An Enhanced Model for Machine Learning-Based DoS Detection in Vehicular Networks

Secil Ercan*, Leo Mendiboure* (Member, IEEE), Lylia Alouache†, Sassi Maaloul*‡, Tidiane Sylla*§, Hasnaa Aniss*
* COSYS-ERENA Lab, Université Gustave Eiffel, 33400 Talence, France
† ETIS UMR8051, CY Cergy Paris University, ENSEA, CNRS, F-95000, Cergy, France
‡ Computer Sciences & Mathematics, JUNIA Engineering School, 59000 Lille, France
§ GEII-ISA, University of Sciences, Techniques and Technologies of Bamako, Bamako, Mali
E-mail: secil.ercan@univ-eiffel.fr, leo.mendiboure@univ-eiffel.fr, lylia.alouache@cyu.fr, sassi.maaloul2@univ-eiffel.fr,
tidiane.sylla@univ-eiffel.fr, hasnaa.aniss@univ-eiffel.fr

***Abstract*—Vehicular communication networks should play an important role in deploying future automated and connected vehicles. Indeed, these vehicular networks could exchange information (position, speed, obstacle detection, slowing down, etc.) that could improve road safety and traffic efficiency. Therefore, it is essential to ensure the cybersecurity of these communication systems to prevent malicious entities from disrupting them. This is why, in this paper, we focus on one of the most common types of attacks in the vehicular environment: Denial-of-Service (DoS) attacks that impact the availability of services. The existing algorithms for DoS attacks detection, mainly based on Artificial Intelligence tools (Machine Learning, Deep Learning), only consider a limited number of features to build their models (position, speed). Therefore, in this paper, we quickly compare state-of-the-art approaches and introduce a new Machine Learning model considering a larger number of features and aiming at guaranteeing better performances for DoS attacks detection. We also propose an implementation and a comparative analysis of existing models to demonstrate the benefits of our approach both in terms of accuracy and F1-score.**

***Index Terms*—C-ITS, VANET, Intrusion Detection Systems, DoS Attacks, Artificial Intelligence, Machine learning**

I. Introduction

Vehicular communication networks and Cooperative Intelligent Transportation Systems (C-ITS) will play a significant role in the advent of future Connected and Automated Vehicles (CAVs) [1], [2]. Indeed, C-ITS, relying on different underlying Radio Access Networks (4G/5G, ITS-G5, etc.) [3], will allow vehicles and roadside infrastructure to exchange a set of information that could improve both road safety and traffic efficiency: vehicle position and speed, lane changes, obstacle detection, emergency braking, etc.

However, these vehicular networks could be vulnerable to many attacks [4], [5] that can affect both 1) data confidentiality (eavesdropping), 2) data integrity (timing attack), 3) data availability (Denial-of-Service (DoS)/Distributed DoS (DDoS), 4) Broadcast Tampering, Malware, Block Hole, etc.), 5) users authentication/identification (GPS Spoofing, Tunneling, Position Faking, etc.) and 6) users privacy (Identity Revealing). The consequences of such attacks could, in the worst case, lead to an accident and the endangerment of road users, especially for attacks related to data integrity or availability.

To counter these attacks and guarantee vehicular network security, an approach commonly used in both research and industry is the implementation of Intrusion Detection Systems (IDS) [6]. These systems are generally classified into two main categories [7]: signature-based and anomaly-based detection. They are used to identify abnormal/suspicious activities. They, therefore, represent an essential first step in implementing a response to attacks, and the performance level of these IDSs must be high.

DoS attack is probably the most well-known attack on service availability, particularly in vehicular networks [8], [9]. This attack is simple to implement by attackers (they could send many messages for example) and can lead to the unavailability of the C-ITS services. The consequences of such DoS attacks could therefore be significant. That is why many approaches based on anomaly detection have already been proposed in the literature for DoS attacks detection in vehicular networks. These solutions, such as those described in [10]–[12], are based on using Artificial Intelligence (AI) tools [13]: Machine Learning, Deep Learning and Federated Learning. However, they only take into account a limited number of features (position, speed, acceleration, heading), which may reduce their level of performance and limit the ability of IDS to detect DoS attacks.

Therefore, in this paper, we propose to introduce a new Machine Learning model for detecting DoS attacks in vehicular networks. The objective of the proposed solution is to guarantee a higher level of performance than existing solutions and to reinforce security in vehicular networks. To independently evaluate the performance level of the proposed method and the selected features, we will implement our detection approach both with the original features commonly used in the literature and with our proposed features. The main contributions of this paper are:

- To introduce a suitable feature combination to detect DoS attacks using well-known Machine Learning (ML) techniques;

ISBN 978-3-903176-57-7©2023 IFIP

- To provide an Open Source implementation of the proposed solution (for data preparation and ML part)[1];
- To compare the performance level of the approach proposed in this paper with the solutions already introduced in the literature using an Open Dataset called VeReMi.

The rest of this paper is organized as follows. Section 2 compares state-of-the-art solutions for misbehavior detection in vehicular networks. Then, Section 3 presents a new Machine Learning model for DoS attacks detection. Finally, in Section 5, the performances of our system are compared to existing solutions using an Open Source Dataset.

II. Related Work

This section presents existing works related to the detection of DoS attacks. Their limitations are also identified to justify the need to propose a new model.

A. Common solutions for DoS attack detection

In a wider context than vehicular communication networks, DoS attacks have been considered in the wired and wireless networks literature. As a result, many Machine Learning and Deep Learning methods have been applied to detect these DoS attacks [14]–[20]. We can mention in particular: Support Vector Machines (SVM), Logistic Regression (LR), k-Nearest Neighbor (kNN), Decision Tree (DT), Random Forest (RF), Extreme Gradient Boosting (XGBoost), Neural Network (NN)) and deep learning methods (Long-Short Term Memory (LSTM), Deep Neural Network (DNN), Convolutional Neural Network (CNN), Multilayer Perceptron (MLP)).

In these contexts, the basic feature used in Machine Learning models can contain different types of information related to the TCP/IP connection, such as TCP flow features that could include a large set of parameters: data transmission time, average window advertisement [14], network addresses range, packet rate [15], [16], number of bytes/packets sent by the source/destination, duration of a bidirectional flow, packet size, inter-packet interval, protocol, bandwidth, IP destination address cardinality and novelty [17], destination port, packet length mean, number of backward packets per second, maximum packet length [18], source/destination IP, source/destination ports, and provide a specific taxonomy for each non-bonafide traffic [19], flow duration, packet length mean [20], etc. Since traffic networks have a lot of features, feature reduction or feature selection is a common approach in attacks detection.

B. Solutions specifically designed for DoS attack detection in vehicular networks

In the specific context of this paper, which focuses on vehicular communication networks, in addition to the features mentioned above, various features based on vehicle information have been studied to detect DoS attacks in the vehicular environment. In particular, the authors of [21] have proposed an attacked packet detection algorithm using frequency and speed. Similarly, the authors of [22] considered the distance, number of packets sent, and Packet Delivery Ratio (PDR) as features and applied a PDR metric-based detection algorithm. Finally, in [23], a statistical model was applied (median absolute deviation) to detect attackers with neighboring vehicle information such as time, MAC address and IP address. However, the detection algorithms used in these studies are not ML-based. Therefore, as noted by the authors of [10], it seems possible to improve the performance level of these approaches by employing new AI-based methods.

Some other papers have already worked on detecting DoS attacks in vehicular networks using Machine Learning methods, for example, in [10]–[12], [24], but these papers considered a limited number of features. In [10], a sampling technique was proposed to detect DoS attacks in vehicular networks considering the basic features of network connections, features of network packets and flows, etc. This solution was compared with other sampling techniques (in particular: oversampling and undersampling) on a public dataset. The authors of this paper used well-known Machine Learning and Deep Learning classification methods Random Forest, Support Vector Machines, XGBoost and Long-Short Term Memory. Similarly, the authors of [11] also used sampling techniques for a balanced dataset before implementing different machine learning methods to detect nine types of attacks, including DoS: Logistic Regression, Naive Bayes, Decision Tree, Random Forest, k-Nearest Neighbor, Support Vector Machines, AdaBoost, XGBoost. We can note that features related to the network state, such as source and destination IP, were considered in the learning phase of this study. The authors of [24] utilized similar ML methods, i.e. k-Nearest Neighbor, Decision Tree, AdaBoost, Random Forest, but then combined their results by implementing Logistic Regression to detect DoS, Sybil, replay, and distributed DoS attacks. In addition, they proposed some plausibility checks (communication range, position, speed) and consistency checks (position, intersection), which require area calculations. Finally, in [12], the authors applied an unsupervised learning method, k-Means, with the following features: the relative speed between the sender and receiver, received signal strength and interference (RSSI), PDR, and signal-to-interference-plus-noise-ratio (SINR).

Thus, different solutions based on AI [10]–[12] or statistical [21]–[23] tools have already been proposed in the literature for the detection of DoS attacks in vehicular networks. However, speed-related and position-related features could be used to enhance the performance of DoS detection and have not been considered yet. Similarly, information regarding the relationship between the sender and the receiver is also an important element that could be useful for detecting DoS attacks. Therefore, it would be necessary to define new solutions considering these features to explain ever more effective attack detection systems.

III. Proposed Detection Approach

In this section, after a detailed presentation of the context of attacks considered, we describe our proposed solution for

[1] https://github.com/ercansec/AttackDetectionMachineLearning

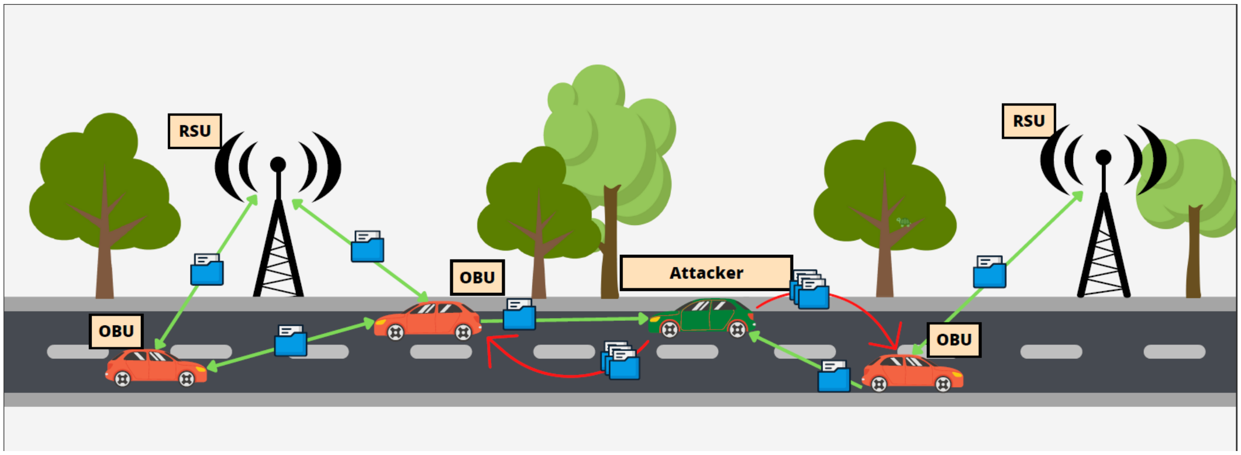

Fig. 1. A basic system for vehicle communication with DoS attack.

DoS attacks detection in vehicular networks. It consists of three main elements: 1) a selection of features adapted to the vehicular environment and DoS attacks, 2) a presentation of the considered data sampling approach and 3) the Machine Learning method used for the attack detection.

A. Attack scenarios considered

During a DoS attack, the attacker will emit a large number of messages (possibly several times the same one) to block, thanks to this high frequency of messages, the target's communication/processing system and thus make the service inaccessible.
Figure 1 proposes an example of a simple DoS attack in vehicular networks. RoadSide Units (RSUs) and On-Board Units (OBUs) communicate regularly through I2V (infrastructure-to-vehicle) and V2I (vehicle-to-infrastructure) communications. In addition, the OBUs can also communicate with each other through V2V (vehicle-to-vehicle) communications. Green arrows denote regular communication between entities with normal behaviour in this figure. In contrast, red ones correspond to DoS attacks with a high frequency of messages intended to disrupt the operation of nearby vehicles.

DoS attacks are often associated with Sybil attacks [25]. In a Sybil attack, an attacker attempts to generate false identities (pseudonyms) and have them accepted by other vehicles. For example, in a Sybil attack, an attacker may pretend that it is communication on behalf of X vehicles (X-1 false identities emitting messages, for example, Cooperative Awareness Messages [26]) and not just on its behalf.

To define our detection approach, we considered four types of DoS attacks, some of them combined with Sybil attacks:

- DoS Random attack: the content of the messages generated and sent by the attacker is random;
- DoS Disruptive attack: the attacker retransmits messages received from a set of neighboring vehicles/entities by replacing their signatures with its own;
- DoS Random Sybil: messages sent by the attacker are signed with random identities (pseudonyms) that he has generated himself;
- DoS Disruptive Sybil: messages sent by the attacker are signed with the identities (pseudonyms) of neighboring vehicles;

B. Feature Selection

Feature selection plays an important role in ML algorithms since it is necessary to use the critical features and avoid the features that can cause overfitting problems. The problem, DoS detection in our case, should be analyzed to extract the most convenient features. Thus, we selected the features according to the requirements of DoS attack detection in VANETs. Detection of an attacker depends on not only the information related to the sender but also the information of the relation between sender and receiver.

The sender position, speed, acceleration, and heading values are already obtained in messages sent by vehicles during V2V communication. The sending time is used to calculate the time difference between two consecutive messages, which is essential to detect DoS-related attacks since several messages are sent to block the communication. The number of packets sent [22] is basically counted for each sender. The frequency depends on sender speed [21] is determined by the absolute value of the difference between the sender speed and the half of maximum speed.

As well as these features, the features corresponding to the relation between sender and receiver are considered because their relative position and speed are also significant to analyze a DoS-related attack which repeats previously received messages (by using its own signature) or randomly generated messages. The distance between the sender and receiver is computed by using the Euclidean distance metric. Then, the estimated angle of arrival (AoA) [27] is calculated by the arctangent function of the position difference in the y-axis over the position difference in the x-axis. Finally, we obtain the speed difference by the Euclidean norm of the speed difference between the sender and receiver.

Considering the attack detection algorithms in the literature and the properties of the DoS attack, the following features are selected to be implemented in the proposed detection mechanism: 1) the frequency depending on sender speed (in x and y axis), 2) the number of packets sent, 3) the time difference between two consecutive messages, 4) the sender position (in x and y axis), 5) the sender speed (in x and y axis), 6) the sender acceleration (in x and y axis), 7) the sender heading (in x and y axis), 8) the distance between sender and receiver, 9) the estimated AoA between sender and receiver and 10) the difference of speed between sender and receiver.

C. Data Sampling

In classification problems, the class of a new sample is determined among a given set of labels/classes. The numbers of samples in each class generally vary, and the dataset becomes imbalanced. This can cause a biased classification [28], so the different numbers of data for each class in the training phase can affect the performance of the algorithm [29]. Hence, it is recommended to use data sampling techniques to obtain a balanced dataset in classification problems.

If the data for different classes have a different size in classification problems, two main solutions can be used to enable that:

- Oversampling: It proposes sampling where data are sampled based on the given data to increase the number of class data with lower numbers randomly. Hence, the numbers of data for both classes become equivalent the class with a higher number;
- Undersampling: It implies the inverse by randomly deleting some data from the class with a higher number to balance the numbers of data.

There are other types of sampling techniques, e.g. synthetic minority oversampling technique, near-miss etc., which benefit from these two main solutions [29].

In our case, the number of normal vehicles almost doubles the number of attackers in the targeted dataset. Thus, these two basic sampling approaches will be applied to the proposed method on the proposed dataset with the selected features to obtain a balanced dataset.

D. Detection Techniques

Ensemble Learning (EL) is a classification method that combines ML classification techniques to improve the training process. Three main EL approaches can be considered: 1) bagging, 2) boosting, and 3) stacking [30]. With boosting, a base learning method is used homogeneously and sequentially, whereas a homogeneous base learning method is used parallel in bagging. Stacking uses different base learning methods.

The model is trained on independent sub-datasets in bagging [31]. Hence, it prevents overfitting issues and classifies the data based on the maximum number of votes. These sub-datasets are randomly chosen with replacements excluding approximately one-third of the training dataset in each split.

On the other hand, boosting [32] provides an iterative algorithm by modifying the weights of the base classifier at each iteration according to its classification results. The objective is to reduce the error by forcing the misclassifying sub-datasets to improve their performances. The final iteration thus includes the possible best classification. However, it can cause overfitting issues, and some boosting techniques handle them (e.g. XGBoosting).

Stacking [33] allows the combination of heterogeneous learning techniques to enhance the classification's performance. 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. Logistic Regression (LR), Generalized Boosted Model (GBM) and Generalized Linear Model (GLM) are two examples of the methods used to combine the results in the Stacking method.

These three EL approaches have different advantages. Boosting decreases the bias since it focuses on correcting the errors made in the previous models of the sequential process. Bagging, contrarily, decreases the variance by implementing randomly chosen subsets with replacement in parallel. Hence, it deals with the over-fitting problem thanks to its subset procedure. Finally, stacking takes advantage of different algorithms, such as decreasing the bias and the variance, to solve underfitting and overfitting problems, respectively. Furthermore, it combines the results of different algorithms by implementing another classifier, i.e. a meta-classifier, to provide a better classification.

Therefore, we propose to use a decision tree as a base learner and then implementing it in boosting with XGBoost method and bagging with Random Forest.

Boosting algorithms define an objective function, which is a loss function to be minimized. This function mainly includes a part for the sum of errors and a part of regularization by adding a penalty term. XGBoost proposes more robust regularization techniques (namely, L1 and L2 for classification) to prevent the overfitting problem in boosting algorithms [34]. L1 uses the sum of the absolute value of weights, whereas L2 uses the sum of the square of weights. Moreover, XGBoost uses parallel processing within a tree in each iteration, providing a faster algorithm [35].

In RF as a type of bagging approach [36], the features are randomly selected for the training of the sub-datasets. The randomness aims to reduce the correlation between the used features.

Finally, we combine the results of boosting (XGBoost) and bagging (RF) approaches by stacking them with LR for generalization.

Figure 2 shows the main workflow of the proposed approach for the detection method, the dataset with new features and with data sampling techniques.

IV. Evaluation

In this section, we present the evaluation that was performed to demonstrate the level of performance of the proposed solution for detecting DoS attacks in vehicular networks. We

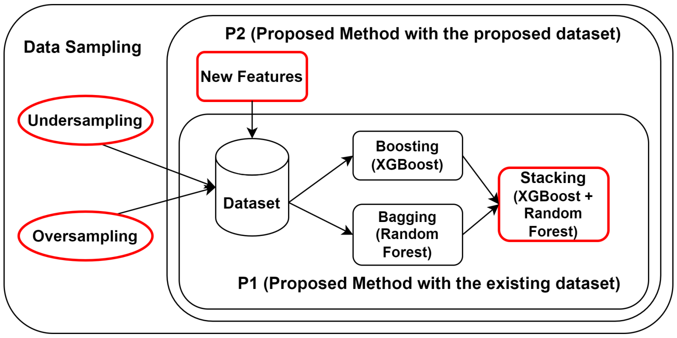
Data Sampling
Undersampling
Oversampling
P2 (Proposed Method with the proposed dataset)
New Features
Dataset
Boosting (XGBoost)
Bagging (Random Forest)
Stacking (XGBoost + Random Forest)
P1 (Proposed Method with the existing dataset)

Fig. 2. The proposed methodology to detect DoS attack.

describe the dataset used, the experimentation environment considered, the indicators used, and the results obtained.

A. Dataset Description

The VeReMi dataset, which allows for evaluating misbehavior detection approaches, was used in this work as a public Dataset. The authors of [37] performed the simulations with the Luxembourg traffic scenario (LuST) for realistic mobility patterns in a city and used Veins [38], based on OMNET++ [39] and Simulation of Urban MObility (SUMO) [40], for the vehicular network simulations. The combination of this realistic scenario and this tool allowed them to obtain the results available in the VeReMi database. The attacks defined here are related to position falsification. Reference [41] introduces an extension of the VeReMi dataset by adding new attack types and enlarging the data points.

In this extended database, there are different types of attacks, i.e. constant position, constant position offset, constant speed, constant speed offset, data replay, data replay sybil, delayed messages, disruptive, DoS (denial-of-service), DoS disruptive, DoS disruptive sybil, DoS random, DoS random sybil, eventual stop, grid sybil, mix all, random position, random position offset, random speed, random speed offset. The attacks related to DoS are the ones that are considered in this study.

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

If the message type is BSM, the following values are obtained for that BSM: Receiving time, position vector and position noise vector, speed vector and speed noise vector, sending time, sender, and message ID. On the other hand, the GPS type contains the values of time, sender, attacker type, message ID, position vector and position noise vector, speed vector and speed noise vector, acceleration vector and acceleration noise vector, header vector and header noise vector.

We compare our proposal with the results of the extended version of this dataset [41] by introducing new features and a different detection approach.

B. Environment

For each type of attack, two scenarios simulated at different hours, i.e. 07:00-09:00 and 14:00-16:00, are presented. According to these scenarios, traffic density is 37.03 and 16.36 V/km^2, the number of attackers is 1220 and 505, the number of normal vehicles is 2846 and 1179 for high and low traffic density, respectively [41]. The other information for the scenarios can be seen in Table I.

TABLE I
INFORMATION FOR SCENARIOS OF DATASET

	Scenario 1	Scenario 2
Hours for simulation	07:00-09:00	14:00-16:00
Traffic density (V/km^2)	37.03	16.36
Number of attackers	1220	550
Number of normal vehicles	2846	1179
Number of attacker messages	924251	249612
Number of normal messages	2221825	569723

We have the sender information in BSM with the receiving time for these two scenarios. However, the related receiver information does not exist in BSM. Therefore, it is necessary to know the receiver ID first to gather its information. This data is written in the name of each JSON file and can be extracted easily. After getting the receiver ID, its information related to the mentioned BSM can be found by comparing the receiving time in BSM and the sending time in GPS. The GPS line for this receiver ID is selected on the closest sending time. This brings the receiver information when it received the sent message. Thanks to this aggregation, one can have the sender and the receiver information on the same data line. This helps so to calculate the necessary information between the sender and receiver.

C. Evaluation Indicators

The performance of the proposal is evaluated by accuracy, precision, recall, and F-Score, which are well-studied performance indicators in classification problems. The confusion matrix is first determined to calculate these performance indicators, which consist of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). Table II explains the meanings of these attributes where "1" denotes the attacker and "0" the normal vehicle.

TABLE II
CONFUSION MATRIX FOR CLASSIFICATION

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

Accuracy gives the general correct classification ratio, the rate of true prediction of both 0 and 1 to all cases (Eq. (1)). As well as accuracy, precision and recall denote the true positive rate compared to all predicted positive cases and all actual positive cases, respectively, as shown in Eqs. (2)-(3).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1}$$

$$Precision = \frac{TP}{TP+FP} \tag{2}$$

$$Recall = \frac{TP}{TP+FN} \tag{3}$$

Precision and recall show an inverse trend: when precision increases, recall will decrease, and vice versa. Therefore, a trade-off indicator between precision and recall is needed, called F1-Score. F1-Score calculates the harmonic average of these two indicators to solve this issue in Eq. (4).

$$F1\text{-}Score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

D. Results and Comparison

The experiments are done by using Python. Each log file in JSON format has been converted to a CSV file and combined for the selected individual scenario. 80% of the dataset is selected as the training set, and the remaining as the testing set in all the experiments [42]. To combine the results of XGBoost and Random Forest in stacking, Logistic Regression is conducted as a second classifier.

1) Results and Comparison for the Proposed Method: The first experiment is carried out to compare the proposed ML method with a similar related work [24]. In this work, the authors also used well-known ML methods firstly: k-Nearest Neighbor, Decision Tree, AdaBoost, and Random Forest. Then, the results of these methods were combined by using Logistic Regression. Finally, they considered four attacks: DoS, Sybil, Replay, and Distributed DoS. Since the attack types and the detection methods seem similar to our proposal, this work was chosen to compare the method performance.

Table III compares the methods used in the aforementioned related work [24] and the proposed detection method (P1) on the existing dataset in terms of accuracy. For all types of attacks, the P1 gives accuracy values equal to or better than the related work. However, DoS Disruptive and DoS Disruptive Sybil attacks are detected with less accuracy in the two methods, probably due to the features used. This is discussed in Section IV-D2.

Since F1-Score already calculates the harmonic average of precision and recall, we present the results in terms of F1-Score. As seen in Figure 3, the F1-Score values show a similar pattern to the accuracy results: The results for the P1 are slightly higher than the results of [24], and both methods are less efficient in detecting DoS Disruptive and DoS Disruptive Sybil attacks compared to the other attack types.

For the implementation of ML methods, the computational time, i.e. the time for learning and test phases, is important, as well as the classification performance indicators (accuracy and F1-Score). Therefore, the improvements in learning time and test time are also presented in Table IV. Furthermore, we significantly reduce the time for both phases for almost all types of attack detection. Therefore, the proposed method achieves results that are at least as good as the benchmarked solution while reducing the computation time. Thus, the proposed ML approach provides better attack detection in this case.

TABLE III
ACCURACY OF [24] AND THE PROPOSED METHOD ON THE EXISTING DATASET (P1)

Attack Type and Scenario	[24]	P1
DoS_0709	0.9908	**0.9910**
DoS_1416	0.9899	**0.9905**
DoSRandom_0709	**0.9999**	**0.9999**
DoSRandom_1416	**0.9993**	**0.9993**
DoSDisruptive_0709	0.7516	**0.7521**
DoSDisruptive_1416	0.7477	**0.7485**
DoSRandomSybil_0709	0.9998	**0.9999**
DoSRandomSybil_1416	**0.9996**	**0.9996**
DoSDisruptiveSybil_0709	0.9255	**0.9256**
DoSDisruptiveSybil_1416	0.9151	**0.9152**

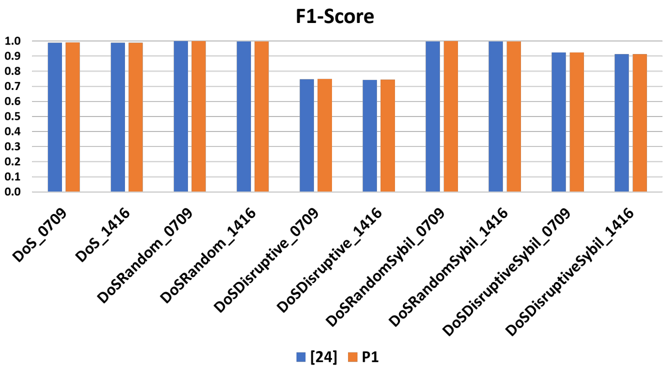

Fig. 3. F1-Score of [24] and the proposed method on the existing dataset (P1).

TABLE IV
IMPROVEMENTS IN LEARNING AND TEST TIME COMPARING TO [24]

Attack Type and Scenario	Learning Time	Test Time
DoS_0709	98.21%	99.96%
DoS_1416	97.88%	99.93%
DoSRandom_0709	97.37%	99.95%
DoSRandom_1416	97.26%	99.95%
DoSDisruptive_0709	98.13%	99.96%
DoSDisruptive_1416	97.84%	99.91%
DoSRandomSybil_0709	96.39%	99.92%
DoSRandomSybil_1416	98.31%	99.97%
DoSDisruptiveSybil_0709	96.23%	99.93%
DoSDisruptiveSybil_1416	97.63%	99.92%

2) Results and Comparison for the Proposed Features: The second experiment is conducted to perform the proposed detection method on the given dataset and the proposed dataset with new features to compare with the first work using the VeReMi dataset [41]. This dataset includes the position, speed, acceleration and heading values in the x and y axes (values in the z axis were 0 for all, so ignored) and the noise of these features in both axes. Figure 4 shows the accuracy values of the first work [41] and the proposed detection method on the proposed dataset with the selected features (P2). The P2 outperforms the P1 and [41] in terms of accuracy for all attack types in both scenarios.

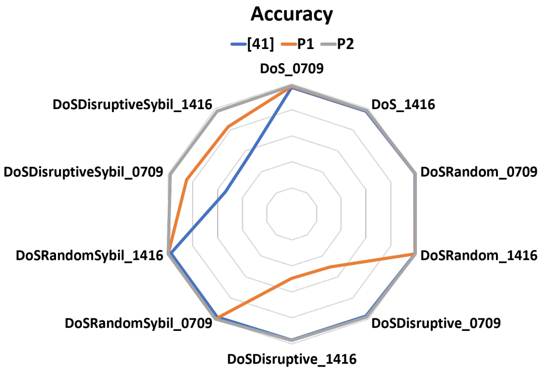

Fig. 4. Accuracy of [41] and the both proposals (P1 and P2).

As depicted in Figure 5, the F1-Score values for detection of DoS Disruptive and DoS Disruptive Sybil attacks are relatively low in both scenarios for the P1. However, the P2, where we propose new features, performs better for all attacks. Moreover, the results for detecting DoS Random and DoS Random Sybil attacks present an F1-Score of 100% by implementing the P2.

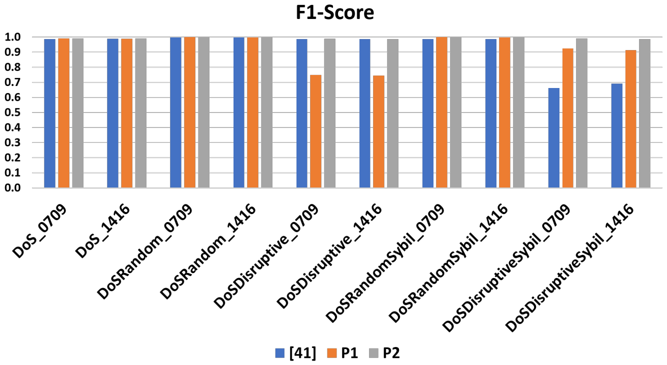

Fig. 5. F1-Score of [41] and the both proposals (P1 and P2).

In all these results, the detection of DoS Disruptive attack shows a significant advance with the new proposed features, followed by DoS Disruptive Sybil attack. Although the other three types have already performed well with the P1 algorithm, their detection performances are still improved, even if it is slightly improved.

3) Results and Comparison for the Proposed Sampling Approaches: The last experiment is realized to present the performance of the proposed method with undersampling and oversampling approaches. These sampling approaches are expected to provide balanced datasets for better classification.

As shown in Figure 6, oversampling achieves the best accuracy results for all types of attacks in both scenarios. Furthermore, Dos Random and Dos Random Sybil attacks are accurately detected using the P2 algorithm regardless of the sampling approach.

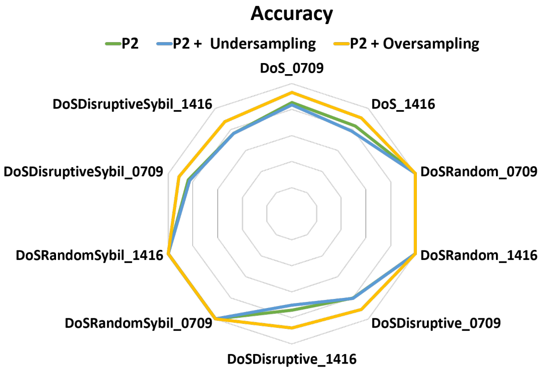

Fig. 6. Accuracy of the proposed method with proposed features (P2) with sampling approaches.

Similarly to the accuracy results, Figure 7 represents the F1-Score results for the mentioned three proposals. These results indicate that the proposed method, features, and sampling approaches develop not only an accurate attack detection system but also a stabilized one for all defined DoS-related attacks in VANETs.

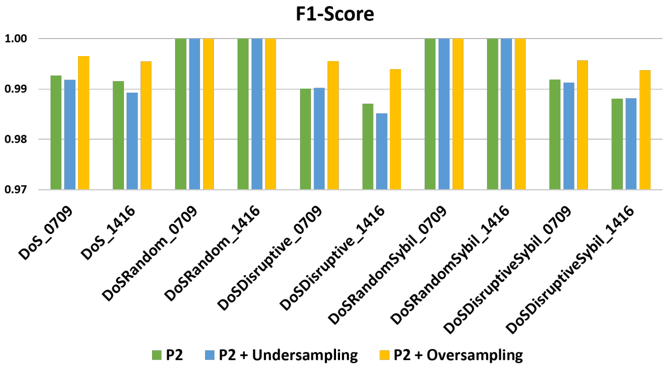

Fig. 7. F1-Score of the proposed method with proposed features (P2) with sampling approaches.

V. Conclusion

The main motivation of this work is to enhance security in vehicular networks, especially concerning data availability. To enable that, we propose a novel AI-based approach with a suitable feature set, to detect Denial-of-Service attacks in VANETs. Well-known ensemble learning techniques are implemented to provide an Intrusion Detection System with high detection performance. Three types of Enesemble Learning algorithms are implemented together, i.e. boosting, bagging and stacking.

The proposed solutions are validated on a public dataset (VeReMi) with different sampling approaches. Since imbalanced data can affect classification performance, undersampling and oversampling approaches are employed to obtain a better dataset. The results of the proposed solution present outstanding performance in terms of all performance indicators, i.e. accuracy and F1-score, and mainly for DoS Random and DoS Random Sybil attacks. Hence, the proposed solution can easily handle the random behavior of the attacker.

This proposal will be extended by using Deep Learning methods and various features depending on the different attack types in VANETs. Moreover, these approaches will also be improved to enable intrusion detection in other wireless networks.

References

[1] S. Maaloul, H. Aniss, L. Mendiboure, and M. Berbineau, "Performance analysis of existing its technologies: Evaluation and coexistence," *Sensors*, vol. 22, no. 24, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/24/9570

[2] A. Festag, "Cooperative intelligent transport systems standards in europe," *IEEE communications magazine*, vol. 52, no. 12, pp. 166–172, 2014.

[3] Z. MacHardy, A. Khan, K. Obana, and S. Iwashina, "V2x access technologies: Regulation, research, and remaining challenges," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 1858–1877, 2018.

[4] M. J. A. Junaid, A. A. Syed, M. N. M. Warip, K. N. F. K. Azir, and N. H. Romli, "Classification of security attacks in vanet: A review of requirements and perspectives," in *MATEC web of conferences*, vol. 150, 2018, p. 06038.

[5] F. Sakiz and S. Sen, "A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov," *Ad Hoc Networks*, vol. 61, pp. 33–50, 2017.

[6] S. Sharma and A. Kaul, "A survey on intrusion detection systems and honeypot based proactive security mechanisms in vanets and vanet cloud," *Vehicular Communications*, vol. 12, pp. 138–164, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209617302784

[7] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, 2019.

[8] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on VANET security challenges and possible cryptographic solutions," *Vehicular Communications*, vol. 1, no. 2, pp. 53–66, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209614000187

[9] V. Raghuwanshi and S. Jain, "Denial of service attack in vanet: a survey," *International Journal of Engineering Trends and Technology (IJETT)*, vol. 28, no. 1, pp. 15–20, 2015.

[10] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *IEEE Access*, vol. 9, pp. 7550–7563, 2021.

[11] A. R. Gad, A. A. Nashat, and T. M. Barkat, "Intrusion detection system using machine learning for vehicular ad hoc networks based on ton-iot dataset," *IEEE Access*, vol. 9, pp. 142 206–142 217, 2021.

[12] D. Karagiannis and A. Argyriou, "Jamming attack detection in a pair of rf communicating vehicles using unsupervised machine learning," *Vehicular Communications*, vol. 13, pp. 56–63, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221420961730222X

[13] A. Mchergui, T. Moulahi, and S. Zeadally, "Survey on artificial intelligence (ai) techniques for vehicular ad-hoc networks (vanets)," *Vehicular Communications*, vol. 34, p. 100403, 2022.

[14] M. Siracusano, S. Shiaeles, and B. Ghita, "Detection of lddos attacks based on tcp connection parameters," in *2018 Global Information Infrastructure and Networking Symposium (GIIS)*, 2018, pp. 1–6.

[15] U. Sabeel, S. S. Heydari, H. Mohanka, Y. Bendhaou, K. Elgazzar, and K. El-Khatib, "Evaluation of deep learning in detecting unknown network attacks," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2019, pp. 1–6.

[16] S. Çalışır, R. Atay, M. K. Pehlivanoğlu, and N. Duru, "Intrusion detection using machine learning and deep learning techniques," in *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 2019, pp. 656–660.

[17] K. Wehbi, L. Hong, T. Al-salah, and A. A. Bhutta, "A survey on machine learning based detection on ddos attacks for iot systems," in *2019 SoutheastCon*, 2019, pp. 1–6.

[18] S. Peneti and H. E, "Ddos attack identification using machine learning techniques," in *2021 International Conference on Computer Communication and Informatics (ICCCI)*, 2021, pp. 1–5.

[19] G. De Carvalho Bertoli, L. A. Pereira Júnior, O. Saotome, A. L. Dos Santos, F. A. N. Verri, C. A. C. Marcondes, S. Barbieri, M. S. Rodrigues, and J. M. Parente De Oliveira, "An end-to-end framework for machine learning-based network intrusion detection system," *IEEE Access*, vol. 9, pp. 106 790–106 805, 2021.

[20] T. G. Zewdie and A. Girma, "An evaluation framework for machine learning methods in detection of dos and ddos intrusion," in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, 2022, pp. 115–121.

[21] S. RoselinMary, M. Maheshwari, and M. Thamaraiselvan, "Early detection of dos attacks in vanet using attacked packet detection algorithm (apda)," in *2013 International Conference on Information Communication and Embedded Systems (ICICES)*, 2013, pp. 237–240.

[22] K. Jeffane and K. Ibrahimi, "Detection and identification of attacks in vehicular ad-hoc network," in *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2016, pp. 58–62.

[23] E. Pastori Valentini, R. Ipolito Meneguette, and A. Alsuhaim, "An attacks detection mechanism for intelligent transport system," in *2020 IEEE International Conference on Big Data (Big Data)*, 2020, pp. 2453–2461.

[24] Y. Liu, H. Xue, W. Zhuang, F. Wang, L. Xu, and G. Yin, "Ct2-mds: Cooperative trust-aware tolerant misbehaviour detection system for connected and automated vehicles," *IET Intelligent Transport Systems*, vol. 16, no. 2, pp. 218–231, 2022. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/itr2.12139

[25] A. K. Sharma, S. K. Saroj, S. K. Chauhan, and S. K. Saini, "Sybil attack prevention and detection in vehicular ad hoc network," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2016, pp. 594–599.

[26] J. Santa, F. Pereñíguez, A. Moragón, and A. F. Skarmeta, "Experimental evaluation of cam and denm messaging services in vehicular communications," *Transportation Research Part C: Emerging Technologies*, vol. 46, pp. 98–120, 2014.

[27] S. Ercan, M. Ayaida, and N. Messai, "New features for position falsification detection in vanets using machine learning," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[28] A. K. Uttam and G. Sharma, "A comparison of data balancing techniques for credit card fraud detection using neural network," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2021, pp. 1136–1140.

[29] S. K. Gunturi and D. Sarkar, "Ensemble machine learning models for the detection of energy theft," *Electric Power Systems Research*, vol. 192, p. 106904, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378779620307021

[30] T. N. Rincy and R. Gupta, "Ensemble learning techniques and its efficiency in machine learning: A survey," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 2020, pp. 1–6.

[31] D. Gaikwad and R. C. Thool, "Intrusion detection system using bagging ensemble method of machine learning," in *2015 International Conference on Computing Communication Control and Automation*, 2015, pp. 291–295.

[32] S. Georganos, T. Grippa, S. Vanhuysse, M. Lennert, M. Shimoni, and E. Wolff, "Very high resolution object-based land use–land cover urban classification using extreme gradient boosting," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 4, pp. 607–611, 2018.

[33] H. Rajadurai and U. D. Gandhi, "A stacked ensemble learning model for intrusion detection in wireless network," *Neural Computing and Applications*, pp. 1433–3058, 2020.

[34] S. Bhattacharya, S. R. K. S, P. K. R. Maddikunta, R. Kaluri, S. Singh, T. R. Gadekallu, M. Alazab, and U. Tariq, "A novel pca-firefly based xgboost classification model for intrusion detection in networks using gpu," *Electronics*, vol. 9, no. 2, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/2/219

[35] S. S. Dhaliwal, A.-A. Nahid, and R. Abbas, "Effective intrusion detection system using xgboost," *Information*, vol. 9, no. 7, 2018. [Online]. Available: https://www.mdpi.com/2078-2489/9/7/149

[36] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[37] 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: Springer International Publishing, 2018, pp. 318–337.

[38] W. Arellano and I. Mahgoub, “Trafficmodeler extensions: A case for rapid vanet simulation using, omnet++, sumo, and veins,” in *2013 High Capacity Optical Networks and Emerging/Enabling Technologies*. IEEE, 2013, pp. 109–115.

[39] A. Varga, “Omnet++,” in *Modeling and tools for network simulation*. Springer, 2010, pp. 35–59.

[40] D. Krajzewicz, “Traffic simulation with sumo–simulation of urban mobility,” in *Fundamentals of traffic simulation*. Springer, 2010, pp. 269–293.

[41] J. Kamel, M. Wolf, R. W. van der Hei, A. Kaiser, P. Urien, and F. Kargl, “VeReMi extension: A dataset for comparable evaluation of misbehavior detection in VANETs,” in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[42] G. Kaur and D. Kakkar, “Hybrid optimization enabled trust-based secure routing with deep learning-based attack detection in vanet,” *Ad Hoc Networks*, vol. 136, p. 102961, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870522001378