attacks. They considered these attacks as multi-class in the classification process of SVM and feed-forward neural network. Some features are determined through a trace file, the significant ones are extracted and fuzzified to be used in the distributed detection mechanism. Ghaleb *et al.* [39] developed a distributed misbehavior-aware on-demand IDS and implemented it on the previously mentioned KDD dataset. The important features are selected using chi-square test. They tested their proposed detection model using RF, XGBoost, and SVM.

Several works, using the public VeRemi dataset provided by [24], implemented well-known ML techniques, e.g. k-Nearest Neighbor, Random Forest, Support Vector Machines, and Logistic Regression (LR), while proposing different features for position falsification attack detection [40]–[44]. So *et al.* [40] observed the relation between RSSI and distance of sender-receiver. Different attacks are also detected by their proposal on three plausibility checks: First-Basic Safety Message (F-BSM), Majority-BSM (M-BSM), and Weighted-BSM (W-BSM). They compared the results with the ones obtained from SVM and kNN. Different feature combinations are proposed in [41] to detect misbehavior over a single class by considering all attack types in one overall class. They proposed a feature combination, which includes the position, the speed, the position difference between sender and receiver, and the speed difference between sender and receiver. They apply the classification methods LR with normalization and without normalization, and SVM. Authors of [42], [43] defined two plausibility scores, as well as the difference between the calculated and the predicted speed, the total displacement between the calculated and predicted displacement based on average speed. The first plausibility score is the location plausibility, which checks whether the sender's GPS (Geographical Positioning System) location is in the predicted range of plausible locations; whereas the second one, movement plausibility, checks whether the average speed is different from 0 m/s, when the total displacement is 0 (to monitor the constant location). They used several ML methods, e.g. kNN, SVM, RF, LR, etc., considering these plausibilities as well. In [44], various ML methods such as LR, kNN, DT, RF, and Bagging are implemented for misbehavior detection. The proposed used features are the change in speed between two beacons, the change in position between two beacons, the receiving distance, the RSSI, the change in the speed, and the change in the position.

Table 1 classifies the related works in terms of implemented decision mechanism, number of detected attack types, usage of ML techniques or not. It also details the used detection method. These works point out that an ML-based misbehavior detection mechanism with suitable features is an effective way. Hence, this paper proposes a new distributed IDS by introducing new features and an ensemble learning method to improve the performance of position falsification detection mechanism in VANETs.

## III. PROPOSED DETECTION SCHEME

In this section, new features in our proposed IDS mechanism, related to the vehicles, are derived to enhance the detection of misbehaving vehicles. Then, a distributed ML-based intrusion detection scheme is developed exploiting these features for position falsification attacks in VANETs. The training phase is executed centrally and then the detection phase is distributed all over the vehicles.

Since the features have an important role in misbehavior detection system, a detailed feature selection mechanism is studied here. The latter integrates few commonly used features and new proposed ones.

To detect malicious vehicles, we propose to implement kNN and RF, which are mostly used in misbehavior detection and yield satisfying results. These two ML techniques will be performed together in order to obtain higher classification performances. Hence, an Ensemble Learning method is

**TABLE 1. A summary of related works.**

| Ref. | Year | Decision Mechanism | Number of Attack Types | Usage of ML Techniques | Detection Method |
|---|---|---|---|---|---|
| [15] | 2015 | Centralized | One | - | XML dependency tree |
| [16] | 2015 | Centralized | One | - | Hashing pseudonym |
| [17] | 2012 | Centralized | One | - | Plausibility checks |
| [7] | 2018 | Decentralized | Multi | - | Multi-layered game theory |
| [18] | 2016 | Decentralized | Multi | - | Kalman filter |
| [19] | 2014 | Decentralized | One | - | Calculating distrust value |
| [20] | 2019 | Distr./Decent. | Multi | - | Consistency and Plausibility rules |
| [21] | 2018 | Distributed | Multi | - | Fuzzy logic and Ant Colony Optimization |
| [10] | 2019 | Distributed | One | - | Mathematical model based on traffic flow |
| [22] | 2017 | Distributed | One | - | Voiceprint (based on RSSI) |
| [23] | 2009 | Distributed | One | - | Neighboring nodes' digital signatures |
| [24] | 2018 | Distributed | Multi | - | Accepted range threshold, sudden appearance warning, simple speed check, distance mover verifier |
| [25] | 2015 | Centralized | One | ✓ | One-class SVM and k-means (k-OCSVM) |
| [26] | 2021 | Centralized | One | ✓ | RF |
| [27] | 2018 | Centralized | Multi | ✓ | SVM |
| [28] | 2019 | Centralized | Multi | ✓ | Threshold, non-cooperative trust, cooperative trust based; XG-Boost, SVM, Multilayer Perceptron (MLP), LSTM |
| [29] | 2020 | Centralized | Multi | ✓ | Threshold, Aggregation, Behavioral Analysis; SVM, MLP, LSTM |
| [30] | 2020 | Centralized | Multi | ✓ | RF, XGBoost, NN |
| [31] | 2018 | Decentralized | Multi | ✓ | Dolphin swarm optimization and SVM |
| [32] | 2016 | Decentralized | One | ✓ | SVM |
| [33] | 2010 | Decentralized | One | ✓ | Neural Network |
| [34] | 2017 | Distributed | Multi | ✓ | Feed-forward back-propagation ANN |
| [35] | 2019 | Distributed | Multi | ✓ | K-means |
| [36] | 2019 | Distributed | Multi | ✓ | RF |
| [37] | 2018 | Distributed | One | ✓ | SVM |
| [38] | 2017 | Distributed | Multi | ✓ | SVM, Feed-Forward Neural Network |
| [39] | 2020 | Distributed | Multi | ✓ | RF, SVM, XGBoost |
| [40] | 2019 | Distributed | Multi | ✓ | First-BSM, Majority-BSM, Weighted-BSM; kNN, SVM |
| [41] | 2019 | Distributed | One | ✓ | SVM, LR |
| [42] | 2018 | Distributed | Multi | ✓ | kNN, SVM |
| [43] | 2019 | Distributed | Multi | ✓ | kNN, RF, LR, Long Short-Term Memory (LSTM) |
| [44] | 2019 | Distributed | Multi | ✓ | LR, kNN, DR, RF, Bagging |

proposed to carry out the proposed feature combinations. The details of the used learning methods will also be explained in the next subsections.

### A. COMMON FEATURES

To detect position falsification attacks, the Position (P) and Speed (S) are considered in this work for both the receiver ($R$) and the sender ($S$). These features are commonly used in the previous works in the literature [41], and are respectively denoted: $P_{R,a\in\{x,y\}}$, $P_{S,a\in\{x,y\}}$, $S_{R,a\in\{x,y\}}$ and $S_{S,a\in\{x,y\}}$, where $x$ and $y$ stand for the 2D map dimension axes. Note that these features are directly extracted for the sender from the Basic Safety Messages (BSM) that are exchanged continuously between the vehicles, and retrieved directly from the GPS for the receiver. Moreover, the difference in position and speed between the last two BSM of the same sender [44] are useful features confirmed by the previous works. They are presented in Equations (1) and (2), respectively.

$$\Delta P_{S,a}(i) = P_{S,a}(i) - P_{S,a}(i-1) \tag{1}$$
$$\Delta S_{S,a}(i) = S_{S,a}(i) - S_{S,a}(i-1) \tag{2}$$

where $P_{S,a}(i)$ and $P_{S,a}(i-1)$ denote, respectively, the positions retrieved from the BSM number $i$ and number $i-1$. The same applies for $S_{S,a}(i)$ and $S_{S,a}(i-1)$.

The current position and speed of the receiver are retrieved from its GPS and the variations of the position and the speed are calculated as shown in Equations (3) and (4).

$$\Delta P_{R,a}(i) = P_{R,a}(i) - P_{R,a}(i-1) \tag{3}$$
$$\Delta S_{R,a}(i) = S_{R,a}(i) - S_{R,a}(i-1) \tag{4}$$

where $P_{R,a}(i)$ and $P_{R,a}(i-1)$ denote, respectively, the positions of the receiver, when the BSM number $i$ and number $i-1$ were received. The same applies for $S_{R,a}(i)$ and $S_{R,a}(i-1)$.

Another common and efficient feature, which is used in many approaches for misbehavior detection, corresponds to the distance between sender and receiver [40]–[43]. This feature can be calculated in all dimensions, as shown in Equation (5).

$$\Delta P_{R\to S,a} = |P_{R,a} - P_{S,a}| \tag{5}$$

Difference in speed between sender and receiver [41] can also be computed as in Equation (6).

$$\Delta S_{R\to S,a} = |S_{R,a} - S_{S,a}| \tag{6}$$

### B. PROPOSED FEATURES

To enhance the performances of the previous approaches, this paper proposes some new suitable features.

Our first new feature corresponds to the angle of arrival (AoA) illustrated in Fig. 3. It is obtained by using the arctangent function with the distance between sender

and receiver, which is given by the Equation (7).

$$AoA = \arctan \frac{\Delta P_{R\to S,y}}{\Delta P_{R\to S,x}} \tag{7}$$

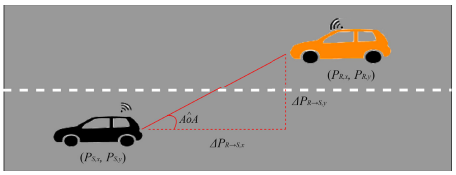


**FIGURE 3.** Angle of Arrival and Distance between sender-receiver.

The second feature suggested in this paper to detect the position falsification attack is the estimated distance between the sender and the receiver. This feature is obtained using the received signal strength indicator (RSSI) according to:

$$RSSI = P_T - P_L(d) \tag{8}$$

where, $P_T$ is the transmission power, $P_L(d)$ corresponds to the path loss in dB at the estimated distance $d$. We assume that the transmission power is constant and is the same for all the nodes. Therefore, it could not be modified by any vehicle.

Then, one can estimate the distance by using the log-normal shadowing model given by Equation (9) [45].

$$P_L(d) = P_L(d_0) + 10n \log(d/d_0) \tag{9}$$

where, $P_L(d_0)$ is the reference power value in dB at a reference distance $d_0$ and $n$ denotes the path loss exponent that depends on the environment.

Eventually, by using the equations (8) and (9), the estimated distance is given by:

$$d = 10^{\frac{P_T - RSSI - P_L(d_0)}{10n}} * d_0 \tag{10}$$

Note that since our objective is to estimate the distance between the sender and the receiver based on the RSSI, the notation $\hat{d}$ will be used in the rest of the paper instead of $d$.

Finally, we propose to exploit the difference between the declared distance between the receiver and sender $d$, and the estimated one $\hat{d}$ as a useful feature to detect misbehavior. Fig. 4 represents the estimated distance $|R \to S|$ and the declared distance $|R \to F|$ between sender and receiver. The difference between these distances is given by:

$$\Delta d_{R\to S} = \left| \sqrt{\Delta P^2_{R\to S,x} + \Delta P^2_{R\to S,y}} - \hat{d} \right| \tag{11}$$

### C. DETECTION TECHNIQUES

Anomaly-based IDS using supervised learning methods have shown their efficiency in many areas, including C-ITS [46]. Among these methods, one can cite SVM, kNN, LR, Decision Tree (DT), Artificial Neural Network (ANN), Bayesian Classifier, etc. In addition, ensemble learning is another alternative classification method that combines ML classification techniques to improve the training process, which could be: bagging, boosting, and stacking [47].

Among all used ML techniques, two of them are mostly preferred and generally show better results in misbehavior detection: kNN and RF [40], [42], [43]. To compare our proposed features with others, these two techniques are suitable in order to avoid the side effects of the used classification technique.

kNN is one of the most known ML techniques for classification problems ($k$ is a pre-defined number of neighbors that should be selected carefully). Fig. 5 gives an example of kNN with different $k$ values. Larger values of $k$ cause less distinct boundaries for classes while decreasing the noise effect on the classification. Therefore, smaller $k$ values are generally preferred. If the problem is binary classification, then an odd number of $k$ is recommended to avoid equality of votes. Since this classification is based on neighbors' votes, the computations rely on distance. The most popular distance calculation methods are Euclidean, Manhattan, Hamming, and Minkowski distance [48].

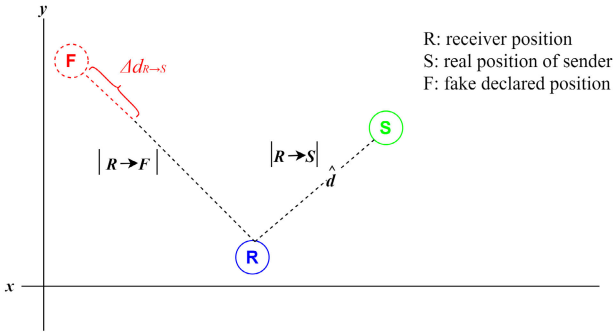



**FIGURE 4.** Estimated and declared distance.

On the other hand, RF is based on DTs that are trained on independent sub-datasets as shown in Fig. 6. Hence, it prevents overfitting issues and classifies the data based on the maximum number of votes. These sub-datasets are randomly chosen with replacement excluding approximately one-third of the training dataset in each split. In addition, the features are randomly selected for the training of the sub-datasets [49].

Stacking is an ensemble learning method that allows the combination of heterogeneous learning techniques in order to enhance the performance of the classification. In such a method, various learning techniques are used in a first classification step on the same dataset. Then, a generalization model, i.e. meta-classifier, combines the predictions of the primary techniques [50]. Generalized Boosted Model (GBM) and Generalized Linear Model (GLM) are two examples of the methods used to combine the results in Stacking method. Thus, this paper proposes to exploit the potential of Stacking method by using kNN and RF for the first step and GLM for the generalization step as shown in Fig. 7. To the best of our knowledge, such an approach has not been yet studied in the literature to detect position falsification attacks in VANETs.

## IV. SIMULATION RESULTS

To show the efficiency of our proposed IDS mechanism and to facilitate the comparison with the existing works, the

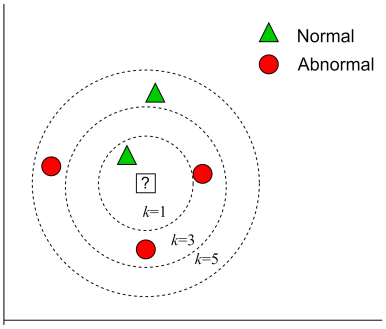



**FIGURE 5.** k Nearest Neighbor.

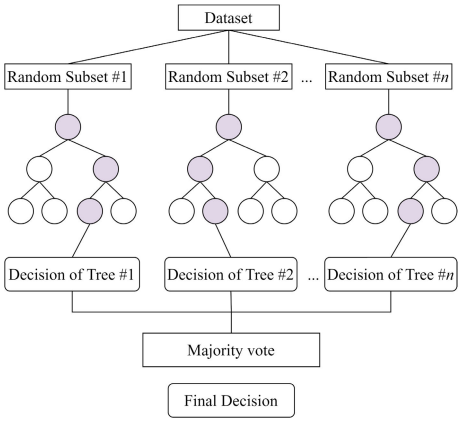



**FIGURE 6.** Random forest.

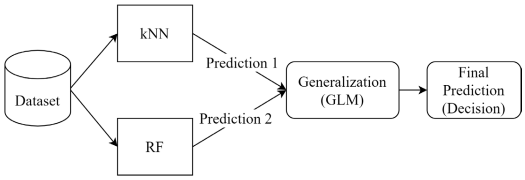



**FIGURE 7.** Stacking procedure in our case.

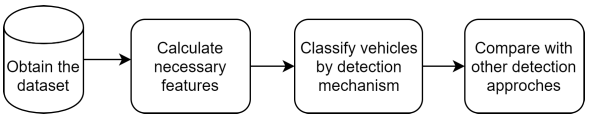



**FIGURE 8.** Flow of the simulation.

simulation results of this paper are based on a public dataset, denoted VeReMi, which has been built specifically for testing misbehavior detection [24].

The main workflow of the simulation is shown in Fig. 8. The first step of this workflow consists of obtaining a real or a simulated dataset and reorganizing it by matching senders and receivers. Then, new features are calculated to be used for the classification of the vehicles as attackers or normal vehicles by using different ML and ensemble learning techniques. Finally, the obtained results are compared with those using features already proposed by other researchers.

### A. DATASET DESCRIPTION AND EVALUATION INDICATORS

VeReMi dataset, which allows evaluating misbehavior detection approaches, is used in this work as a basis for the implementation of the proposed mechanism. Reference [24] performed the simulations with the Luxembourg traffic scenario (LuST) for realistic mobility patterns in a city and used Veins, based on OMNET++ (an event-based network simulator) and Simulation of Urban MObility (SUMO) (a road traffic simulator), for the vehicular network simulations to obtain this dataset.

**TABLE 2.** Attacker parameters in the VeReMi dataset [24].

| Attack ID | Attack Name | Parameters |
|---|---|---|
| 1 | Constant | $x$ = 5560, $y$ = 5820 |
| 2 | Constant Offset | $\Delta x$ = 250, $\Delta y$ = -150 |
| 4 | Random | Uniformly random in playground |
| 8 | Random Offset | $\Delta x$, $\Delta y$ uniformly random from from [-300,300] |
| 16 | Eventual Stop | Stop probability increases by 0.025 each position update (10 $Hz$) |

**TABLE 3.** Confusion matrix for classification.

| | Predicted 0 | Predicted 1 |
|---|---|---|
| Actual 0 | TN | FP |
| Actual 1 | FN | TP |

The published VeReMi dataset consists of 5 different attacks and considers 3 different attacker rates (10%, 20%, and 30%) for different traffic densities (low, medium, and high).

The attacks, defined here, are related to position falsification, i.e. Constant, Constant Offset, Random, Random Offset, and Eventual Stop. In the constant attack with ID 1, an attacker sends a constant position. However, in the constant offset attack with ID 2, it adds a constant offset to its current position. In the random attack with ID 4, an attacker sends a new random position for every message. However, in the random offset attack with ID 8, an attacker sends a new random position in a preconfigured rectangle around the vehicle. Finally, in the eventual stop attack with ID 16, an attacker sometimes attacks by repeatedly sending the actual position. Table 2 gives more details about these attacks.

For each scenario, the dataset consists of a ground truth file and a set of message log files including both GPS data and BSM. GPS data provides the information about the local vehicle and BSM gives the information about the message received from other vehicles through Dedicated Short Range Communications (DSRC).

The performance of the proposal is evaluated using accuracy, precision, recall, and F1-Score, which are well-known performance indicators in classification problems. In order to calculate these performance indicators, the confusion matrix is obtained firstly, which consists of the metrics true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Table 3 explains the meanings of these attributes where, "1" denotes an attacker and "0" a normal vehicle.

Accuracy gives the general correct classification ratio, that is, the rate of true prediction of both 0 and 1 to all cases. Precision is the rate of TP to all cases predicted as 1, whereas recall is the rate of TP to all cases actually 1.

Precision and recall show an inverse trend: when precision increases, recall will decrease and vice versa. Therefore,

**TABLE 4.** Comparison of features and detection techniques.

| Paper | Year | Features | Nb Features | Techniques |
|---|---|---|---|---|
| [40] | 2019 | RSSI, distance between sender and receiver, F-BSM, M-BSM, W-BSM | 5 | kNN, SVM |
| [41] | 2019 | Position, speed, distance between sender and receiver, difference of speed between sender and receiver | 4 | LR, SVM |
| [42], [43] | 2018, 2019 | Location plausibility, movement plausibility, difference between calculated and predicted speed, magnitude of difference for speed, total displacement between calculated and predicted displacement | 5 | kNN, SVM, RF, LR, LSTM |
| [44] | 2019 | RSSI, delta position of sender, delta speed of sender, distance between sender and receiver, delta position of receiver, delta speed of receiver | 6 | kNN, RF, DT, LR, Bagging |

a trade-off indicator between precision and recall is needed, which is called F1-Score. F1-Score calculates the harmonic average of these two indicators to solve this issue.

### B. DISCUSSION FOR RESULTS OF FEATURES COMPARISON

In order to show the interest in the new proposed features and the efficiency of the used approach, this section compares our results with those obtained by other works that are implemented using VeReMi dataset [40]–[44]. A summary of these works is given in Table 4.

To be able to present the performance of the proposed mechanism, the new features proposed in this work (AoA, $\hat{d}$, $\Delta d_{R\rightarrow S}$) are fixed in all possible feature combinations. Besides, previously defined features in the literature are added in pairs to these combinations. After several simulations, the combinations which yield commonly better performances were selected and are listed in Table 5.

Regarding the results of the used methods, the best features combination is selected for each scenario (traffic density, attacker rate, and position falsification attack type). The results of this step are summarized in Table 6, in which the notation $i^{th}$ corresponds to the best combination ID present in Table 5. The analysis of these results indicates clearly that both of $2^{nd}$ and $5^{th}$ combinations are well suitable for high traffic density while there are still other good combinations for low and medium densities.

Since the attacker rate is not obvious to be estimated, one has to optimize the performances of the IDS regarding only the traffic density. For this purpose, we recommend using the combinations summarized in Table 7 that have shown the best performances depending on the traffic density.

The works from [40]–[44] were primarily evaluated and the best one was selected, which was finally the mechanism proposed in [44] for every scenario. This best related work was then compared with our best features' combinations shown in Table 7 that were directly derived from Table 6.

**TABLE 5.** Different feature combinations.

| | Combination ID | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **Features** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| $AoA$ | + | + | + | + | + | + | + | + | + |
| $\hat{d}$ | + | + | + | + | + | + | + | + | + |
| $\Delta d_{R\rightarrow S}$ | + | + | + | + | + | + | + | + | + |
| $\Delta P_{R\rightarrow S,a}$ | | | + | + | | | | | |
| $\Delta P_{S,a}$ | + | | + | | + | | | | |
| $\Delta S_{S,a}$ | | | | | | | + | | |
| $P_{S,a}$ | | | | | | + | | | |
| $\Delta S_{R\rightarrow S,a}$ | + | + | | | | | | | |
| $\Delta P_{R,a}$ | | + | | + | + | + | + | + | + |
| $\Delta S_{R,a}$ | | | | | | | | + | |
| $S_{S,a}$ | | | | | | | | | + |

**TABLE 6.** Selection of feature combination for each scenario.

| Traffic Density | Attacker Rate (%) | Attack ID 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|---|
| Low | 10 | $9^{th}$ | $5^{th}$ | $9^{th}$ | $3^{rd}$ | $8^{th}$ |
| Low | 20 | $5^{th}$ | $8^{th}$ | $9^{th}$ | $7^{th}$ | $5^{th}$ |
| Low | 30 | $5^{th}$ | $5^{th}$ | $9^{th}$ | $8^{th}$ | $8^{th}$ |
| Medium | 10 | $9^{th}$ | $6^{th}$ | $6^{th}$ | $7^{th}$ | $5^{th}$ |
| Medium | 20 | $5^{th}$ | $5^{th}$ | $7^{th}$ | $2^{nd}$ | $5^{th}$ |
| Medium | 30 | $5^{th}$ | $5^{th}$ | $7^{th}$ | $2^{nd}$ | $5^{th}$ |
| High | 10 | $5^{th}$ | $5^{th}$ | $2^{nd}$ | $2^{nd}$ | $2^{nd}$ |
| High | 20 | $2^{nd}$ | $2^{nd}$ | $2^{nd}$ | $5^{th}$ | $2^{nd}$ |
| High | 30 | $5^{th}$ | $5^{th}$ | $2^{nd}$ | $5^{th}$ | $2^{nd}$ |

Then, the average results of the three methods (kNN, RF, and Stacking) in terms of accuracy, F1-Score, and computation time are computed for each attack type at different traffic densities as shown in Table 8. The accuracy and F1-Score results vary between 81.2% and 92.5% for different scenarios. One can also infer that the computation time increases as the traffic density increases for each Attack ID.

Furthermore, the percentages of the average improvements comparing with the work [44] are given in Table 8 in parentheses. According to this table, it can be easily seen that the computation time is reduced in each case for our proposal. Computation time mainly depends on the number of used features for classification and the learning method as well as the size of the data. When comparing our mechanism with the one in [44], while using the same learning methods and the same datasets with different feature combinations, one can notice that our proposal provides better or similar accuracy and F1-score by considering fewer number of features, which requires considerably less computation time. This decrease is very significant and it impacts positively the learning phase, while reducing its duration and the consumed resources needed for this phase. Furthermore, the accuracy and F1-score for Attack IDs 2, 4, and 16 are increased at almost all traffic densities, when implementing our proposal.

In the case that the position falsification type is unknown, one can need to implement a feature combination at a certain traffic density in order to detect any kind of these attacks. The $5^{th}$ feature combination, which consists of using the features $AoA$, $\hat{d}$, $\Delta d_{R\rightarrow S}$, $\Delta P_{S,a}$, and $\Delta P_{R,a}$, is the most suitable one for this purpose and outperforms other combinations for different traffic densities. Since the position falsification attacks

**TABLE 7. Selection of feature combination for each density.**

| Traffic Density | Attack ID 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Low | $5^{th}$ | $5^{th}$ | $9^{th}$ | $8^{th}$ | $8^{th}$ |
| Medium | $5^{th}$ | $5^{th}$ | $7^{th}$ | $2^{nd}$ | $5^{th}$ |
| High | $5^{th}$ | $5^{th}$ | $2^{nd}$ | $5^{th}$ | $2^{nd}$ |

**TABLE 8. The average performances of the proposed mechanism using feature combinations in Table 7 and improvements comparing with [44].**

| Attack ID | Traffic Density | Accuracy | F1-Score | Computation Time |
|---|---|---|---|---|
| 1 | Low | 0.820 (-0.5) | 0.889 (-0.1) | 10.7 (46.4) |
| | Medium | 0.860 (0.4) | 0.910 (0.2) | 188.7 (41.0) |
| | High | 0.866 (-0.6) | 0.915 (-0.3) | 9390.5 (23.8) |
| 2 | Low | 0.825 (-0.1) | 0.890 (0.0) | 19.9 (27.8) |
| | Medium | 0.869 (0.7) | 0.916 (0.4) | 277.7 (30.7) |
| | High | 0.873 (-0.1) | 0.920 (0.0) | 9472.2 (26.7) |
| 4 | Low | 0.846 (1.7) | 0.909 (1.1) | 18.7 (40.6) |
| | Medium | 0.835 (1.2) | 0.893 (0.7) | 298.2 (33.6) |
| | High | 0.848 (1.4) | 0.903 (0.7) | 10728.4 (36.4) |
| 8 | Low | 0.812 (-1.2) | 0.888 (-0.5) | 19.6 (47.5) |
| | Medium | 0.852 (0.9) | 0.900 (0.6) | 325.0 (42.3) |
| | High | 0.852 (-0.5) | 0.907 (-0.4) | 9959.9 (31.1) |
| 16 | Low | 0.822 (0.5) | 0.886 (0.6) | 16.5 (40.9) |
| | Medium | 0.881 (1.6) | 0.925 (0.9) | 255.2 (43.9) |
| | High | 0.874 (0.9) | 0.920 (0.5) | 9948.2 (38.9) |

are examined here, it is reasonable that a combination with the features related to location and distance provides an efficient IDS to detect these attacks. Table 9 presents the average performances of the used methods for the $5^{th}$ feature combination and the percentages of the improvements (between parentheses) compared with the best related work [44]. When applying the $5^{th}$ combination to all the scenarios, one can see real improvements in accuracy, F1-score, and computation time compared with the best related work [44] for Attack IDs 2, 4, and 16. For Attack ID 1, despite that our proposal enhances the performances for the medium traffic density, it does not improve the accuracy and the F1-score at low and high traffic densities. The results for Attack ID 8 do not show a real improvement comparing with the work [44]. Nevertheless, these decreases in detection performances of Attack IDs 1 and 8 are less than 0.7% on average, which can be neglected regard to the huge improvement in computation time.

The improvements were made by our proposed detection mechanism compared with the best related work [44] due to the new proposed features for position falsification detection. In [44], both the delta position and delta speed of the sender and receiver were considered as well as the RSSI and the declared distance between sender and receiver. However, our proposal exploits the features related to the position rather than the speed. Considering not only the declared distance between sender and receiver but also the estimated distance ensures these improvements in detection of position falsification.

### C. DISCUSSION FOR RESULTS OF DETECTION METHODS

All the averaged results presented in the previous subsection were obtained using the methods kNN, RF, and Stacking in order to select the best feature combination for different traffic densities. On the other hand, it is obvious to compare the performances of these classification methods after selecting the $5^{th}$ combination since it shows the best results. Thus, the performances of every method in terms of accuracy and F1-Score are retrieved at each traffic density for different attack types. Note here that the presented results are averaged by the attacker rates (10%, 20%, and 30%) to avoid its impact.

**TABLE 9. The average performances of the proposed mechanism using the $5^{th}$ feature combination and improvements comparing with [44].**

| Attack ID | Traffic Density | Accuracy | F1-Score | Computation Time |
|---|---|---|---|---|
| 1 | Low | 0.820 (-0.5) | 0.889 (-0.1) | 10.7 (46.4) |
| | Medium | 0.860 (0.4) | 0.910 (0.2) | 188.7 (41.0) |
| | High | 0.866 (-0.6) | 0.915 (-0.3) | 9390.5 (23.8) |
| 2 | Low | 0.825 (-0.1) | 0.890 (0.0) | 19.9 (27.8) |
| | Medium | 0.869 (0.7) | 0.916 (0.4) | 277.7 (30.7) |
| | High | 0.873 (-0.1) | 0.920 (0.0) | 9472.2 (26.7) |
| 4 | Low | 0.840 (0.9) | 0.905 (0.6) | 16.1 (52.7) |
| | Medium | 0.830 (0.7) | 0.890 (0.4) | 263.3 (35.0) |
| | High | 0.840 (0.4) | 0.898 (0.3) | 10109.3 (31.1) |
| 8 | Low | 0.806 (-2.0) | 0.884 (-1.0) | 17.8 (56.7) |
| | Medium | 0.839 (-0.7) | 0.892 (-0.4) | 264.7 (45.6) |
| | High | 0.852 (-0.5) | 0.907 (-0.4) | 9959.9 (31.1) |
| 16 | Low | 0.821 (0.3) | 0.882 (0.1) | 12.3 (64.0) |
| | Medium | 0.881 (1.6) | 0.925 (0.9) | 255.2 (43.9) |
| | High | 0.868 (0.2) | 0.917 (0.1) | 9735.6 (38.5) |

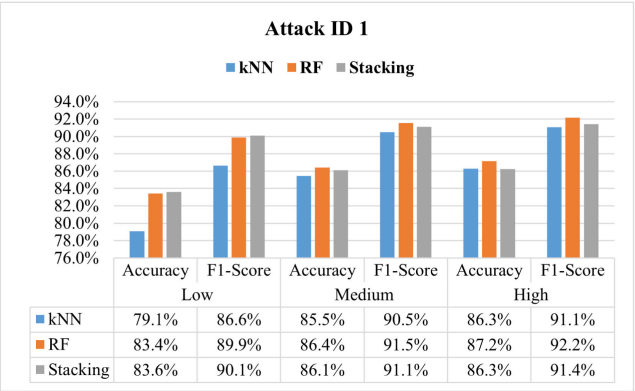
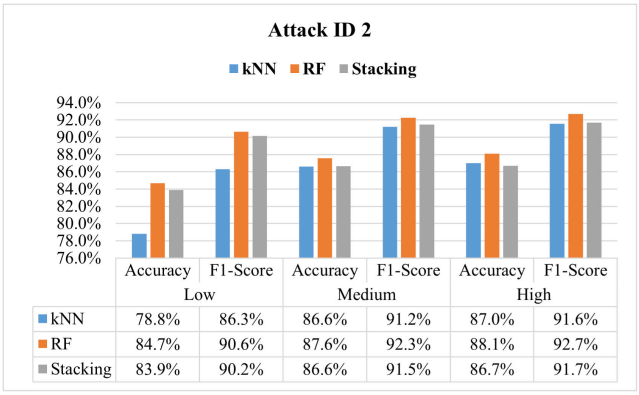



**FIGURE 9. Comparison of detection methods for Attack ID 1.**

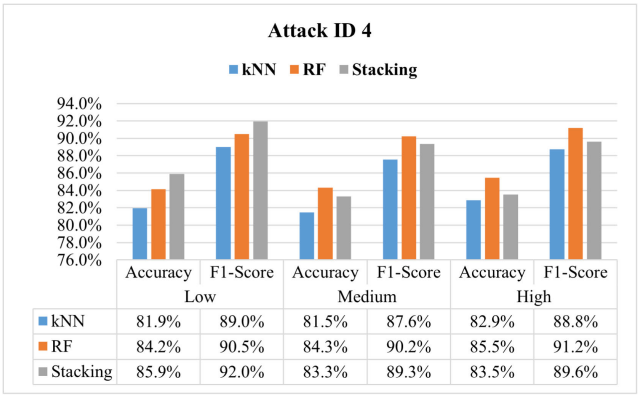
Fig. 9 represents the performances of the three classification methods (kNN, RF, and Stacking) for Attack ID 1 (Constant position) for different densities. It shows that Stacking method is the best one with 83.6% accuracy and 90.1% F1-score at the low traffic density to detect Attack ID 1, whereas RF is the best choice at the medium and high densities with better performances in both indicators. Moreover, the detection rate at the high density is the best compared to other densities for this type of attack, with an accuracy of 87.2% and an F1-score of 92.2%.

For Attack ID 2 (constant offset), performances, depicted in Fig. 10, show that RF is the best method for each traffic density. The accuracy rates are respectively 84.7%, 87.6%, and 88.1%, which indicate that the performance of the detection increases with traffic density. In addition, we achieved




| | Low | | Medium | | High | |
|---|---|---|---|---|---|---|
| | Accuracy | F1-Score | Accuracy | F1-Score | Accuracy | F1-Score |
| kNN | 78.8% | 86.3% | 86.6% | 91.2% | 87.0% | 91.6% |
| RF | 84.7% | 90.6% | 87.6% | 92.3% | 88.1% | 92.7% |
| Stacking | 83.9% | 90.2% | 86.6% | 91.5% | 86.7% | 91.7% |

**FIGURE 10.** Comparison of detection methods for Attack ID 2.




| | Low | | Medium | | High | |
|---|---|---|---|---|---|---|
| | Accuracy | F1-Score | Accuracy | F1-Score | Accuracy | F1-Score |
| kNN | 81.9% | 89.0% | 81.5% | 87.6% | 82.9% | 88.8% |
| RF | 84.2% | 90.5% | 84.3% | 90.2% | 85.5% | 91.2% |
| Stacking | 85.9% | 92.0% | 83.3% | 89.3% | 83.5% | 89.6% |

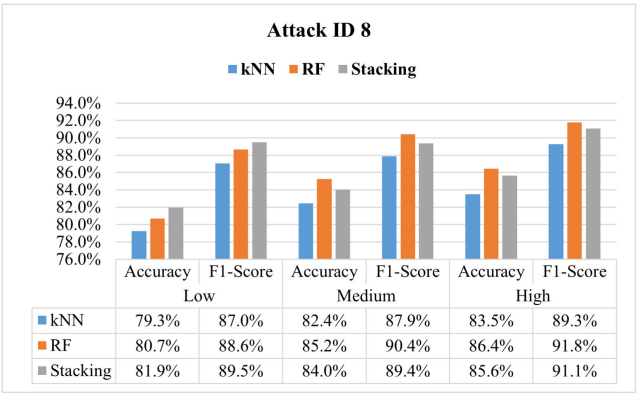
**FIGURE 11.** Comparison of detection methods for Attack ID 4.

the highest accuracy at the high density with Attack ID 2 comparing to other attack types (1, 4, 8, and 16).

Fig. 11 details the results of Attack ID 4 (random position), which show similar behavior than the ones obtained for Attack ID 1. The best method choices for all densities are the same than the Attack ID 1. Stacking can detect this attack with the highest accuracy, which is 85.9%, compared to all other attack types (1, 2, 8, and 16) at the low density. Despite this success, RF is the best method at the medium and high traffic densities with an accuracy of 84.3% and 85.5%, respectively.

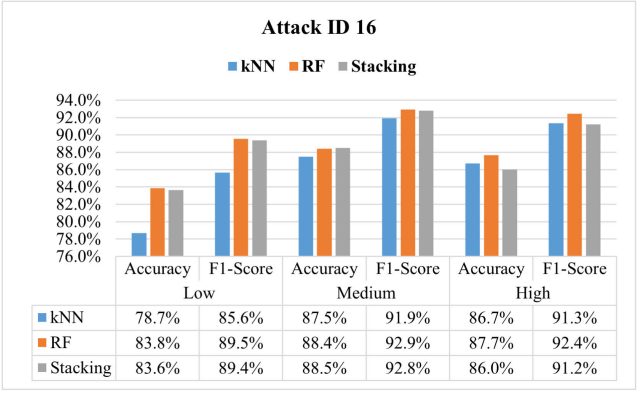
Fig. 12 shows that Stacking can be chosen at the low traffic density to detect Attack ID 8 (random offset) with an accuracy of 81.9%. However, it is the lowest rate compared to other attack types (1, 2, 4, and 16) at this density. Moreover, the results for random offset (ID 8) attack are close to the ones for random attack (ID 4) at the medium and high densities. This could be explained by the fact that these two types of attacks are similar in their way of attack launching. In fact, RF is the best detection method at these densities with an accuracy of 85.2% and 86.4%, respectively.

Finally, Fig. 13 illustrates the performances of each method for Attack ID 16 (eventual stop). RF is the best choice for all traffic densities for this type of attack. The accuracy and F1-score rates for the detection of Attack ID 16 are almost equal with RF and Stacking at the medium density. Thus, RF is chosen as well to be coherent with the other attack types.




| | Low | | Medium | | High | |
|---|---|---|---|---|---|---|
| | Accuracy | F1-Score | Accuracy | F1-Score | Accuracy | F1-Score |
| kNN | 79.3% | 87.0% | 82.4% | 87.9% | 83.5% | 89.3% |
| RF | 80.7% | 88.6% | 85.2% | 90.4% | 86.4% | 91.8% |
| Stacking | 81.9% | 89.5% | 84.0% | 89.4% | 85.6% | 91.1% |

**FIGURE 12.** Comparison of detection methods for Attack ID 8.




| | Low | | Medium | | High | |
|---|---|---|---|---|---|---|
| | Accuracy | F1-Score | Accuracy | F1-Score | Accuracy | F1-Score |
| kNN | 78.7% | 85.6% | 87.5% | 91.9% | 86.7% | 91.3% |
| RF | 83.8% | 89.5% | 88.4% | 92.9% | 87.7% | 92.4% |
| Stacking | 83.6% | 89.4% | 88.5% | 92.8% | 86.0% | 91.2% |

**FIGURE 13.** Comparison of detection methods for Attack ID 16.

**TABLE 10.** Decisions of Detection Method for each density.

| Traffic Density | Attack ID 1 | 2 | 4 | 8 | 16 |
|---|---|---|---|---|---|
| Low | Stacking | RF | Stacking | Stacking | RF |
| Medium | RF | RF | RF | RF | RF |
| High | RF | RF | RF | RF | RF |

Furthermore, the highest accuracy, compared to other attack types (1, 2, 4, and 8) at the medium density, is still obtained for this type of attack (88.4%).

Based on these results, the best methods for each traffic density and for each attack type are summarized in Table 10. It is seen that Stacking is preferred for Attack IDs 1, 4, and 8 in the low traffic density, whereas RF gives the best results for Attack IDs 2 and 16. However, RF is mainly suggested for each attack type at the medium and high densities.

In order to obtain a robust detection mechanism, a common detection method is proposed in this paper for each traffic density, regardless of the attack type. At the low traffic density, although Attack IDs 2 and 16 are detected with a higher performance using RF, Stacking can still be chosen as it produces close results to RF (see again Fig. 10 and Fig. 13). Consequently, the best detection method is Stacking at low traffic density and RF at medium and high traffic densities.

The suggested IDS mechanism for position falsification at different traffic densities is illustrated in Fig. 14. First, a vehicle has to check the level of traffic density. If it is located on a road with a low traffic density, Stacking is preferred

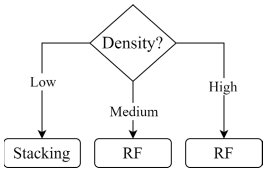



**FIGURE 14.** Proposed IDS for different traffic densities.

to detect the position falsification attack. Otherwise, RF is selected at the medium and high densities.

## V. CONCLUSION

In this paper, a distributed IDS, including new proposed features, is presented for VANETs to detect position falsification attacks. The well-known ML techniques, kNN and RF, and an ensemble learning by Stacking of kNN and RF are implemented to build a coherent mechanism. This learning step is launched in a central server. Then, the IDS scheme is deployed on the vehicles to detect distributively the attackers. Each scenario for different traffic densities, attacker rates, and attack types are analyzed separately using these three detection methods. By this comprehensive analysis, a generalized detection mechanism is constructed for position falsification attacks in VANETs.

The new proposed features to detect misbehavior are the angle of arrival between sender and receiver, the estimated distance calculated using the RSSI, and the difference of the declared and estimated distance between sender and receiver. Previously used common features are considered as well as these proposed ones. The features to detect each attack type in each traffic density are obtained. The proposed mechanism gives overall better results than the best previous mechanisms. Therefore, a common feature combination for different attack types can have significant performances. In such a situation, this feature combination is proposed in this paper for different traffic densities. It is coherent since it includes location-related features, i.e. the angle of arrival, the estimated distance, the difference of declared and estimated distance between the sender and the receiver, the delta position of sender, and the delta position of receiver.

Furthermore, the proposed detection methods were evaluated separately to obtain the most suitable method as well as the best feature combination. Then, a common detection method is suggested in each traffic density for any attack type: Stacking at the low density, RF at the medium and high densities.

Although the proposed scheme includes an efficient detection mechanism, it does not suggest any reaction mechanism for vehicles. Thus, to avoid this limitation, the work can be expanded with a suitable reaction mechanism as well (for example sending notifications to other vehicles, passive or active revocation of attackers, etc.).

Different other types of attacks, than the position falsification (e.g. DoS, DDoS, Sybil, black-hole attacks, etc.), could be also considered. A detection mechanism for a multi-class scheme that detects not only the attack but its type in the same algorithm can be developed.

## ACKNOWLEDGMENT

This work was supported by the C-Roads France project funded by the European Commission from the CINEA Agency within the 2015 CEF Transport Programme under Grant 2015-FRTM-0378-S. The statements made herein are solely the responsibility of the authors.

## REFERENCES

[1] M. C. Bragagnolo, N. Messai, and N. Manamanni, "Attack detection in a cluster divided consensus network," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 1091–1096.
[2] X. Ya, Z. Shihui, and S. Bin, "Trusted GPSR protocol without reputation faking in VANET," *J. China Univ. Posts Telecommun.*, vol. 22, no. 5, pp. 22–55, Oct. 2015.
[3] X. Yao, X. Zhang, H. Ning, and P. Li, "Using trust model to ensure reliable data acquisition in VANETs," *Ad Hoc Netw.*, vol. 55, pp. 107–118, Feb. 2017.
[4] K. Guibene, M. Ayaida, L. Khoukhi, and N. Messai, "Black-box system identification of CPS protected by a watermark-based detector," in *Proc. IEEE 45th Conf. Local Comput. Netw. (LCN)*, Nov. 2020, pp. 341–344.
[5] C. Kruegel and T. Toth, "Using decision trees to improve signature-based intrusion detection," in *Recent Advances in Intrusion Detection*, G. Vigna, C. Kruegel, and E. Jonsson, Eds. Berlin, Germany: Springer, 2003, pp. 173–191.
[6] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV," *Ad Hoc Netw.*, vol. 61, pp. 33–50, Jun. 2017.
[7] B. Subba, S. Biswas, and S. Karmakar, "A game theory based multi layered intrusion detection framework for VANET," *Future Gener. Comput. Syst.*, vol. 82, pp. 12–28, May 2018.
[8] H. Amirat, N. Lagraa, C. A. Kerrach, and Y. Ouinten, "Fuzzy clustering for misbehaviour detection in VANET," in *Proc. Int. Conf. Smart Commun. Netw. Technol. (SaCoNeT)*, Oct. 2018, pp. 200–204.
[9] M. Ayaida, N. Messai, G. Wilhelm, and S. Najeh, "A novel Sybil attack detection mechanism for C-ITS," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 913–918.
[10] M. Ayaida, N. Messai, S. Najeh, and K. B. Ndjore, "A macroscopic traffic model-based approach for Sybil attack detection in VANETs," *Ad Hoc Netw.*, vol. 90, pp. 1–12, Jul. 2019.
[11] A. Luckshetty, S. Dontal, S. Tangade, and S. S. Manvi, "A survey: Comparative study of applications, attacks, security and privacy in VANETs," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2016, pp. 1594–1598.
[12] M. A. Al-Shareeda, M. Anbar, I. H. Hasbullah, and S. Manickam, "Survey of authentication and privacy schemes in vehicular ad hoc networks," *IEEE Sensors J.*, vol. 21, no. 2, pp. 2422–2433, Jan. 2021.
[13] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Veh. Commun.*, vol. 1, no. 2, pp. 53–66, 2014.
[14] J. Zhang and Y. Xu, "Privacy-preserving authentication protocols with efficient verification in VANETs," *Int. J. Commun. Syst.*, vol. 27, no. 12, pp. 3676–3692, 2014.
[15] A. K. Malhi and S. Batra, "Decision inference system for misbehavior detection in VANETs," in *Proc. 2nd Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2015, pp. 1558–1563.
[16] A. S. Lal and R. Nair, "Region authority based collaborative scheme to detect Sybil attacks in VANET," in *Proc. Int. Conf. Control Commun. Comput. India (ICCC)*, Nov. 2015, pp. 664–668.
[17] N. Bißmeyer, J. Njeukam, J. Petit, and K. M. Bayarou, "Central misbehavior evaluation for VANETs based on mobility data plausibility," in *Proc. 9th ACM Int. Workshop Veh. Inter-Netw., Syst., Appl. (VANET)*, 2012, pp. 73–82.
[18] T. Bouali, S.-M. Senouci, and H. Sedjelmaci, "A distributed detection and prevention scheme from malicious nodes in vehicular networks," *Int. J. Commun. Syst.*, vol. 29, no. 10, pp. 1683–1704, 2016.
[19] U. Khan, S. Agrawal, and S. Silakari, "Detection of malicious nodes (DMN) in vehicular ad-hoc networks," *Proc. Comput. Sci.*, vol. 46, pp. 965–972, Jan. 2015.
[20] F. A. Ghaleb, M. A. Maarof, A. Zainal, M. A. Rassam, F. Saeed, and M. Alsaedi, "Context-aware data-centric misbehaviour detection scheme for vehicular ad hoc networks using sequential analysis of the temporal and spatial correlation of the consistency between the cooperative awareness messages," *Veh. Commun.*, vol. 20, Dec. 2019, Art. no. 100186.

[21] H. Zhang, A. Bochem, X. Sun, and D. Hogrefe, "A security aware fuzzy enhanced reliable ant colony optimization routing in vehicular ad hoc networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1071–1078.
[22] Y. Yao, B. Xiao, G. Wu, X. Liu, Z. Yu, K. Zhang, and X. Zhou, "Voiceprint: A novel Sybil attack detection method based on RSSI for VANETs," in *Proc. 47th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*. Denver, CO, USA: IEEE, Jun. 2017, pp. 591–602.
[23] C. Chen, X. Wang, W. Han, and B. Zang, "A robust detection of the Sybil attack in urban VANETs," in *Proc. 29th IEEE Int. Conf. Distrib. Comput. Syst. Workshops*, Jun. 2009, pp. 270–276.
[24] R. W. van der Heijden, T. Lukaseder, and F. Kargl, "VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs," in *Security and Privacy in Communication Networks*, R. Beyah, B. Chang, Y. Li, and S. Zhu, Eds. Cham, Switzerland: Springer, 2018, pp. 318–337.
[25] L. A. Maglaras, "A novel distributed intrusion detection system for vehicular ad hoc networks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 6, no. 4, pp. 3676–3692, 2015.
[26] M. Zang and Y. Yan, "Machine learning-based intrusion detection system for big data analytics in VANET," in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Apr. 2021, pp. 1–5.
[27] C. Zhang, K. Chen, X. Zeng, and X. Xue, "Misbehavior detection based on support vector machine and Dempster–Shafer theory of evidence in VANETs," *IEEE Access*, vol. 6, pp. 59860–59870, 2018.
[28] J. Kamel, I. B. Jemaa, A. Kaiser, L. Cantat, and P. Urien, "Misbehavior detection in C-ITS: A comparative approach of local detection mechanisms," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2019, pp. 1–8.
[29] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6631–6643, Jun. 2020.
[30] 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*, A. Laouiti, A. Qayyum, and M. N. M. Saad, Eds. Singapore: Springer, 2020, pp. 73–86.
[31] S. Sharma and A. Kaul, "Hybrid fuzzy multi-criteria decision making based multi cluster head dolphin swarm optimized IDS for VANET," *Veh. Commun.*, vol. 12, pp. 23–38, Apr. 2018.
[32] O. A. Wahab, A. Mourad, H. Otrok, and J. Bentahar, "CEAP: SVM-based intelligent detection model for clustered vehicular ad hoc networks," *Expert Syst. Appl.*, vol. 50, pp. 40–54, May 2016.
[33] D. Tian, Y. Wang, G. Lu, and G. Yu, "A vehicular ad hoc networks intrusion detection system based on BUSNet," in *Proc. 2nd Int. Conf. Future Comput. Commun.*, vol. 1, 2010, pp. 225–229.
[34] F. A. Ghaleb, A. Zainal, M. A. Rassam, and F. Mohammed, "An effective misbehavior detection model using artificial neural network for vehicular ad hoc network applications," in *Proc. IEEE Conf. Appl., Inf. Netw. Secur. (AINS)*, Nov. 2017, pp. 13–18.
[35] F. A. Ghaleb, M. A. Maarof, A. Zainal, B. A. S. Al-Rimy, F. Saeed, and T. Al-Hadhrami, "Hybrid and multifaceted context-aware misbehavior detection model for vehicular ad hoc network," *IEEE Access*, vol. 7, pp. 159119–159140, 2019.
[36] F. A. Ghaleb, M. A. Maarof, A. Zainal, B. A. S. Al-Rimy, A. Alsaeedi, and W. Boulila, "Ensemble-based hybrid context-aware misbehavior detection model for vehicular ad hoc network," *Remote Sens.*, vol. 11, no. 23, p. 2852, Dec. 2019.
[37] E. A. Shams, A. Rizaner, and A. H. Ulusoy, "Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks," *Comput. Secur.*, vol. 78, pp. 245–254, Jul. 2018.
[38] M. Kim, I. Jang, S. Choo, J. Koo, and S. Pack, "Collaborative security attack detection in software-defined vehicular networks," in *Proc. 19th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2017, pp. 19–24.
[39] F. A. Ghaleb, F. Saeed, M. Al-Sarem, B. A. S. Al-Rimy, W. Boulila, A. E. M. Eljialy, K. Aloufi, and M. Alazab, "Misbehavior-aware on-demand collaborative intrusion detection system using distributed ensemble learning for VANET," *Electronics*, vol. 9, no. 9, p. 1411, Sep. 2020.
[40] S. So, J. Petit, and D. Starobinski, "Physical layer plausibility checks for misbehavior detection in V2X networks," in *Proc. 12th Conf. Secur. Privacy Wireless Mobile Netw. (WiSec)*, May 2019, pp. 84–93.
[41] P. K. Singh, S. Gupta, R. Vashistha, S. K. Nandi, and S. Nandi, "Machine learning based approach to detect position falsification attack in VANETs," in *Security and Privacy*, S. Nandi, D. Jinwala, V. Singh, V. Laxmi, M. S. Gaur, and P. Faruki, Eds. Singapore: Springer, 2019, pp. 166–178.
[42] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in VANET," in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 564–571.
[43] P. Sharma, D. Austin, and H. Liu, "Attacks on machine learning: Adversarial examples in connected and autonomous vehicles," in *Proc. IEEE Int. Symp. Technol. Homeland Secur. (HST)*, Nov. 2019, pp. 1–7.
[44] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
[45] J. Zheng, C. Wu, H. Chu, and Y. Xu, "An improved RSSI measurement in wireless sensor networks," *Proc. Eng.*, vol. 15, pp. 876–880, Jan. 2011.
[46] S. Singh, R. Sulthana, T. Shewale, V. Chamola, A. Benslimane, and B. Sikdar, "Machine learning assisted security and privacy provisioning for edge computing: A survey," *IEEE Internet Things J.*, early access, Jul. 19, 2021, doi: 10.1109/JIOT.2021.3098051.
[47] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Comput. Secur.*, vol. 65, pp. 135–152, Mar. 2017.
[48] I. S. Atawodi, "A machine learning approach to network intrusion detection system using $K$ nearest neighbor and random forest," M.S. thesis, School Comput. Sci. Comput. Eng., Univ. Southern Mississippi, Hattiesburg, MS, USA, 2019.
[49] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[50] X. Xu, J. Li, Y. Yang, and F. Shen, "Toward effective intrusion detection using log-cosh conditional variational autoencoder," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6187–6196, Apr. 2021, doi: 10.1109/JIOT.2020.3034621.

**SECIL ERCAN** received the Ph.D. degree in industrial engineering from Istanbul Technical University, Turkey, in 2017. She has studied stochastic scheduling of energy systems in her Ph.D. thesis.

She is currently a Research Engineer with the University of Reims Champagne-Ardenne, France. Her current research interests include optimization, mathematical modeling, data science, and artificial intelligence in intelligent transport systems.

**MARWANE AYAIDA** received the Diploma degree in electronics of embedded systems from the High National School in Electronics and their Applications (E.N.S.E.A.), Cergy-Pontoise, France, in 2009, the master's degree in electronics of autonomous systems from the University of Cergy-Pontoise, France, in 2009, and the Ph.D. degree from the University of Reims Champagne-Ardennes, France, in 2012. His research interests include interoperability modeling for embedded systems in the field of transportation. Also, his work focuses on vehicular communications (V2V and V2I). Specifically, he studies the routing protocols in Vehicular Ad-hoc Networks (VANETs). He has been an Associate Professor with the University of Reims, since September 2013.

**NADHIR MESSAI** (Member, IEEE) received the M.S. and the Ph.D. degrees from the University of Technology of BelfortMontbliard, France, in 2000 and 2003, respectively, all in automatic control and computer engineering.

He is currently an Associate Professor with the University of Reims Champagne-Ardenne (URCA), France. His research interests include safety and the security of cyber physical systems with application to manufacturing and intelligent transportation systems.