Dealing with Smart GPS Spoofing Attacks in VANETs: 3BSM Approach

Zahin Wahab[1], Bishal Basak Papan[1], Md. Shohrab Hossain[1] and Mohammed Atiquzzaman[2]
[1]Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Bangladesh
[2] School of Computer Science, University of Oklahoma, Norman, Oklahoma, USA
Email: zahinwahab@gmail.com, bbasakpapan@gmail.com, mshohrabhossain@cse.buet.ac.bd and atiq@ou.edu

***Abstract*—Nowadays VANETs are being used to ensure road safety and reduce traffic congestion. Vehicles exchange real time position with one another through basic safety messages and these periodic updates must be authenticated for safety purpose. Malicious vehicles can broadcast spoofed position and such an insider attack cannot be dealt with cryptography and digital signature. However, previous machine learning-based approaches that used distributed data-centric misbehaviour detection schemes could detect such attacks in most of the scenarios. However, in case of smarter attacks in a sparse network (where the attacker vehicle pretend to be stand still even though it is on the move), previous solutions could not always detect such attacks. We propose a novel detection scheme which outperforms previously published methods in this specific scenario. For evaluating the performance of this approach, we have used VeReMi dataset, a public repository for the malicious node detection in VANETs. We combine three consecutive basic safety messages and surpass currently existing methods in detecting eventual stop attacks, especially in sparse networks. Our proposed method can help in securing VANET, thereby preventing fatal accidents.**

***Index Terms*—Misbehavior detection, machine learning, position falsification attack, vehicular ad-hoc network (VANETs).**

I. Introduction

Vehicular Ad Hoc Networks (VANETs) are enabling technology useful for wireless communication, providing applications ranging from cargo transport, on-road entertainment streaming to internet access, public safety, traffic management and propagating warning or emergency message. Vehicles share their positions periodically with their neighbours, thus helping other nodes choose route carefully and avoid any unwanted incident. However, malicious vehicles can easily falsify their positions, generate spoofed packets and broadcast them [1]. This is known as position or GPS spoofing attack illustrated in Fig. 1. In order to guarantee that no exchanged messages are inserted or modified by the attacker, cryptography can be used. Public key infrastructure (PKI) supports the distribution and identification of public encryption keys [2], enabling users to encrypt exchanged data over the network and verify the authenticity of the other party. However, any authorized vehicle can broadcast encrypted spoofed packets over the network. Due to digital signature, their identity can also be verified. Hence, detecting position falsification attacks launched by insiders is a challenging task itself.

Heijden et al. [3] published the Vehicular Reference Misbehavior (VeReMi), a publicly accessible extensible dataset in 2018 as a baseline for comparing misbehaviour detection mechanism. Many researchers [4]–[9] ran ML models on the VeReMi dataset. Montenegro et al. [6] introduced trust metrics and plausibility checks as features. Sharma et al. [9] proposed 2BSM approach where two consecutive basic safety messages (BSMs) are concatenated and showed good prospect in detecting position falsification attacks. However, it cannot detect eventual stop attack (where attackers suddenly pretend that it had stopped). Moreover, there might not be enough data for training ML model in low density networks. We have taken these issues of 2BSM approach [9] into consideration for our devised research work. Our objective is to deal with smart attacks especially in sparse networks, where adequate messages are unavailable for feeding into machine learning models. So we augmented three consecutive BSMs together for improved performance, leading to our 3BSM approach.

The contributions of this paper are i) proper pre-processing of the simulation log files of the VeReMi dataset [3], ii) augmenting three consecutive basic safety messages to detect position falsification attack more accurately, and iii) showing density-wise performance comparison between the 2BSM approach and the 3BSM approach for multi-class classification of attacks in all three traffic density types.

For our experimentation, we have provided a comparative study based on VeReMi dataset [3] between 2BSM [9] and 3BSM approach. We used some basic ML models, ran our extensive experimental analysis varying node density - low, medium and high and studied accuracy, precision, recall, and f1-score for both binary and multi-class classification. We analyzed that the 3BSM approach works significantly better in sparse networks as per our hypothesis.

The rest of the paper is organized as follows. In section II-A, we review VANETs topology and system models and in II-B, we summarize some related research works. Then, we present our 3BSM approach for misbehaviour detection in section III. We go through the experiment setup for our approach in section IV, and analyze the results of our experiment in section V. Finally, we conclude the paper in section VI.

II. Background

In this section, we describe the key components of VANETs technology, the adversary model in our problem and finally summarize some of the previous findings on misbehavior detection in VANETs.

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10623066

Authorized licensed use limited to: Purdue University. Downloaded on September 09,2026 at 18:11:27 UTC from IEEE Xplore. Restrictions apply.

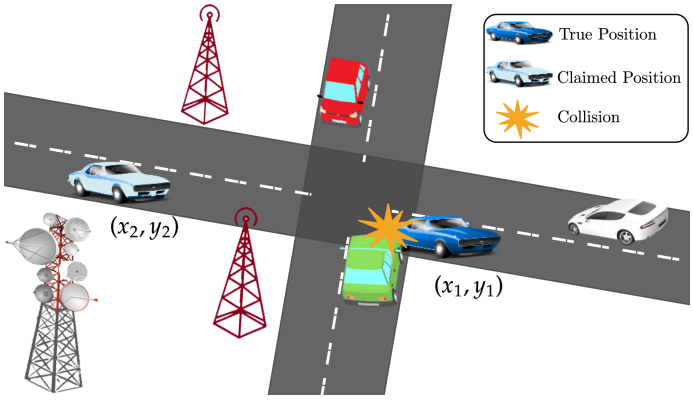


Fig. 1. Causing accident by GPS spoofing attack in a VANET: attacker is in position (x_1, y_1), but claims to be in position to (x_2, y_2) which causes collision with other vehicle.

A. VANETs Technology

Here, we will discuss about technologies used in VANET:

1) System Model: Two different types of plane constitute system model of VANET:

- *Vehicular Plane:* Vehicles have On Board Unit (OBUs) containing GPS for navigation, processors, sensors, antennas and wireless connection.
- *RSU Plane:* Road Side Units (RSUs) have wireless connection that provides connectivity with vehicles in the front end whereas the backend is wired, connected to the gateway via switches and routers used for packet forwarding to the service plane.

Two classes of packets contain physical layer data. OBUs get their current GPS location from the satellite in GPS packets. On the other hand, Basic Safety Message (BSM) packets are broadcasted by each vehicle. This packet contains GPS information of the sender which can be forged by attacker.

2) Communication Model: VANET consists of mobile vehicles that can communicate with one another over the 5.9 GHz frequency band via Dedicated Short Range Communication (DSRC) [10] based device.

3) Adversary Model: In this study, we focus on *active insider attackers* [11]. They can broadcast spoofed position and thus, resulting in chaos, accident etc. In Vehicular Reference Misbehavior (VeReMi) [3] dataset, authors worked with five types of position falsification attacks as discussed below:

1) *Constant attack (Attack Type 1):* Attacker transmits fixed position in the network.
2) *Constant offset attack (Attack Type 2):* Attacker transmits a position with a fixed offset added to its actual position.
3) *Random attack (Attack Type 4):* Attacker transmits random position from the playground.
4) *Random offset attack (Attack Type 8):* Attacker transmits a uniformly random position from a pre-configured rectangle around the vehicle.
5) *Eventual stop attack (Attack Type 16):* Attacker behaves like a legitimate vehicle for specified time interval and then transmits the current position repeatedly in the network (i.e., behaving like it had stopped).

As attack type 16 is smarter than the other types, the 2BSM approach [9] has faced difficulties to detect this type of attack.

B. Literature Review

Due to some special characteristics of VANETs, detection of GPS spoofing attacks in VANETs is a very challenging problem which has already been attended by many researchers.

1) Classification of MDS: According to Heijden et al. [12], misbehaviour detection scheme (MDS) is classified into two types: node centric and data centric. Node-centric mechanisms depend on authenticity and authorization of nodes on the network. But, data-centric mechanisms does not work with nodes' identity rather on the messages sent by the node.

2) ML Based Approaches: ML algorithms are used in a wide variety of applications including VANETs.

a) Trust Metric and Plausibility Checks: Feature vector extracted from VeReMi dataset [3] contains suitable information about sender's behaviour (advertised location, speed). Along with these, So et al. [8] included two plausibility checks,for location and position, as features and four other numerical features. Plausibility checks are important to denote how much a vehicle was spreading forged information about its own coordinate. These updated dataset were finally used to train ML models and label an attacker vehicle.

Singh et al. ran SVM and Logistic Regression on VeReMi dataset [4]. They concluded that speed transmitted by both attackers and non-attackers is in a similar range and thus tagged speed as less important feature. However difference between positions of sender and receiver are crucial because a receiver cannot receive information from a vehicle that is way out of its communication range.

Montenegro et al. [6] applied *trust metric* depending on the transmitted message's data. Gyawali et al. [5] used received beacons (periodic safety messages) to compute new features like change in speed between two beacons, change in position between two beacons, receiving distance, RSSI, change in its own speed, change in its own position etc.

b) 2BSM Approach: Sharma et al. [9] uses an vehicle-RSU pair approach which combines two consecutive BSMs in order to determine whether there has been an attack or not.

C. Gap Analysis

The 2BSM approach [9] trumps single BSM approach for all attack types, except for "eventual stop" attack type (discussed in II-A3). They did not separately evaluate multi-class classification performance in sparse and dense networks.

III. Proposed Approach

Here, we discuss our approach, the dataset used to evaluate our method, advantages and trade-offs associated with it.

In our proposed work, we extend the 2BSM approach [9]. They implemented MDS in RSUs. RSUs communicate with each other and reach an unanimous decision based on incoming data. Rather than relying on a single BSM to

Authorized licensed use limited to: Purdue University. Downloaded on September 09,2026 at 18:11:27 UTC from IEEE Xplore. Restrictions apply.

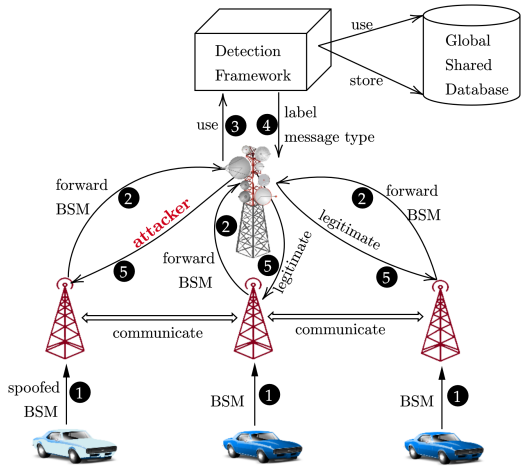


Fig. 2. Architecture for ML based detection of attacker using 3BSM approach

predict its authenticity, this 2BSM approach augments two safety messages sent out by a vehicle consecutively, and prepare a new dataset for ML models. Although this approach trumps single BSM approach but it shows poor performance in detecting eventual stop attack discussed in II-A3. So we wish to improve this performance by introducing 3BSM approach where we prepare augmented dataset by aggregating 3 consecutive BSMs. This 3BSM approach is a data-centric misbehaviour detection scheme. The following steps constitute our proposed approach: (illustrated in fig 2)

- Each RSU collects BSMs sent out by vehicles.
- RSU forwards these BSMs to the center server (or detection framework).
- Detection framework appends 3 consecutive BSMs sent out by one vehicle and run ML algorithm.
- After labelling each vehicle as legitimate or attacker, detection framework passes its decision to the RSUs.
- RSUs notify other vehicles about the potential attacker.

A. VeReMi dataset [3]

VeReMi stands for Vehicular Reference Misbehavior Dataset. It is a publicly available extensible dataset and serves as a baseline to assess how misbehavior detection mechanisms operate on a city scale. VeReMi consists of *message logs* for every vehicle in the simulation and a *ground truth* file that specifies the attacker's behavior. The ground truth file records the transmission time, sender, attacker type, message ID, and actual position/speed vectors each time a message is broadcast. Message logs contain claimed transmission time of the packet by the sender, reception time, claimed position and speed of the sender and RSSI for each receiver. If a message is sent out by legitimate vehicle, it is labeled 0 in ground truth file and similarly, if an attackers sends the message, it is labelled with 1, 2, 4, 8, 16 for five different types of attackers discussed in II-A3 respectively.

3 types of vehicle density has been used in the simulations: a) low (35 to 39 vehicles), b) medium (97 to 108 vehicles, and c) high (492 to 520 vehicles).

The dataset contains 225 different simulations with

- 5 types of attacker
- 3 types of traffic density (low, medium, and high)
- 3 types of attacker density (10%, 20%, and 30%)
- 5 random repetitions for each parameter set

B. Advantage

In eventual stop attack, attackers start with being a legitimate vehicle and then transmit the same position repeatedly as if it had stopped. Now this is not recognizable from a single BSM. Multiple BSMs are required to detect this attack properly. Also, even when an attacker was behaving normally, ground truth file labeled it as an attacker, confusing the machine learning model. 3BSM approach is more likely and suitable to capture this behavioral change.

While dealing sparse networks with limited number of vehicles, it is seen that number of messages broadcast is also less. This small amount of data may not be enough for training a machine learning model. However, concatenating 3 consecutive BSMs increases information carried in one data helping in better prediction by the ML models.

C. Trade-offs

Since this approach needs three consecutive basic safety messages, it might take a little longer for the RSUs to process incoming data. Also, this RSU-based misbehaviour detection scheme needs all the RSUs to communicate with each other thus using up bandwidth and power. The detection model and the shared database of the BSM logs are stored by the central authority of the network which must be secured. If the central authority system is compromised, the security of the whole network will fail. As VeReMi dataset only simulates 2D regions, it can not model road networks in a 3D region.

IV. Experiment

In this section, we present the experimental setup of our method. We start by explaining about how to extract data from raw VeReMi dataset and preprocess the extracted data. Then we talk about ML models run on preprocessed data.

A. Extracting Data

Each vehicle generates a log containing all GPS updates and all received BSMs from other vehicles. Each of these log entries contains a reception time stamp, the claimed transmission time, the claimed sender, an unique message ID, position, speed, RSSI, position noise and speed noise. Ground truth file contains attacker type for each message broadcasted in the system. Since a beacon is broadcast by any vehicle, all vehicles in its communication range receive the beacon. So one particular beacon is logged in more than one vehicle's receiver log. We extract BSMs from all receiver logs and remove duplicate entries. Then we label those BSMs using ground truth file.

Authorized licensed use limited to: Purdue University. Downloaded on September 09,2026 at 18:11:27 UTC from IEEE Xplore. Restrictions apply.

TABLE I
SAMPLE DATASET FOR 3BSM APPROACH

Sender	pos_x^1	pos_y^1	pos_x^2	pos_y^2	pos_x^3	pos_y^3	spd_x^1	spd_y^1	spd_x^2	spd_y^2	spd_x^3	spd_y^3	Type	Label
7	3609.39	5446.80	3605.87	5477.35	3603.26	5507.99	-3.53	30.62	-3.53	30.62	-2.52	30.72	0	0
7	3605.87	5477.35	3603.26	5507.99	3600.75	5538.66	-3.53	30.62	-2.52	30.72	-2.51	30.68	0	0
13	3834.61	5741.17	3836.54	5783.81	3839.70	5826.38	1.81	42.58	2.84	42.55	3.16	42.60	2	1
13	3836.54	5783.81	3839.70	5826.38	3843.20	5868.91	2.84	42.55	3.16	42.60	4.48	42.47	2	1

B. Pre-processing Data

We remove less important features like: RSSI, time, noise vectors for both position and speed. As z coordinate of position and speed is constant, we drop it as well. Then we group all BSMs by their sender and sort them according to the claimed transmission time. For 3BSM approach, we take 3 consecutive packets and augment them to form one instance of our dataset. For each simulation log directory, we create a combined log table by following the above steps. So, features of our 3BSM dataset are Sender, pos_x^1, pos_y^1, pos_x^2, pos_y^2, pos_x^3, pos_y^3, spd_x^1, spd_y^1, spd_x^2, spd_y^2, spd_x^3, spd_y^3, Type and Label. Results of multiclass classification is not published for 2BSM approach in [9]. So for conducting extensive experimentation, we generated data for 2BSM approach as well. Features of our 2BSM dataset are Sender, pos_x^1, pos_y^1, pos_x^2, pos_y^2, spd_x^1, spd_y^1, spd_x^2, spd_y^2, Type and Label.

Here in both the 2BSM and 3BSM datasets, pos_x^1, pos_y^1, spd_x^1, and spd_y^1 are the x and y coordinates of claimed position and speed, respectively, of sender in 1^{st} BSM packet. Same goes for 2^{nd} and 3^{rd} packet as well. Label 0 denotes a legitimate vehicle, and 1, an attacker vehicle for binary classification. For each entry in the table, where $Label = 1$, the value of *Type* can be any of attack types: {1, 2, 4, 8, 16}(discussed in II-A3), needed for multiclass classification. Sample dataset for 3BSM approach is showed in Table. I.

For binary classification, from a collection of 225 initial simulation log tables, for each attack type and traffic density, we merge 15 simulation log tables. For multiclass classification, we perform two analysis:

(i) *Performance in each density:* For each density type, we merge the 75 simulation log tables (15 simulation log tables for each of the 5 attack types).

(ii) *Combined performance over all density types:* We merge all of the 225 simulation log tables. As we don't need the $Type$ column (specifying attacker type) values for binary classification, we drop that column for binary classification. Again, for multi-class classification, we drop the $Label$ column. Then, we split dataset into training and test sets (4:1).

C. Machine Learning Models

We started with very basic ML models such as K-Nearest Neighbours (KNN), Naive Bayes, Decision Tree and Random Forest. These models yield competitive performance.

1) Classification: There are mainly two types of classifications in machine learning:

- *Binary classification:* Here, we consider legitimate vehicles as positive and attacker vehicles as negative class.
- *Multi-class classification:* We consider five different position falsification attacks (discussed in II-A3) as 5 classes.

V. RESULTS

A. Performance Metrics

The metric that first comes to our mind while evaluating binary classification is accuracy, defined as the number of correct predictions (TP +TN) over total number of predictions (TP + FP + TN + FN), given in eqn.1. But it suffers from the accuracy paradox for imbalanced sets.

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

We can use precision and recall here. Here, total number of attacker-transmitted messages is (TP + FN) while number of legit messages is (FP + TN) considering positive as the label for attacker and negative for legitimate vehicles.

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

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

Finally, we consider F1 score as our evaluation metric because F1 score balances precision and recall on the positive class.

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

Parameter Sensitivity: Parameter sensitivity is used to test changes in evaluation metrics depending on the model's hyper-parameter. Here, we hyper-tuned KNN by varying nearest neighbors and Random Forest by varying depth and number of estimators. We observed a decline in f1-score when we increased the number of nearest neighbours in KNN in multi-class classification. This is because increasing the number of nearest neighbours is taking outliers into account. We changed the number of estimators in random forest and highest f1-score is achieved when it is the default value in scikit-learn (i.e. 100). Finally, as we increased the depth, f1-score also increased.

Baseline Parameter: For KNN, K(number of nearest neighbours) is 5. In random forest classifier, number of estimators is 100 by default. Maximum depth was allowed for both random forest and decision tree classifiers.

B. Result Analysis

Here, we present the results of our 3BSM approach and compare them with 2BSM approach [9]. Then, we analyze the results observed for binary and multi-class classification.

Authorized licensed use limited to: Purdue University. Downloaded on September 09,2026 at 18:11:27 UTC from IEEE Xplore. Restrictions apply.

1) Binary Classification: For space constraint, we only show results for attack type 16 in Table II.

- *Low Density Traffic*: From Table II, we see that for type 16 in low density, in 3BSM approach, the F1 scores for kNN, Decision Tree and Random Forest models are 97.5, 96.5 and 97.6 which are better than those of 2BSM approach in the same scenario. So it is evident that our proposed 3BSM approach works better than the 2BSM approach for binary classification in low density traffic.
- *Medium Density Traffic*: In Table II, we see that for type 16 in medium density, F1 scores of 3BSM using kNN, Decision Tree and Random Forest models are 98.5, 98.4 and 98.9 which are higher than those of 2BSM approach.
- *High Density Traffic*: We observe from Table II, that for type 16 in high density, 3BSM approach outperforms 2BSM method for all classifiers.

As Sharma et al. [9] found that the performance of 2BSM approach was significantly worse in binary classification for attack type 16 than the other attack types, they modified the dataset of attack type 16 and tested the performance of 2BSM on this *modified attack type 16*. 2BSM approach performed marginally better in this modified attack type 16 than the original attack type 16. But, we see that, our proposed 3BSM approach performs better in original attack type 16. In table II, we show the difference of performance between the 3BSM approach and the 2BSM approach. And, it is obvious that, for any traffic density and for any machine learning model, the 3BSM approach trumps 2BSM approach in binary classification for attack type 16.

If we look back into the definitions of the 5 attack types, we will understand why 2BSM and 3BSM approaches perform similarly for attack types 1, 2 and 4 and marginally different for attack types 8 and 16. In type 16 attack, the attacker consistently sends its actual position within the BSM packets for some time and then it sends a constant position in the following BSM packets. As a result the receivers of the BSM packets assume that the attacker vehicle has stopped moving. The 2BSM approach performs slightly worse to classify this attack as it uses only two consecutive BSM packets. Our 3BSM approach works better than the 2BSM appraoch in this case as it analyzes three consecutive BSM packets and correlates the speeds and positions sent by the attacker. For binary classification, our 3BSM approach trumps 2BSM in type 16 in every traffic density type.

2) Multi-Class Classification:

a) Comparison on merged dataset: Sharma et al. [9] performed multi-class classification on merged dataset (combining low,medium and high density) using KNN (NN = 3). To compare our performance, we generated confusion matrix on combined dataset with KNN as classifier (illustrated in fig. 3). For type 1 to 4, both of these approaches behaved similarly on merged data. 3BSM showed significant improvement by predicting type-8 attacks correctly for 99% of the time, which is 4% greater than the 2BSM approach. 3BSM classified type 16 accurately in 98% cases, in contrast to 2BSM, which predicted correctly only 94% of the time.

TABLE II
COMPARISON BETWEEN 2BSM AND 3BSM ON ATTACK TYPE 16

Attack Type 16 - Low Density				
Metric	KNN		Naive Bayes	
	2BSM	3BSM	2BSM	3BSM
Precision	96.7	*100*	11.4	*49.4*
Recall	94.2	*95*	100	*82.3*
F1 score	95	*97.4*	20.5	*61.7*
	Decision Tree		Random Forest	
	2BSM	3BSM	2BSM	3BSM
Precision	95.3	*97.7*	97.1	*98.6*
Recall	92.4	*95.3*	93.4	*96.7*
F1 score	94.1	*96.5*	95.2	*97.6*
Attack Type 16 - Medium Density				
Metric	KNN		Naive Bayes	
	2BSM	3BSM	2BSM	3BSM
Precision	96.5	*98.7*	40.8	*51*
Recall	95.4	*98.3*	21.2	*34.8*
F1 score	95.7	*98.6*	27.9	*41.4*
	Decision Tree		Random Forest	
	2BSM	3BSM	2BSM	3BSM
Precision	94.1	*98.0*	96.5	*99.4*
Recall	96.1	*98.7*	95.5	*98.5*
F1 score	95.1	*98.4*	96	*98.9*
Attack Type 16 - High Density				
Metric	KNN		Naive Bayes	
	2BSM	3BSM	2BSM	3BSM
Precision	96.8	*99.3*	53.1	*43.4*
Recall	95.2	*97.4*	11	*26.4*
F1 score	96	*98.4*	18.3	*32.9*
	Decision Tree		Random Forest	
	2BSM	3BSM	2BSM	3BSM
Precision	94.1	*97.3*	96.7	*99.4*
Recall	94.6	*97.7*	94.6	*97.9*
F1 score	94.3	*97.5*	95.5	*98.6*

b) Density-wise comparison: Sharma et al. [9] did not report multi-class classification with respect to density for 2BSM. So we ran 2BSM method on our preprocessed data and reported the results. We used random forest as classifier since this estimator shows significant improvement of 3BSM over 2BSM in multi-class classification. (refer to table III) In low density networks, 2BSM correctly predicts type 8 attack 98.9% of the time whereas 3BSM does that for 100% of the time. 3BSM predicts type 16 significantly better than 2BSM (98%). These results support our hypothesis of superior performance

TABLE III
DENSITY-WISE MULTICLASS CLASSIFICATION:
COMPARISON BETWEEN 2BSM AND 3BSM USING RANDOM FOREST

Density	Attack type	2BSM	3BSM
Low	1	99.79	*99.94*
	2	96	*100*
	4	100	*100*
	8	98.92	*100*
	16	96.11	*98.17*
Medium	1	100	*100*
	2	95.45	*97.97*
	4	99.87	*99.95*
	8	98.7	*99.33*
	16	97.7	*99.25*
High	1	100	*100*
	2	97.96	*99.36*
	4	99.84	*100*
	8	99.24	*99.76*
	16	97.54	*98.44*

Authorized licensed use limited to: Purdue University. Downloaded on September 09,2026 at 18:11:27 UTC from IEEE Xplore. Restrictions apply.

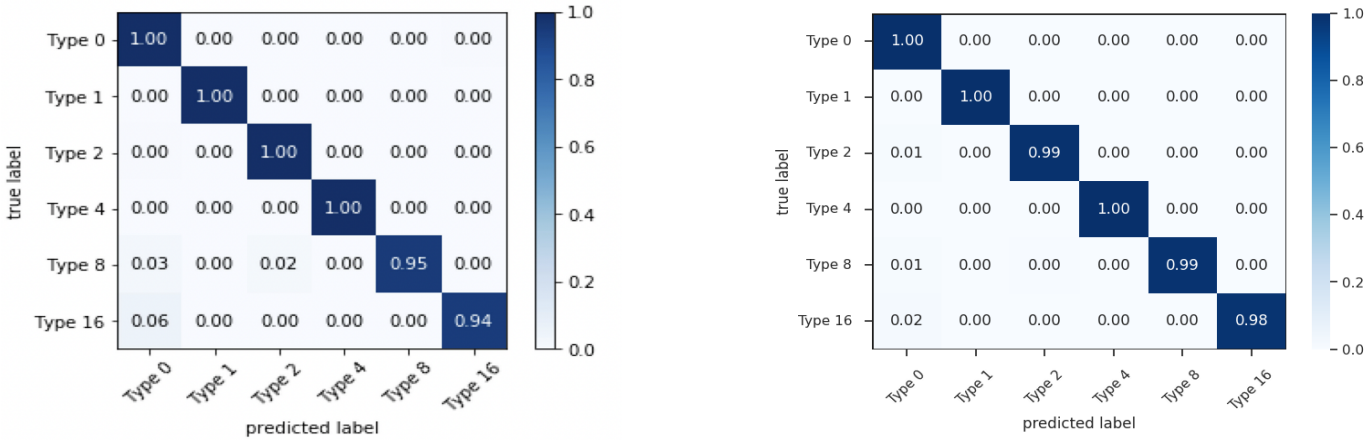


Fig. 3. Multi-class classification: (a) 2BSM Approach [9] and (b) 3BSM Approach

TABLE IV
COMPARISON OF 3BSM AGAINST EXISTING APPROACHES

Attack Type 1 - KNN				
Metric	Approach 1 [8]	Approach 2 [5]	2BSM [9]	3BSM
Precision	95.2	100	100	99.97
Recall	83.2	99	100	99.97
Attack Type 2 - KNN				
Metric	Approach 1 [8]	Approach 2 [5]	2BSM [9]	3BSM
Precision	56.1	94	99.8	99.36
Recall	19.3	80	99.7	99.36
Attack Type 4 - KNN				
Metric	Approach 1 [8]	Approach 2 [5]	2BSM [9]	3BSM
Precision	95	100	100	99.99
Recall	83.6	99	99.8	99.98
Attack Type 8 - KNN				
Metric	Approach 1 [8]	Approach 2 [5]	2BSM [9]	3BSM
Precision	96.2	97	100	99.5
Recall	82.5	95	99.8	99.3
Attack Type 16 - KNN				
Metric	Approach 1 [8]	Approach 2 [5]	2BSM [9]	3BSM
Precision	71.2	98	96.8	99.2
Recall	42.5	93	95.2	98.6

shown by 3BSM in sparse networks. 3BSM trumps 2BSM in detecting all types (except for type 1) in medium density networks. Finally, in highly dense networks, 3BSM shows a superior performance compared to 2BSM.

C. Performance Comparison with other approaches

We considered two approaches ([8] and [5]) and 2BSM to compare against our proposed 3BSM approach (Table IV). Although our method performs best using random forest classifier, for the sake of comparison we considered KNN as the classifier here. Our proposed 3BSM approach showed around 99% recall and precision in all attack types which is a consistently good performance. And in attack type 16, it showed around 2% improvement over 2BSM approach.

VI. CONCLUSION

VANETs are subjects to variety of attacks. This paper presents a novel machine learning-based data-centric misbehaviour detection scheme in VANETs, the 3BSM approach. We used publicly available VeReMi dataset [3] and augmented three consecutive basic safety messages broadcasted by any sender. We experimented with basic ML models and found out that they perform well in both binary and multi-class classification, even with low number of vehicles. To test the performance of our proposed approach in real world scenarios, we need to use more robust dataset.

REFERENCES

[1] C. A. Kerrache, C. T. Calafate, J.-C. Cano, N. Lagraa, and P. Manzoni, "Trust management for vehicular networks: An adversary-oriented overview," *IEEE Access*, vol. 4, pp. 9293–9307, 2016.
[2] H. Hasrouny, A. E. Samhat, C. Bassil, and A. Laouiti, "Vanet security challenges and solutions: A survey," *Vehicular Communications*, vol. 7, pp. 7–20, 2017.
[3] R. W. Heijden, T. Lukaseder, and F. Kargl, "Veremi: A dataset for comparable evaluation of misbehavior detection in vanets," in *International conference on security and privacy in communication systems*. Springer, 2018, pp. 318–337.
[4] 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 *International Conference on Security & Privacy*. Springer, 2019, pp. 166–178.
[5] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.
[6] J. Montenegro, C. Iza, and M. Aguilar Igartua, "Detection of position falsification attacks in vanets applying trust model and machine learning," in *Proceedings of the 17th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 2020, pp. 9–16.
[7] A. Sonker and R. Gupta, "A new procedure for misbehavior detection in vehicular ad-hoc networks using machine learning." *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 11, no. 3, 2021.
[8] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 564–571.
[9] A. Sharma and A. Jaekel, "Machine learning based misbehaviour detection in vanet using consecutive bsm approach," *IEEE Open Journal of Vehicular Technology*, vol. 3, pp. 1–14, 2021.
[10] A. Intl, "Standard specification for telecommunications and information exchange between roadside and vehicle systems-5 ghz band dedicated short range communications (dsrc)," *Medium Access Control and Physical Layer specifications, E2213-03*, 2003.
[11] P. Tyagi and D. Dembla, "A taxonomy of security attacks and issues in vehicular ad-hoc networks (vanets)," *International Journal of Computer Applications*, vol. 91, no. 7, 2014.
[12] R. van der Heijden, S. Dietzel, and F. Kargl, "Misbehavior detection in vehicular ad-hoc networks," *1st GI/ITG KuVS Fachgespräch Inter-Vehicle Communication. University of Innsbruck*, pp. 23–25, 2013.

Authorized licensed use limited to: Purdue University. Downloaded on September 09,2026 at 18:11:27 UTC from IEEE Xplore. Restrictions apply.