A Machine Learning Framework for Intrusion Detection in VANET Communications

Nourhene Ben Rabah and Hanen Idoudi

1 Introduction

Vehicular ad hoc networks (VANET) stand for different communication schemas that can be performed between connected vehicles and anything (V2X). This includes vehicle-to-vehicle communications, vehicle-to-roadside infrastructure components, or intra-vehicle communications.

A VANET system relies on two main components: Roadside Unit (RSU) and On-Board Unit (OBU). RSU is the roadside communication equipment. It provides Internet access to vehicles and ensures exchanging data between vehicles. The OBU is the mobile treatment and communication unit embedded on the vehicle. It allows communication with other vehicles or with the infrastructure's equipment. VANET communication can be deployed according to different architectures, such as vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), infrastructure-to-vehicle (I2V), infrastructure-to-infrastructure (I2I), and hybrid [1]. Furthermore, a VANET system is composed of three planes: vehicular plane, RSU plane, and services plane. In the vehicular plane, each vehicle is equipped with OBU. The latter allows V2V communication. The RSU plane facilitates V2I, I2V, and I2I communications. In the service plane, different types of services can be deployed such as safety, infotainment, payment, Internet, and cloud-based services. A VANET has some similar features of MANET (mobile ad hoc networks) such as omnidirec-

N. Ben Rabah
Centre de Recherche en Informatique, Université Paris 1 Panthéon Sorbonne, Paris, France

ESIEE-IT, Pontoise, France
e-mail: nbenrabah@esiee-it.fr

H. Idoudi (✉)
National School of Computer Science, University of Manouba, Manouba, Tunisia
e-mail: hanen.idoudi@ensi-uma.tn

K. Daimi et al. (eds.), *Emerging Trends in Cybersecurity Applications*,
https://doi.org/10.1007/978-3-031-09640-2_10

tional broadcast, short transmission range, and low bandwidth. In contrast, it has particular characteristics. First, a VANET has a highly dynamic topology due to the high mobility of vehicles. This leads also to frequent disconnections. Secondly, target vehicles can be reached upon their geographical location. Thirdly, signal propagation is affected by the environment such as buildings, trees, etc. [1]. Finally, energy, storage failure, and computing capacity are less critical for VANETs as for MANET. Despite that, the serious challenge for VANET is processing huge amount of data in a real-time manner.

This diversity of communication schemas and the inherent characteristics of wireless communications make VANETs vulnerable to many security attacks and vulnerabilities. This is emphasized by the critical aspect of some exchanged information that is used for road safety purposes. Security breaches are several and can affect all network layers and all communication aspects in VANET. Moreover, VANETs suffer from traditional vulnerabilities that affect any wireless environment but are also subject to new and specific attacks exploiting inherent vehicular characteristics [1]. Most of the security solutions defined for traditional networks are not suitable for vehicular networks. Subsequently, researchers are looking for appropriate systems that support vehicular network characteristics and provide robust security mechanisms.

Different security countermeasures have been proposed such as key management systems, anonymity, traceability techniques, cryptographic algorithms, trust management methods, etc. [2]. Recently, many researchers showed that integrating artificial intelligent (AI) methods in intrusion detection systems increases their effectiveness in detecting attacks on V2X networks. IDS are a widely used approach that analyzes the traffic for indicators of security breaches and creates an alert for any observed security anomaly. Moreover, machine learning (ML) can realize anomaly-based detection systems capable of detecting unknown and zero-day attacks, learning, and training itself by analyzing network activity and increasing its detection accuracy over time.

Applying ML techniques for intrusion detection in VANET is of particular interest due to the huge amount of exchanged data and the diversity of attacks that can occur. In recent years, many published datasets, describing real traces of VANET communication, have allowed the assessment of ML techniques performances for intrusion detection.

This work intends to define a novel comprehensive framework to design an IDS for V2X communications. Furthermore, unlike most existing works, we use a very recent dataset to evaluate and compare both ensemble and standalone learning techniques to detect various types of DOS and DDOS attacks in VANET.

We define first a novel framework for applying ML techniques to detect anomalies in VANET communication. Then, we use a very recent dataset, VDOS-LRS dataset, that describes urban vehicular traffic to assess and compare the performances of well-known standalone ML methods and ensemble ML methods to detect DOS and DDOS attacks in urban environment.

The rest of this chapter is structured as follows.

In Sect. 2, we review related works related to security issues in VANET, and we review most important works on ML-based IDS for VANETs. In Sect. 3, we expose our framework for designing ML-based IDS for VANET communication. Main results are discussed in Sect. 4 where we study the performances of several ML techniques, both standalone and ensemble learning techniques, on detecting DOS and DDOS attacks in urban traffic using a very recent VANET dataset, namely, VDOS-LRS dataset.

Finally, Sect. 5 gives the conclusion of the study.

2 Security of VANET Communications

In this section, we discuss the security issue in VANET communication; then, we focus on the most important works that considered the use of machine learning-based intrusion detection systems for VANET.

2.1 Security Attacks and Vulnerabilities in VANET

In-vehicle communications involve embedded units mainly interacting via CAN-Bus, Ethernet, or WiFi standards whereas inter-vehicle networks refer to different kind of interactions between vehicles and other components of the ITS system. These latter can be vehicle-to-infrastructure (V2I), vehicle-to-cloud (V2C), vehicle-to-vehicle (V2V), and vehicle-to-device (V2D) communications [1]. This diversity of architectures and communication schemes led to the inception of the vehicle-to-anything or V2X paradigm.

Many security attacks are targeting VANET communications taking profit from the highly heterogeneity of such environments, the highly dynamic topology induced by mobility, and the lack of standard security so far [1, 2]. Security requirements such as availability, data integrity, confidentiality, authenticity, and non-repudiation can be compromised.

Denial of Service (DoS) and Distributed DoS (DDOS) attacks aim to disrupt network service's availability by flooding the OBU (On-Board Unit) and/or RSU (Roadside Unit) communication channels with an unhandled huge amount of requests, resulting in network out of service [3]. In black hole and gray hole attacks, an attacker can capture illegitimate traffic, then drops, retains, or forwards them to erroneous destinations [4]. In Sybil attacks, malicious nodes may create several virtual cars with the same identity to mislead some functionalities. Node impersonation attack tries to impersonate legitimate node's identity. Additionally, GPS spoofing or position faking attacks, also known as hidden vehicle attacks, generate fake position alarms [5].

Different attacks can also threaten the integrity and/or the confidentiality of data such as tampering attacks and spoofing [1, 2].

In-vehicle communications are equally vulnerable as the inter-vehicle communications and can also suffer from all kinds of attacks following the illegitimate intrusion of malicious data [6].

We compare the characteristics of some notable security attacks in Table 1 with regard to the targeted environment and the compromised security requirement.

2.2 Security Countermeasures

Many security mechanisms are considered to secure vehicular communication while taking into account their inherent characteristics. Most important cover the following categories.

– *Cryptography*

Its aim is to ensure confidentiality of data while being transmitted from one source node to a destination node. Moreover, they involve encryption algorithms, hash functions, and digital signature algorithm and can provide solutions for diverse types of threats at different levels in VANET. New lightweight solutions for data encryption are more considered to tackle the limited computation capacities of different VANET equipment. The Elliptic Curve Digital Signature Algorithm (ECDSA) is one of the most widely used digital signatures algorithms in IoT in general and in securing VANET communications [7] [8].

– *Key Management Systems*

PKI are core ITS component for identity and key management and can be implemented as centralized, decentralized, or distributed systems. Many enhanced solutions based on PKI are proposed to secure authentication and revocation [9]. For instance, in [33], authors define Enhanced Secure Authentication and Revocation (ESAR) scheme for VANETs which is responsible for revocation checking, processing, and PKI key pair updating.

– *Anonymity, Unlinkability, and Traceability Techniques*

These strategies intend to ensure the privacy of users' data by means of data suppression, randomization, or cloaking to prevent unauthorized access. They offer a countermeasure against several attacks such as eavesdropping, trajectory tracking, or location disclosure.

For instance, anonymity techniques are based on the use of pseudonyms by Group Signature and Pseudonymous Authentication schemes. In a group signature approach, a group private key will be used by all vehicles, whereas in pseudonymous authentication schemes, each vehicle is assigned a set of identities that it stores locally. Hybrid approaches that combine both group signature and pseudonymous authentication schemes are also considered [10, 11].

To achieve traceability, unique electronic license plate (ELP) should be used. Pseudonyms could be linked with a specific ELP identity. This would allow

Table 1 Characteristics of some VANET security attacks

	Attack	Internal	External	Inter-veh.	In-veh.	Targeted security requirement: Availability	Integrity	Confidentiality	Authenticity	Non-repudiation
t3.1	DOS/DDOS	X	X	X	X	X			X	
t3.2	Black hole/grey hole		X	X		X				
t3.3	Hidden vehicle	X		X				X	X	
t3.4	Node impersonation	X		X		X			X	
t3.5	Spoofing	X	X	X	X				X	
t3.6	Position falsification		X	X			X			
t3.7	Sybil	X		X			X		X	X
t3.8	Replay	X		X	X		X		X	X
t3.9	Fuzzy	X			X					

authorities to trace a misbehaved user whenever it is needed. Moreover, in group signatures, a tracing manager can revoke the malicious vehicles by analyzing their signatures [12].

– *Security Protocols*

Standard communication and routing protocols need to be secured, hence the need for integrating with security protocols at network, transport, or application level. Several security protocols are proposed or adapted to the context of IoT communications in general such as TLS and DTLS [13].

– *Intrusion Detection Systems*

Intrusion detection systems (IDS) are an efficient way to detect and prevent malicious or abnormal activities. A typical IDS relies on three main components:

- Information collector: It relies on sensors commonly deployed at different sensitive locations.
- Analysis engine: Its main purpose is to analyze information collected via sensors.
- Reporting engine: This component is responsible for logging and raising alarms when a malicious node or an abnormal event is detected.

In VANET networks, IDS sensors are generally located at RSU and on vehicles. First, these sensors collect nodes' communication information. Second, the data collected is sent to the analysis engine. Third, the analysis engine analyzes the received data using different methods which depend on the IDS type. If an abnormal event or a malicious node is detected, a report is sent to the reporting engine. Finally, the reporting engine informs the appropriate nodes about the attack.

IDS for VANET are mainly classified into four categories. This classification is based on the techniques used to detect threats. These classes are signature based, watchdog based, behavior based, and hybrid IDS [2].

2.3 *ML-Based Intrusion Detection Systems for VANETs*

Behavior-based IDS, also known as anomaly-based, use AI and ML as well as other statistical methods to analyze data on a network to detect malicious behavior patterns as well as specific behaviors that may be linked to an attack.

ML-based IDS are part of behavior-based IDS. This approach assumes that intrusive activities are a subclass of abnormal activities. In ML-based IDS, different ML techniques can be used to recognize the misbehavior pattern. In fact, it extracts relations between different attributes and builds attack models [7]. This mechanism allows the RSU or OBU to detect any misbehavior in the network by analyzing received messages and network information. The main advantage of this approach is its ability to detect zero-day attacks and anomalies.

So far, many works adopted ML techniques to build efficient IDS.

In [14], Fuad A. Ghaleb et al. proposed a misbehavior detection model based on ML techniques. Authors used real-world traffic dataset, namely, Next Generation Simulation (NGSIM) to train and evaluate the model. They used artificial neural network.

In [3], authors aimed at detecting wormhole attacks in VANET using ML-based IDS. Firstly, they generated a dataset by using both the traffic simulator Simulation of Urban Mobility Model (SUMO) and NS3. Secondly, two different ML algorithms were applied on the generated dataset to train the model, namely, k-nearest neighbors (kNN) and support vector machines (SVM). Finally, to evaluate the different models, the authors used the accuracy rate and four different alarms which are true positive (TP), false positive (FP), true negative (TN), and false negative (FN). As a result, authors pointed out that both the SVM and kNN performed well on detecting wormhole attacks.

In [15], authors proposed a ML-based IDS to detect position falsification attack in VANET. To train and evaluate ML models, the authors used Vehicular Reference Misbehavior Dataset (VeReMi dataset). Authors used logistic regression (LR) and SVM models. To evaluate the work, they used F-measure. As a result, they proved that SVM performed better than LR.

In [16], authors developed an intrusion detection system based on gradient boosting decision tree (GBDT) for CAN-Bus and proposed a new feature based on entropy as the feature construction of GBDT and used a dataset from a real domestic car to evaluate the model.

Authors in [17] showed that tree-based and ensemble learning models show more performance in detection compared to other models. Random forest, bagging, and AdaBoosting methods are trained and tested on the Can-hacking dataset, and the DT-based model results in yield performance.

Vuong et al. [18] proposed a decision tree-based method for detecting DoS and command injection attacks on robotic vehicles using cyber and physical features to show the importance of incorporating the physical features in improving the performance of the model. They tested their model in a collected dataset. In addition to DoS and command injection attack detection, they also provide in [19] a lightweight intrusion detection mechanism that can detect malware against network and malware against CPU using both cyber and physical input features using the decision tree model.

A tree-based intelligent IDS for the internet of vehicles (IoV) that detects DoS and fuzzy attacks is proposed by Li Yang et al. [20]. Firstly, they tested the performance of decision tree (DT), random forest (RF), extra trees (ET), and XGradient Boost (XGB) methods and applied multi-threading to get a lower execution time. Then, they selected three models that generate the lowest execution time as a meta-classifier in the second layer of the stacking ensemble model. Besides, they used an ensemble feature selection (FS) technique to improve the confidence of the selected features. Finally, the authors tested the model on the car-hacking dataset.

In [34], authors define a novel machine learning model using random forest and a posterior detection based on coresets. Their model showed high accuracy for DOS attack detection.

The use of ML techniques is undoubtedly efficient, but due to the numerous opportunities that ML techniques offer, more works are still needed to investigate the design of the best ML framework for VANET IDS.

In our work, we intend to define a comprehensive framework to design VANET IDS. Furthermore, unlike most existing works, we use a very recent dataset to evaluate and compare both ensemble learning and standalone learning techniques to detect various types of DOS attack.

Our contribution is exposed in the next section.

3 Proposed ML Framework

In this section, we introduce a novel machine learning framework for intrusion detection in V2X communications. The elaboration process comprises three major phases: dataset description, data preprocessing, and the application of standalone and ensemble learning methods, as shown in Fig. 1.

3.1 First Phase: Dataset Description

One of the challenges of building efficient V2X ML-based IDS is the lack of public datasets with a big collection of network traffic logs depicting both normal and abnormal activities. More recent works that tried to tackle IDS design using ML or DL (deep learning) techniques to mitigate more complex or new attacks have pointed out this problem, and some tried to build simulated datasets at that end [21–

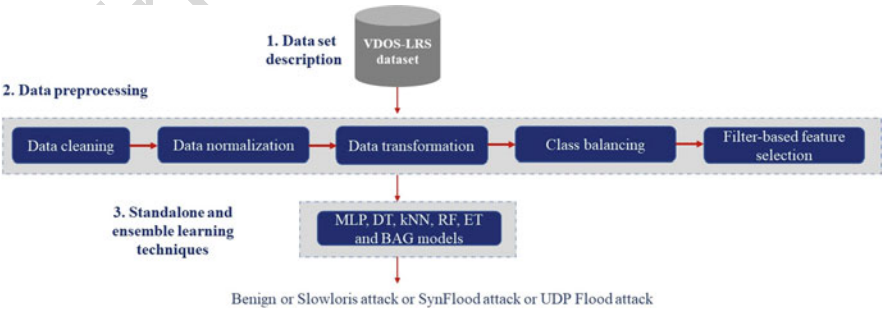

Fig. 1 Proposed ML framework

23]. A survey of the most important and most recent datasets dedicated to VANET communication and involving some well-known security attacks is given in [24].

To evaluate the proposed framework, we used the Vehicular Denial of Service Networks and Systems Laboratory (VDOS-LRS) dataset [25]. It is one of the most recently published datasets that incorporate real network traffic collected in different environments (urban, rural, and highway). This dataset involves traces of three DoS attacks:

– SYN flood attack is based on sending a huge number of TCP-SYN requests to a vehicle to make it out of service.
– UDP flood overloads random ports on the targeted host with UDP datagrams.
– Slowloris attack is an application layer DDoS attack that uses partial HTTP requests to open multiple connections towards a target.

For this study, we focused on the urban environment. It is initially presented as a PCAP file. For this purpose, we used the network traffic flow generator and analyzer, CICFlowMeter [26], which allowed us to generate bidirectional flows described through 84 statistical features such as duration, number of packets, number of bytes, packet length, etc. These flows are then saved as a csv file, representing our dataset. It includes 26,334 normal instances, 124,968 SYN flood attack instances, 122,457 UDP flood attack instances, and 650 Slowloris attack instances.

3.2 Second Phase: Data Preprocessing

These different steps are used to improve the data quality and, consequently, the performance of the machine learning models. It includes data cleaning, data normalization, data transformation, and class balancing.

1. *Data Cleaning*

It is used to handle erroneous, inaccurate, and irrelevant data to improve the dataset quality. Indeed, we do not consider source and destination IP addresses and ports, as attackers can easily modify them [22]. Therefore, we removed these five features: “Flow ID,” “Src IP,” “Src Port,” “DST IP,” and “DST Port.” Thus, we replaced the missing values of some features with the mean values of these features.

2. *Data Normalization*

It is performed to avoid bias when feature values belong to very different scales. Some features in our dataset vary between 0 and 1, while others can reach infinite values. Therefore, we normalized these features according to Eq. 1, defined as follows:

$$X_{\text{normalized}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (1)$$

where $X_{\text{normalized}}$ is the normalization result and X is the initial value. Here, $X_{\max}$ and $X_{\min}$ represent the maximum and the minimum values of each feature, respectively.

3. *Data Transformation*

It is used to modify data to fit the input of any ML model. Indeed, some ML models can work with qualitative data (i.e., non-numerical data) such as k-nearest neighbors (kNN), naive Bayes (NB), and decision trees (DT). However, most of them require numerical inputs and outputs to work properly. Therefore, it is important to convert qualitative data to numerical data. In our dataset, each instance is represented by 77 numerical features and one object feature ("Timestamp") that represents the date and time values of the flow. In this step, we propose to replace this feature by six features of numerical type: "Timestamp_year," "Timestamp_month," "Timestamp_day," "Timestamp_hour," "Timestamp_minute," and "Timestamp_second."

4. *Class Balancing*

Class imbalance is a major problem in supervised ML methods. It usually occurs when the dataset traces are collected from a real environment. Indeed, in such an environment, the data is usually unbalanced, and the models learned from the data may have better accuracy on the majority class but very poor accuracy on the other classes. There are three main ways to deal with this problem: modifying the ML algorithm, introducing a misclassification cost, and data sampling [27]. Data sampling is the only solution that can be done independently of the classification algorithm, since the other two require direct or indirect modifications to the algorithm. Data sampling is performed using two methods: undersampling the majority class or oversampling the minority class.

Since the classes in our dataset are unbalanced (see Fig. 2), we use the Synthetic Minority Oversampling Technique (SMOTE) [28, 29] to solve this problem. SMOTE involves synthesizing new examples of the minority classes so that the number of examples of the minority class gets closer to or matches the number of examples of the majority class. After performing it, we get 124,968 instances of each class (see Fig. 3).

5. *Filter-Based Feature Selection*

Feature selection [30] is a very important step that consists in selecting from the initial dataset the most relevant features. Indeed, if there are too many features, or if most of them are not relevant, the models will consume more resources and be difficult to train. On the other hand, if there are not enough informative features, the models will not be able to perform their ultimate tasks.

To achieve such a goal, we propose to use a filter-based feature selection method that consists of selecting the most relevant subsets of features according to their relationship with the target variable. We, therefore, use statistical tests that consist in (a) evaluating the relationship between each input feature and the output feature and (b) discarding input variables that have a weak relationship with the target variable.

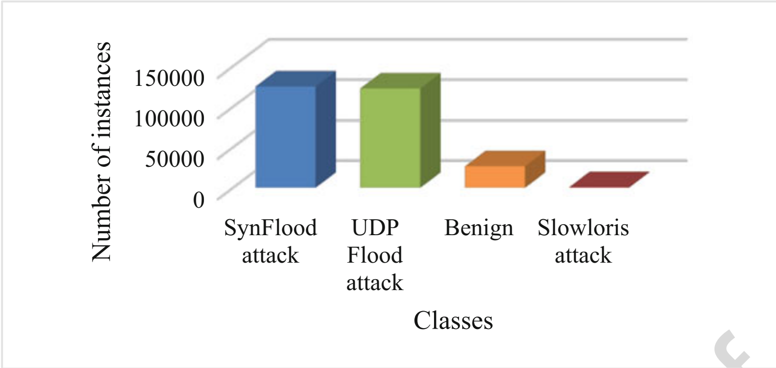

Fig. 2 Number of instances of each class before SMOTE: 124,968 instances of "SYN flood attack" (majority class), 122,457 instances of "UDP flood attack," 26,334 instances of "Benign," and 650 instances of "Slowloris attack" (minority class)

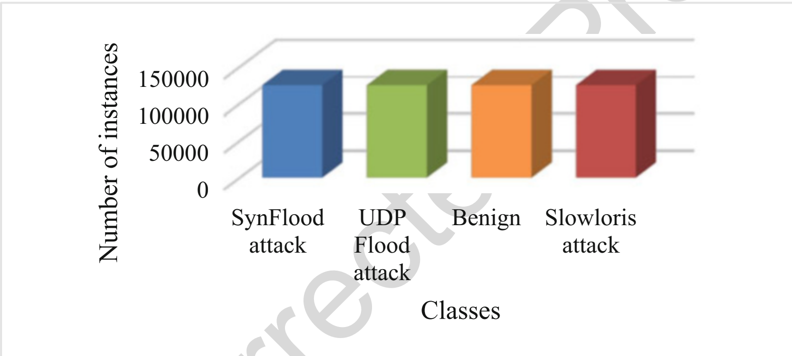

Fig. 3 Number of instances of each class after SMOTE: 124,968 instances of each class

AQ1 In other hand, keeping the input features that gives a strong statistical relationship with the output variable.

There are many statistical tests such as chi-squared, Pearson correlation, permutation feature importance, ANOVA F-value test, and others. The choice of statistical measures depends strongly on the types of input variables and the output variable (numerical or categorical). In our dataset, the input variables are of numerical type, and the output variable is of categorical type (the class), hence the interest to use two statistical measures which are:

- ANOVA F-value test that estimates the degree of linear dependence between an input variable and an output variable while giving a high score to strongly correlated features and a low score to weakly correlated features.
- Mutual Information (MI) that measures the reduction in uncertainty between each input variable and the target variable. The features in the set are classified

according to their MI value. A feature with a low MI value implies that it does not have much effect on the classification. Therefore, features with a low MI value can be discarded without affecting the performance of the models [31].

3.3 Third Phase: Standalone and Ensemble Learning Techniques

To validate our framework, we use two types of ML algorithms:

- Standalone algorithms such as multilayer perceptron (MLP), decision tree (DT), and k-nearest neighbors (kNN).
- Ensemble algorithms such as random forest (RF), extra tree (ET), and bagging (BAG).

We used the Scikit-learn library implementation of these algorithms [32]. The choice of these algorithms' hyperparameters has an impact on their performance. For this study, we have used the default values specified by Scikit-learn as they work reasonably well. It should be noted that the hyperparameters may be set using grid search or randomized search, but these methods are slow and costly.

4 Experimental Results

This section presents two strategies to check the results obtained by our proposed framework. First, we evaluate the performance of ML algorithms presented above before and after using the SMOTE method. Then, we outline the most relevant features according to the two filter-based feature selection methods: the ANOVA F-value test and the Mutual Information. All experiments were performed using ten-fold cross-validation.

4.1 Performance Metrics

To measure the performance of ML models, we used different metrics, such accuracy and F-measure. These metrics are calculated from four basic measures assessed for each class:

– True positive of the class C_i (TP_i)
– True negative of the class $C_i(TN_i)$
– False negative of the class $C_i(FN_i)$
– False positive of the class $C_i(FP_i)$

Table 2 Multi-class confusion matrix to illustrate TP_{Benign}, TN_{Benign}, FN_{Benign}, FP_{Benign}, and MS_{Benign}

	Benign	**SynFlood**	**UDP Flood**	**Slowloris**
Benign	TP_{Benign}	FN_{Benign}	FN_{Benign}	FN_{Benign}
Slowloris	FP_{Benign}	TN_{Benign}	MS_{Benign}	MS_{Benign}
SynFlood	FP_{Benign}	MS_{Benign}	TN_{Benign}	MS_{Benign}
SynFlood	FP_{Benign}	MS_{Benign}	MS_{Benign}	TN_{Benign}

with $i \in$ {Benign, SYN flood attack, UDP flood attack, and Slowloris attack}

In the following, we present these metrics calculated according to these outcomes:

- Accuracy represents the ratio of correctly recognized records to the entire test dataset. It is measured as follows:

$$\text{Accuracy} = \frac{\sum_{i=1}^{l} \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i}}{l} \tag{2}$$

- F-score (Eq. 3) is used to measure precision (Eq. 4) and recall (Eq. 5) at the same time. The F-score is the harmonic mean of precision and recall values and reaches its best value at 1 and worst value at 0. It is calculated as follows:

$$F - \text{score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

$$\text{Precision} = \frac{\sum_{i=1}^{l} \frac{TP_i}{TP_i + FP_i}}{l} \tag{4}$$

$$\text{Recall} = \frac{\sum_{i=1}^{l} \frac{TP_i}{TP_i + FN}}{l} \tag{5}$$

l is the number of classes.

We also propose to use the confusion matrix (CM) as it is representing performance results in an intuitive way for non-experts in ML. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in a real class. For example, we present in Table 2 a multi-class confusion matrix to illustrate TP_{Benign}, TN_{Benign}, FN_{Benign}, and FP_{Benign}. TP_{Benign} refers to the normal instances that are correctly classified, TN_{Benign} means attack instances (SYN flood, UDP flood, and Slowloris) that are correctly predicted, FN_{Benign} refers to the normal instances that are classified as attacks (i.e., false alarms that are triggered without a real attack), and FP_{Benign} means attack instances that are predicted as normal traffic. The diagonal of the matrix represents the well-classified instances (TP_{Benign} and TN_{Benign}). MS_{Benign} means attacks that are classified as other attacks.

4.2 Evaluation of ML Models Before and After SMOTE

Tables 3 and 4 show the detection performance of the standalone and ensemble models before and after oversampling with the SMOTE method, respectively. Looking at these results, we can see that, whatever the used algorithm, the accuracy is high in the original dataset. It exceeds 98% for all models. For kNN and MLP, the accuracy of the original dataset is even higher than that of SMOTE. Therefore, these results are incorrect because when the classes are not balanced, the minor classes have a negative effect on the accuracy. Therefore, the F-score is the best metric when working with an unbalanced dataset.

By analyzing these tables, we can see also that F-score values of DT, MLP, BAG, RF, and ET models are improved after oversampling by the SMOTE method. On the other hand, the F-score value of the kNN model decreased after oversampling by the SMOTE method, and this shows that the algorithm is not influenced by the class distribution. The model gave better results on the unbalanced dataset. Further observations show that DT, BAG, ET, and RF have the best accuracy using SMOTE (no significant difference). That's why we focus on those classifiers in the following.

To help non-experts in ML understand the performance of models after using SMOTE method, we present in Tables 5, 6, 7, and 8 the confusion matrices of the DT, BAG, RF, and ET models, respectively.

Table 3 Evaluation of standalone models before and after SMOTE

	Standalone models					
	DT		kNN		MLP	
Methods	Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
None	99.998	0.99960	99.814	0.99704	98.113	0.72191
SMOTE	99.998	**0.99998**	99.672	0.99672	94.176	**0.94150**

Table 4 Evaluation of ensemble models before and after SMOTE

	Ensemble models					
	BAG		RF		ET	
Methods	Accuracy	F-score	Accuracy	F-score	Accuracy	F-score
None	99.998	0.99978	99.991	0.99985	99.999	0.99977
SMOTE	99.998	**0.99998**	99.991	**0.99991**	99.999	**0.99999**

Table 5 Multi-class confusion matrix after SMOTE for DT

	Benign	SynFlood	UDP Flood	Slowloris
Benign	124963	2	2	1
Slowloris	0	124968	0	0
SynFlood	2	0	124966	0
SynFlood	0	0	0	124968

Table 6 Multi-class confusion matrix after SMOTE for BAG

	Benign	SynFlood	UDP Flood	Slowloris
Benign	124965	1	2	0
Slowloris	0	124968	0	0
SynFlood	4	0	124964	0
SynFlood	1	0	0	124967

Table 7 Multi-class confusion matrix after SMOTE for RF

	Benign	SynFlood	UDP Flood	Slowloris
Benign	124933	0	35	0
Slowloris	0	124968	0	0
SynFlood	5	0	124963	0
SynFlood	0	0	0	124968

Table 8 Multi-class confusion matrix after SMOTE for ET

	Benign	SynFlood	UDP Flood	Slowloris
Benign	124933	0	35	0
Slowloris	0	124968	0	0
SynFlood	5	0	124963	0
SynFlood	0	0	0	124968

These confusion matrices show that the different models globally correctly classify "Benign" instances and instances of different attacks. In other words, BAG and ET contain less false alarms than DT and RF (see orange columns). We get 3 false alarms for BAG and ET, 5 false alarms for DT and 35 for RF. We thus observe that the models classify very well Slowloris and SYN flood attacks but less for SYN flood attacks.

4.3 Feature Selection and Analysis

In Table 9, we present the performance of the different ML models incorporating the two feature selection methods, ANOVA F-value and Mutual Information, while varying the number of selected features.

The results analysis can be concluded in the following points:

- Of the two feature selection methods implemented, mutual information is comparatively the better performing.
- Among the 4 classifiers implemented, RF and ET give the best accuracies by varying the number of features from 10 to 45.
- Feature selection method using Mutual Information identifies features that have the strongest impact on the prediction. As an example, we can see in Table 10, 10, 12, and 25 features selected by Mutual Information.

5 Conclusion

VANETs suffer from several vulnerabilities due to the inherent characteristics of vehicles and the open radio environment. Security of VANET communications is hence a critical issue due to the diversity of VANET applications, architectures, and characteristics. Many works have been done to study security attacks and countermeasures that can tackle VANET vulnerabilities. Intrusion detection systems (IDS) are an efficient way to detect and prevent malicious activities; hence, they are necessary before triggering the appropriate countermeasure. The use of machine

Table 9 Comparison between the performance of ANOVA F-value and Mutual Information

Feature selection method	Number of features	Accuracy			
		DT	BAG	RF	ET
ANOVA F-value	**10**	95.746	95.757	95.757	96.969
	12	95.743	95.820	95.760	96.970
	25	99.612	99.618	99.497	99.604
	30	99.709	99.728	99.612	99.723
	35	99.717	99.729	99.597	99.710
	40	99.708	99.728	99.600	99.715
	45	99.709	99.734	99.584	99.709
Mutual Information	**10**	99.979	99.986	**99.990**	99.988
	12	99.992	99.993	99.994	**99.995**
	25	99.993	99.993	99.995	**99.996**
	30	99.993	99.994	**99.996**	99.995
	35	99.994	99.995	**99.996**	**99.996**
	40	**99.998**	99.997	**99.998**	**99.998**
	45	**99.998**	99.998	99.993	**99.999**

Table 10 The selected features using mutual information

Number of features	Features	
10	Flow Duration, Flow Pkts/s, Flow IAT Mean, Flow IAT Max, Flow IAT Min, Fwd Header Len, Bwd Header Len, Fwd Pkts/s, Bwd Pkts/s, Timestamp_hour	t15.1
12	Flow Duration, Flow Pkts/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd Header Len, Bwd Header Len, Fwd Pkts/s, Bwd Pkts/s, Init Bwd Win Byts, Timestamp_hour	t15.2
25	Protocol, Flow Duration, Flow Pkts/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min, Fwd IAT Tot, Fwd IAT Mean, Fwd IAT Max, Fwd IAT Min, Bwd IAT Tot, Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Fwd Header Len, Bwd Header Len, Fwd Pkts/s, Bwd Pkts/s, SYN Flag Cnt, Init Bwd Win Byts, Idle Mean, Idle Max, Idle Min, Timestamp_hour	t15.3

learning techniques is particularly interesting to tackle unknown and zero-day attacks.

In our work, we introduced a novel comprehensive framework to design VANET IDS. Furthermore, unlike most existing works, we use a very recent dataset to evaluate and compare both ensemble learning and standalone learning techniques to detect various types of DOS and DDOS attacks.

For data preprocessing phase, and after data cleaning, normalization, and transformation, we adopted the Synthetic Minority Oversampling Technique (SMOTE) for class balancing; then, we used ANOVA F and Mutual Information for selecting the most relevant features. Afterward, we applied several standalone ML techniques and ensemble ML techniques.

Experiments showed that using SMOTE improves F-score for both standalone and ensemble ML methods. When comparing the two considered feature selection methods, ANOVA F-value and Mutual Information, while varying the number of selected features, we noticed that Mutual Information performs better and is able to identify features that have the strongest impact on the prediction. Moreover, among the four classifiers implemented, RF and ET give the best accuracies by varying the number of features from 10 to 45.

Incorporating ML techniques when designing IDS is undoubtedly efficient, but due to the numerous opportunities that ML techniques offer, more works are still needed to investigate the design of the best ML framework for VANET IDS. For instance, federated learning is a promising approach that can adapt better to the distributed nature of VANET communication by alleviating the vehicle from a big amount of data processing. We intend in future work to investigate this direction.

AQ2

Acknowledgment We would like to thank the research team of the Networks and Systems Laboratory-LRS, Department of Computer Science, Badji Mokhtar University, Annaba, Algeria, for sharing with us their work on the VDOS-LR security dataset.

References

1. A. Ghosal, M. Conti, Security issues and challenges in V2X: a survey. Comput. Netw. **169**, 107093, ISSN:1389-1286 (2020)
2. A. Alnasser, H. Sun, J. Jiang, Cyber security challenges and solutions for V2X communications: a survey. Comput. Netw. **151**, 52–67 (2019)
3. N.A. Alsulaim, R. Abdullah Alolaqi, R.Y. Alhumaidan, Proposed solutions to detect and prevent DoS attacks on VANETs system, in *3rd International Conference on Computer Applications & Information Security (ICCAIS)*, (2020), pp. 1–6
4. K. Stępień, A. Poniszewska-Marańda, Security methods against Black Hole attacks in Vehicular Ad-Hoc Network, in *IEEE 19th International Symposium on Network Computing and Applications (NCA)*, (2020), pp. 1–4
5. J. Montenegro, C. Iza, M.A. Igartua, Detection of position falsification attacks in VANETs applying trust model and machine learning, in *PE-WASUN '20: Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, (2020), pp. 9–16
6. A. Alshammari, M.A. Zohdy, D. Debnath, G. Corser, Classification approach for intrusion detection in vehicle systems. Wirel. Eng. Technol. **9**(4), 79–94 (2018)
7. M.A. Al-Shareeda, M. Anbar, S. Manickam, A. Khalil, I.H. Hasbullah, Security and privacy schemes in vehicular Ad-Hoc network with identity-based cryptography approach: a survey. IEEE Access **9**, 121522–121531 (2021)
8. D. Koo, Y. Shin, J. Yun, J. Hur, An online data-oriented authentication based on Merkle tree with improved reliability, in *2017 IEEE International Conference on Web Services (ICWS)*, (2017), pp. 840–843
9. R. Barskar, M. Ahirwar, R. Vishwakarma, Secure key management in vehicular ad-hoc network: a review, in *International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, (2016), pp. 1688–1694
10. D. Manivannan, S.S. Moni, S. Zeadally, Secure authentication and privacy-preserving techniques in Vehicular Ad-hoc NETworks (VANETs). Veh. Commun. **25**, 100247 (2020) ISSN:2214-2096
11. N. Parikh, M.L. Das, Privacy-preserving services in VANET with misbehavior detection, in *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, (2017), pp. 1–6
12. L. Chen, S. Ng, G. Wang, Threshold anonymous announcement in VANETs. IEEE J. Sel. Areas Commun. **29**, 605–615 (2011)
13. S.S.L. André Perez, *TLS and DTLS Protocols, Network Security* (Wiley). ISBN:9781848217584
14. F.A. Ghaleb, A. Zainal, M.A. Rassam, F. Mohammed, An effective misbehavior detection model using artificial neural network for vehicular Ad hoc network applications, in *IEEE Conference on Application, Information and Network Security (AINS)*, (2017), pp. 13–18
15. P.K. Singh, R.R. Gupta, S.K. Nandi, S. Nandi, Machine learning based approach to detect wormhole attack in VANETs, in *Workshops of the International Conference on Advanced Information Networking and Applications*, (Springer, 2019), pp. 651–661
16. D. Tian, Y. Li, Y. Wang, X. Duan, C. Wang, W. Wang, R. Hui, P. Guo, An intrusion detection system based on machine learning for can-bus, in *International Conference on Industrial Networks and Intelligent Systems*, (Springer, 2017), pp. 285–294
17. S.C. Kalkan, O.K. Sahingoz, In-vehicle intrusion detection system on controller area network with machine learning models, in *11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, (2020), pp. 1–6
18. T.P. Vuong, G. Loukas, D. Gan, A. Bezemskij, Decision tree-based detection of denial of service and command injection attacks on robotic vehicles, in *IEEE International Workshop on Information Forensics and Security (WIFS)*, (2015), pp. 1–6

19. T.P. Vuong, G. Loukas, D. Gan, Performance evaluation of cyber-physical intrusion detection on a robotic vehicle, in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, (2015)
20. L. Yang, A. Moubayed, I. Hamieh, A. Shami, Tree-based intelligent intrusion detection system in internet of vehicles, in *IEEE Global Communications Conference (GLOBECOM)*, (2019)
21. S. Iranmanesh, F. S. Abkenar, A. Jamalipour and R. Raad. A heuristic distributed scheme to detect falsification of mobility patterns in internet of vehicles.. IEEE Internet Things J., 2021.
22. A.R. Gad, A.A. Nashat, T.M. Barkat, Intrusion detection system using machine learning for vehicular Ad hoc networks based on ToN-IoT dataset. IEEE Access **9**, 142206–142217 (2021)
23. D.M. Kang, S.H. Yoon, D.K. Shin, Y. Yoon, H.M. Kim, S.H. Jang, A study on attack pattern generation and hybrid MR-IDS for in-vehicle network, in *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, (2021), pp. 291–294
24. D. Swessi, H. Idoudi, A comparative review of security threats datasets for vehicular networks, in *International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, (2021), pp. 746–751
25. R. Rahal, A. Amara Korba, N. Ghoualmi-Zine, Towards the development of realistic DoS dataset for intelligent transportation systems. Wirel. Pers. Commun. **115**, 1415–1444 (2020)
26. A. Habibi Lashkari, CICFlowMeter (formerly known as ISCXFlowMeter): a network traffic Bi-flow generator and analyzer for anomaly detection 2018. https://github.com/ahlashkari/CICFlowMeter
27. P.D. Gutiérrez, M. Lastra, J.M. Benítez, F. Herrera, Smote-gpu: big data preprocessing on commodity hardware for imbalanced classification. Prog. Artif. Intell. **6**(4), 347–354 (2017)
28. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
29. R. Alshamy, M. Ghurab, S. Othman, F. Alshami, Intrusion detection model for imbalanced dataset using SMOTE and random forest algorithm, in *International Conference on Advances in Cyber Security*, (Springer, Singapore, 2021), pp. 361–378
30. J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: a new perspective. Neurocomputing **300**, 70–79 (2018)
31. A. Thakkar, R. Lohiya, Attack classification using feature selection techniques: a comparative study. J. Ambient Intell. Human. Comput. **1**, 1249–1266 (2021)
32. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, E. Duchesnay, Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
33. U. Coruh, O. Bayat, ESAR: enhanced secure authentication and revocation scheme for vehicular Ad Hoc networks. J. Inf. Secur. Appl. **64** (2022). Elsevier
34. H. Bangui, M. Ge, B. Buhnova, A hybrid machine learning model for intrusion detection in VANET. Computing, Springer (2021)

AUTHOR QUERIES

AQ1. Please check sentence starting "In other hand..." for clarity.
AQ2. Please check the sentence "We intend in future..." for clarity.