A Grid-based Misbehavior Detection System for Vehicular Communication Networks

Chamath Gunawardena*, Owana Marzia Moushi*, Feng Ye†, Rose Qingyang Hu‡, and Yi Qian*,
*Department of Electrical and Computer Engineering, University of Nebraska-Lincoln, Omaha, NE, USA
†Department of Electrical and Computer Engineering, University of Wisconsin-Madsion, Madison, WI, USA
‡Department of Electrical and Computer Engineering, Utah State University, Logan, UT, USA

ICC 2024 - IEEE International Conference on Communications | 978-1-7281-9054-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICC51166.2024.10623059

***Abstract*—A vehicular communication network allows vehicles on the road to be connected by wireless links, providing road safety in vehicular environments. Vehicular communication network is vulnerable to various types of attacks. Cryptographic techniques are used to prevent attacks such as message modification or vehicle impersonation. However, cryptographic techniques are not enough to protect against insider attacks where an attacking vehicle has already been authenticated in the network. Vehicular network safety services rely on periodic broadcasts of basic safety messages (BSMs) from vehicles in the network that contain important information about the vehicles such as position, speed, received signal strength (RSSI) etc. Malicious vehicles can inject false position information in a BSM to commit a position falsification attack which is one of the most dangerous insider attacks in vehicular networks. Position falsification attacks can lead to traffic jams or accidents given false position information from vehicles in the network. A misbehavior detection system (MDS) is an efficient way to detect such attacks and mitigate their impact. Existing MDSs require a large amount of features which increases the computational complexity to detect these attacks. In this paper, we propose a novel grid-based misbehavior detection system which utilizes the position information from the BSMs. Our model is tested on a publicly available dataset and is applied using five classification algorithms based on supervised learning. Our model performs multi-classification and is found to be superior compared to other existing methods that deal with position falsification attacks.**

***Index Terms*—Vehicular networks, grid-based misbehavior detection, network security, machine learning**

I. Introduction

A vehicular communication network consists of wireless multi-hop connections with fast dynamic topology due to high speed mobility from moving vehicles. Vehicular networking is an important component to Intelligent Transportation Systems (ITS) which enables monitoring of road traffic density to optimize transportation system and reduce the number of accidents. Cooperative Intelligent Transportation Systems (C-ITS) allow wireless technologies between two or more ITS sub-systems to communicate for an enhanced ITS service and provide better road safety. Such systems involve vehicle-to-everything (V2X) communications which includes vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P) and vehicle-to-network (V2N) communications collectively. With cellular based V2X communications, vehicles can communicate beyond line-of-sight sensors for much longer range of communications [1].

In vehicular network systems, basic safety messages (BSMs) are used to communicate between vehicles. Vehicles send BSMs out periodically, containing their current location, speed, acceleration, heading, etc. These messages are vulnerable to threats such as attacks on integrity and denial of service attacks. Taking advantage of the vulnerabilities of these messages can lead to impersonating the identities of other vehicles and broadcasting fake traffic warnings to disrupt traffic flow. Such external attacks can be avoided using cryptographic solutions, however, internal attacks where vehicles are already authenticated are much more difficult to detect [2]. A misbehavior detection system (MDS) is efficient in detecting malicious vehicles that are already authenticated in the vehicular network. Although misbehavior detection systems are more efficient in detecting internal attacks on vehicular networks, there still exists challenges of deploying an MDS in a vehicular network. In vehicular networks, there is high speed mobility from moving vehicles which results in a fast dynamic topology of the network. An MDS that is deployed in a vehicular network must adapt to such a dynamic topology as it affects routing, mobility and security. As traffic is occurring throughout the roads, an MDS must be deployed at different points to capture and analyze all of the traffic. To detect these internal attacks, an MDS will involve all participating nodes and make a decision based on aggregated results from data collected through the participating nodes.

One of the internal attacks that an MDS must be able to detect is the position falsification attack where attackers broadcast false position information through the BSMs to disrupt traffic information flow. Machine Learning (ML) has been used for security in vehicular networks and recently, several ML-based misbehavior detection systems have been proposed to detect position falsification attacks in vehicular networks. However, the existing methods make use of a high amount of features which increases the computation complexity and the size of data needed to train their models.

In this work, we propose a grid-based MDS to detect position falsification attacks. In the proposed scheme, a multi-class machine learning based approach is used on grid-based data. The grid based data is taken by plotting all of the transmitted positions for each vehicle and capturing the transmitted positions in a grid. We propose new features based on the grid data to capture transmitted positions of each vehicle. We reduce the dataset size by using only the proposed features based on the grid data from each vehicle instead of using the data from each BSM.

The rest of the paper is organized as follows. Section II gives a literature review on various related work about detecting position falsification attacks in vehicular networks. Section III discusses the system model, dataset, and attack models. Section IV presents the proposed grid-based misbehavior detection system. Section V shows the experimental results and Section VI concludes the paper.

II. Related Work

There are several existing approaches using MDSs to detect insider attacks such as position falsification attacks. In [3], the authors proposed and evaluated ML-based solutions for local misbehavior detection information collected through misbehavior reports (MBRs) which uses On-Board Units (OBUs) and Road-Side Units (RSUs). The detection process is executed by the misbehavior authority (MA) which is a central entity in C-ITS. In [4], the authors proposed a machine learning approach to classify multiple misbehaviors in vehicular networks. The approach makes use of behavioral and concrete features from nodes that are sending out BSM packets. In [5], the authors proposed an ML-based MDS and used the analysis of n consecutive positions from vehicles to create three features. In [6], the authors proposed three physical layer plausibility checks to exploit the RSSI of BSMs. The plausibility checks are able to be run individually by each vehicle and have multi-step mechaisms to raise the detection rate. In [7], the authors proposed a cooperative ML-based MDS to detect false alert attacks and position falsification attacks. The ML-based scheme is used to detect both attacks and is tested on their generated labelled dataset and publicly available datasets. In [8], the authors proposed a ML-based scheme that uses three novel features which are created based on the sender position to detect position falsification attacks. The proposed features include the angle of arrival (AoA) of a message sent from one vehicle to another and the estimated distance between the sender and receiver. The final proposed feature takes the difference of the estimated distance feature and the declared distance between the sender and the receiver. In [9], the authors used ML-based techniques to detect position falsification attacks. The authors showed the efficiency of the ML-based approach on detecting modeled attack patterns. In [10], the authors proposed a method for feature extraction based on the positions of vehicles and detect position falsification attacks with a multi-class classifier. In [11], the authors detected position falsification attacks using a data-centric method through an ML-based MDS. The proposed approach combines information from two consecutive BSMs sent by a vehicle. The existing methods in detecting position falsification attacks for vehicular networks use many features for their schemes, resulting in longer computation time.

In our proposed scheme, we reduce the number of features by introducing a grid-based approach where the features are constructed by using plotted position information of each vehicle. We evaluate our grid-based scheme using different machine learning algorithms through multi-class classification. In addition, we evaluate our grid-based scheme on a publicly available dataset used for evaluation of misbehavior detection mechanisms in vehicular networks. We find that our grid-based scheme have superior performance compared to previous multi-class classification schemes evaluated using the same dataset.

III. System Model, Dataset, and Attack Models

A. System Model

There are three types of MDS in vehicular networks [12]. The first type is the standalone MDS where each vehicle collects its own data and uses its own resources to detect misbehavior in the network. In this type, vehicles do not have information on other vehicles and they make a decision on detection by themselves with no cooperation from other vehicles in the network. The second type of MDS is the cooperative and distributed MDS where vehicles will cooperate with each other to detect misbehavior in the network. This is a distributed architecture, so the decision is made by aggregating the results from other misbehavior detection systems. The third type of MDS is the hierarchical MDS where the network is made up of clusters of vehicles. Each cluster has a cluster head chosen cooperatively between the nodes in the cluster and the cluster head aggregates the results made by the nodes in the cluster to make a decision on detecting misbehavior in the network.

In our system model, we propose a cooperative grid based machine learning scheme to detect the position falsification attacks. In this scheme, each vehicle is equipped with a MDS which makes a decision based on the data collected from the BSMs on the vehicles in the network. The results/decisions from each vehicle are aggregated and used to ultimately remove misbehaving vehicles from the network.

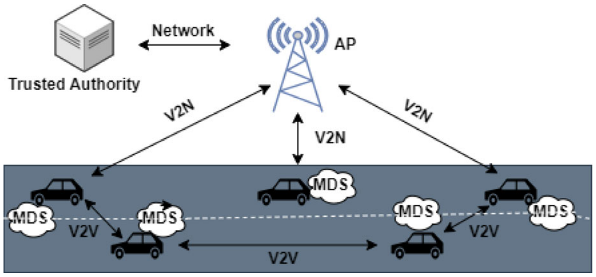


Fig. 1: Vehicular Network Model for ML-based MDS

As shown in Figure 1, the system architecture consists of the Trusted Authority (TA), the Access Point (AP), the vehicles and the MDS. The TA is a trustworthy control center that maintains all significant transactions such as vehicle registration, key management and vehicle verification. The AP communicates between the TA and the vehicles through the cellular communication infrastructure. The MDS is equipped in each vehicle to be active in detecting malicious vehicles upon every BSM arrival. Vehicles periodically broadcast BSMs which include their location, time, speed, etc. Vehicles report to the TA through the APs from misbehavior detection system's decision based on the basic safety messages that the vehicle has received.

B. Dataset

The VeReMi dataset [13] is a publicly accessible dataset for evaluating misbehavior detections in vehicular networks. The VeReMi dataset is a simulated dataset, generated from using LuST [14] and Veins [15]. The LuST scenario is based on a real mid-size European city in Luxembourg and is simulated in the simulation of urban mobility model (SUMO) [16] for generating real world traffic. Veins is an open source framework based on SUMO and OMNET++ [17]. SUMO is a road traffic simulator used to generate real world traffic and OMNET++ is an event based network simulator.

The VeReMi dataset consists message logs of vehicles from many simulations. The message logs are either GPS data about the vehicle or BSM messages received from other vehicles. The dataset consists of five position falsification attacks, three vehicle densities (Low, Medium, High) and three attacker densities (10%, 20%, 30%). Each parameter set is reused five times, resulting in 225 unique simulations.

Each BSM log entry contains a reception time stamp, claimed transmission time, claimed sender, simulation based unique message ID, position vector, speed vector, Received Signal Strength Indicator (RSSI), position noise vector and speed noise vector. Each simulation corresponds with a ground truth file which is updated when a message is sent by a vehicle. The ground truth file contains the transmission time, sender, attacker type, message ID, actual position vector and actual speed vector. The ground truth file contains the actual information about vehicles from message logs while the log entry files may contain fake vehicle information transmitted from malicious vehicles.

C. Attacker Models

The VeReMi dataset includes a set of position falsification attacks of five different types including the constant attack, constant offset attack, random attack, random offset attack and eventual stop attack. For our grid based model, we plot the positions from all the basic safety messages sent by a vehicle onto a grid and try to detect any attack patterns. Figure 2 shows plotted positions from transmitted BSMs of normal vehicle behavior and different types of position falsification attack vehicle misbehavior.

Figure 2a shows the positions sent by a legitimate vehicle in a simulation. The positions sent by the legitimate vehicle seems to form a path in which the vehicle took in the simulation. The points on the grid will be captured and used for our proposed features in the grid based misbehavior detection approach.

The first type of attack is the constant attack where a vehicle transmits a fixed, pre-configured constant position. Figure 2b shows an example of a constant attack where it is shown that the attacking vehicle is transmitting the same fixed, pre-configured position in every BSM.

The second type of attack is the constant offset attack where a vehicle transmits a fixed, pre-configured offset added to their actual position. Figure 2c shows an example of a constant offset attack from a vehicle. Figure 2c has a path that is similar to the path in Figure 2a, however the path in Figure 2c is shifted right on the x-axis and down on the y-axis.

The third type of attack is the random attack where a vehicle transmits a random position within the vehicular area. Figure 2d shows an example of a random attack from a vehicle where the transmitted positions are random and have no clear pattern.

The fourth type of attack is the random offset attack where a vehicle transmits a random offset within the vehicular area that is added to their actual position. Figure 2e shows an example of a random offset attack from a vehicle. Figure 2e is similar to Figure 2d where the transmitted positions are random and have no clear pattern, however, the scale in Figure 2e is much smaller in both axes than the scale in Figure 2d.

The fifth type of attack is the eventual stop attack where a vehicle behaves normally and transmits its actual position in the BSMs and then eventually commits a constant attack on its last transmitted position to act like it has stopped. Figure 2f shows an example of an eventual stop attack from a vehicle. The vehicle behaves normally and transmits its actual position as seen in the first three positions, however, on the fourth position the vehicle acts as if it has stopped and continuously transmits the fourth position for the rest of the simulation.

Rather than directly using the values from the VeReMi dataset, we collect data through these grids on each vehicle and create the proposed features based on this grid data for the proposed grid based MDS.

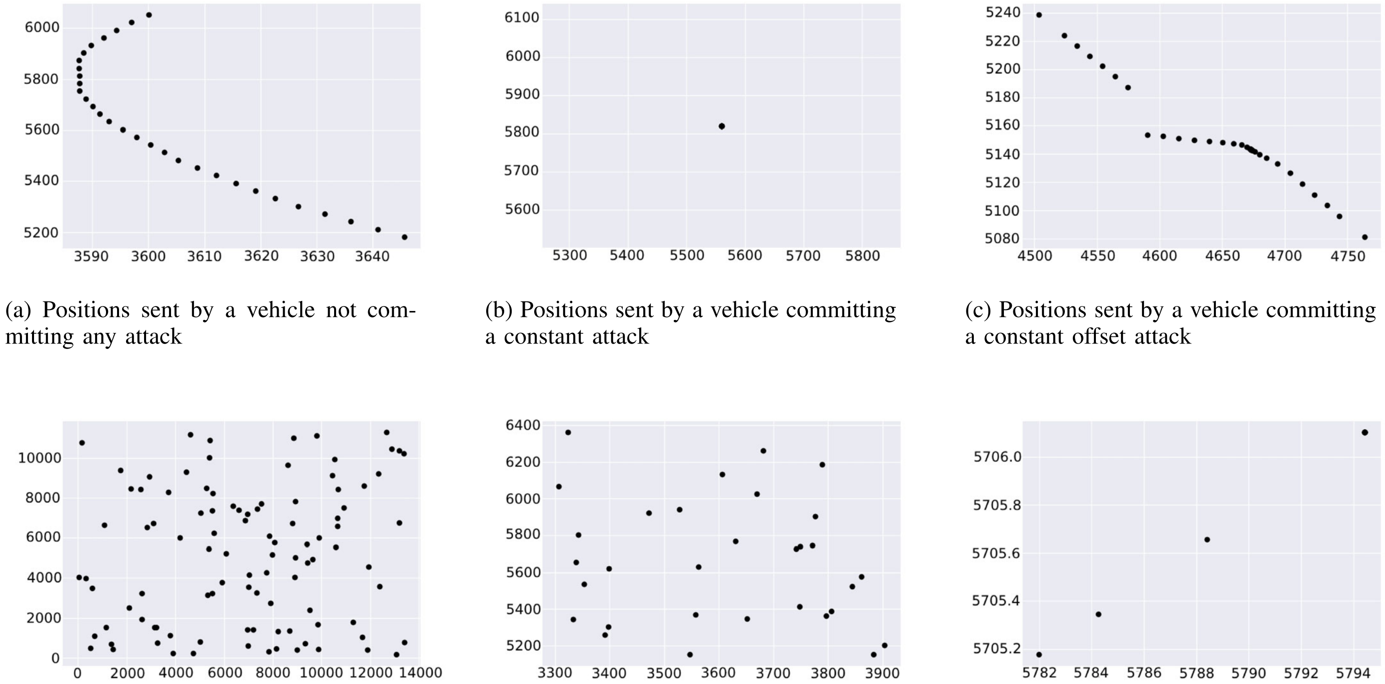
IV. Grid-Based Misbehavior Detection System

In our scheme, we transform the data by taking the positional data from all the BSMs such that each sample represents a vehicle instead to capture the vehicle's positional pattern on the road. Any unusual positions of vehicles where they seem to be well off road would be captured and marked as misbehaving. The positional data for each vehicle is represented on our grid-based scheme in which we create seven features. The proposed MDS involves a grid-based approach to generate seven new features along with one other feature taken from the BSMs. The grid-based MDS uses these features on a multi-class classifier to classify and detect which vehicles are committing the position falsification attacks.

A. Grid Generation

A rectangular grid is created for each vehicle by plotting all of the transmitted positions from the BSMs for each vehicle. A 2-D matrix is created to represent a vehicle's grid and capture the vehicle's transmitted positions.

To normalize our features, the grid for each vehicle needs to be in the same scale on both axes. To address this, let x_{min} be the minimum x-coordinate out of all the transmitted positions in the dataset, let x_{max} be the maximum x-coordinate out of all the transmitted positions in the dataset, let y_{min} be the



(a) Positions sent by a vehicle not committing any attack

(b) Positions sent by a vehicle committing a constant attack

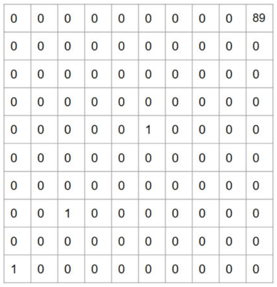
(c) Positions sent by a vehicle committing a constant offset attack

(d) Positions sent by a vehicle committing a random attack

(e) Positions sent by a vehicle committing a random offset attack

(f) Positions sent by a vehicle committing an eventual stop attack

Fig. 2: Plotted positions from transmitted BSMs of normal vehicle behavior and different types of position falsification attack vehicle misbehavior



0	0	0	0	0	0	0	0	0	89
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0

Fig. 3: 10 x 10 Grid of vehicle committing eventual stop attack

minimum y-coordinate out of all the transmitted positions in the dataset and let y_{max} be the maximum y-coordinate out of all the transmitted positions in the dataset. The coordinates: $\{(x_{min}, y_{min}),\ (x_{min}, y_{max}),\ (x_{max}, y_{min}),\ (x_{max}, y_{max})\ \}$ are used as the corners of the rectangular grid. The size of each window by the x-axis is represented by Δx and the size of each window by the y-axis is represented by Δy. Therefore, the size of each window in a grid is Δx by Δy. To create an $n \times n$ grid, we calculate Δx and Δy as follows:

$$\Delta x = \frac{x_{max} - x_{min}}{n} \tag{1}$$

$$\Delta y = \frac{y_{max} - y_{min}}{n} \tag{2}$$

Let K represent the set of all vehicles in the dataset, and V^k ($k \in K$) represent an $n \times n$ matrix in which, the (i, j)th entry of V^k, $v^k_{i,j}$, represents the total number of transmitted positions for the window in the ith row and jth column of the grid.

Let M^k be the set of all messages transmitted by the vehicle $k \in K$. We initialize V^k to a zero matrix. For each message $m \in M^k$, let (x_m, y_m) be the BSM transmitted position. We can find the corresponding window by calculating (i, j) as follows:

$$i = \left\lfloor \frac{y - y_{min}}{\Delta y} \right\rfloor \tag{3}$$

$$j = \left\lfloor \frac{x - x_{min}}{\Delta x} \right\rfloor \tag{4}$$

We increment $v^k_{i,j}$ by 1.

Figure 3 shows a 10 x 10 matrix corresponding to the grid of the vehicle's positions in Figure 2f. The grid shows that the first three BSMs from this vehicle are legitimate with three different positions transmitted once each but the fourth position is transmitted 89 times representing an eventual stop attack.

B. Proposed Features

Rather than using the message log data, we use the grid data for each vehicle and propose new features to greatly reduce the size of the dataset.

The first new feature, p_k, corresponds to the total number of positions from all BSMs transmitted for any vehicle k (i.e., the sum of all the entries in V^k):

$$p_k = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} v_{i,j}^k \tag{5}$$

The second new feature, w_k, represents the number of windows in any vehicle k's grid that contains points. Let a be an integer. We define a function X such that $X(a) = 1$, if $a > 0$, and $X(a) = 0$ otherwise.

$$w_k = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} X(v_{i,j}^k) \tag{6}$$

The third feature is called the spread ratio (sr) which is between the second (number of windows containing points) and first feature (total points) for any vehicle k ($sr_k = \frac{w_k}{p_k}$). sr of a vehicle captures how spread out the vehicle's transmitted positions are on its grid.

We define a binary $n \times n$ attacking matrix A for the grid using the whole training set. We set $a_{i,j} = 0$ if $v_{i,j}^k > 0 \, \forall k \in K$ where $v_{i,j}^k$ has at least one position from a benign vehicle, and $a_{i,j} = 1$ otherwise. The windows with $a_{i,j} = 1$ are considered attacking windows which may include empty windows. The fourth feature, q corresponds to the total number of points in attacking windows for any vehicle k's grid.

$$q = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} v_{i,j}^k a_{i,j} \tag{7}$$

The fifth feature is called the attack ratio (ar_k) which is the ratio between the fourth (total number of points in attacking windows) and first feature (total points) for any vehicle k ($ar_k = \frac{q_k}{p_k}$). With the attack ratio, the vehicle positions outside of the windows containing positions from benign vehicles are captured such as positions that are clearly not on the road.

The sixth feature represents the average absolute difference between all consecutive points for any vehicle k.

Finally, the seventh feature corresponds to the average absolute speed from all the BSMs sent from any vehicle k. Any parked vehicle will be differentiated from a constant attack with this feature because its average speed will be 0 while a constant attacking vehicle will have its actual speed transmitted.

V. Performance Evaluation

To evaluate the proposed grid-based MDS system, we have trained and tested on five classification models using the VeReMi dataset. The models were trained and tested with multi-classification and compared to other existing approaches that detected position falsification attacks on the VeReMi dataset using multi-classification.

A. Evaluation Metrics

The two main metrics used to evaluate the performance of the proposed scheme are model accuracy and F1 score. The model accuracy is the ratio between the number of correctly predicted observations and the number of total observations.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives and FN is the number of false negatives.

Precision is the ratio between the number of correctly predicted malicious vehicles and the total number of predicted malicious vehicles: $\frac{TP}{TP+FP}$. Recall is the ratio between the number of correctly predicted malicious vehicles and the total number of actual malicious vehicles: $\frac{TP}{TP+FN}$. The F1 score is the weighted average of precision and recall.

$$F1score = 2 \times \frac{precision \times recall}{precision + recall} \tag{9}$$

B. Multi-class Classification

The grid-based scheme is evaluated using multi-class classification where each class of position falsification attacks is classified and detected. Normal vehicles are labeled as 0, vehicles that commit a constant attack are labeled as 1, vehicles that commit a constant offset attack are labeled as 2, vehicles that commit a random attack are labeled as 4, vehicles that commit a random offset attack are labeled as 8 and vehicles that commit an eventual stop attack are labeled as 16.

The VeReMi dataset is used to create our grid based data which is shuffled into training (70%) and test (30%) sets for evaluating the model. Five classification algorithms are used to evaluate and test the grid based data including Decision Tree (DT), Random Forest (RF), K-Nearest Neighbor (KNN), Naive Bayes (NB) and Logistic Regression (LR). Five different grid sizes ($n = \{20, 40, 60, 80, 100\}$) are used to evaluate the impact of window sizes on the detection performance.

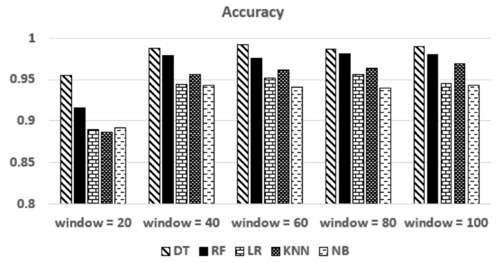


Fig. 4: Model Accuracy

In Figure 4 and Figure 5, the accuracy and F1 score are shown for each classification algorithm at different grid sizes. As n increases, both the accuracy and F1 score increase, however, at $n = 60$, the accuracy and F1 score plateau and increasing n any further does not seem to affect the model

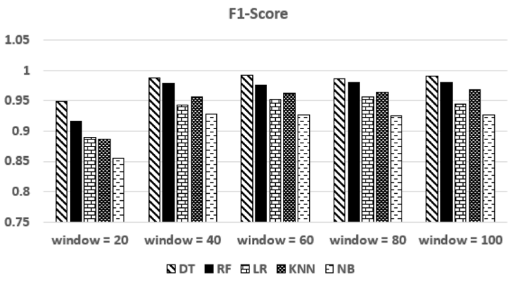


Fig. 5: Model F1-score

Metric	F1-score	Precision	Recall
Ref [10]	0.744	0.744	0.744
Ref [7]	0.954	0.978	0.932
Ref [11]	0.985	0.988	0.981
Proposed	0.992	0.993	0.990

TABLE I: Comparing other multi-class approaches

performance. This may be due to the window size being small enough at $n = 60$ such that decreasing the window size further won't capture any points that the window size at $n = 60$ already has captured. The reason that the accuracy and F1 score increase from $n = 20$ is because the window size is much bigger at $n = 20$ and therefore will not be able to capture the transmitted position patterns of as many vehicles as the window size at $n = 40$ would. Since the scales of the plotted transmitted positions for each vehicle is different, some vehicles' positions with smaller scales may all fit into one window while other vehicles' positions with larger scales will be more spread out on the grid. As for the classification algorithms, the decision tree model performed the best while the Naive Bayes model performed the worst.

Table I shows a comparison of the proposed scheme with other multi-class approaches in detecting position falsification attacks, using the VeReMi dataset. The proposed grid based approach outperforms the other multi-class approaches at $n = 60$ (number of windows = 3600) using the Decision Tree model. We find that $n = 60$ seems to be the optimal value for model performance as decreasing n under performs and increasing n does not improve the detection performance.

VI. Conclusion

Vehicular networks can be vulnerable to insider attacks such as position falsification attacks and misbehavior detection systems are found to be efficient in detecting such attacks. In this paper, we proposed a grid-based MDS by creating a dataset with new features based on plotting vehicles' positions onto a grid and detecting attack patterns by capturing the positions in the windows of the grid. We found that our proposed scheme is effective for multi-classification of position falsification attacks and obtained a high accuracy and F1 score at a grid size of 3600. We found that our scheme out performs existing multi-class approaches for position falsification attacks at $n = 6$ using the Decision Tree model. There is still room for improvement as we can create new features with the proposed grid based model that targets specific position falsification attack types and also omit features of less importance.

Acknowledgement

This work was partially supported by National Science Foundation under grants CNS-2008145, CNS-2007995, CNS-2319486, CNS-2319487, CNS-2344341.

References

[1] 5G America. 5g americas white paper: Cellular v2x communications towards 5g, 2018.

[2] Fengzhong Qu, Zhihui Wu, Fei-Yue Wang, and Woong Cho. A security and privacy review of vanets. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):2985–2996, 2015.

[3] Issam Mahmoudi, Joseph Kamel, Ines Ben Jemaa, Arnaud Kaiser, and Pascal Urien. *Towards a Reliable Machine Learning-Based Global Misbehavior Detection in C–ITS: Model Evaluation Approach*, pages 73–86. 04 2020.

[4] Jyoti Grover, Nitesh Prajapati, Vijay Laxmi, and Manoj Gaur. Machine learning approach for multiple misbehavior detection in vanet. volume 192, pages 644–653, 07 2011.

[5] Anhtuan Le and Carsten Maple. Shadows don't lie: n-sequence trajectory inspection for misbehaviour detection and classification in vanets. In *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pages 1–6, 2019.

[6] Steven So, Jonathan Petit, and David Starobinski. Physical layer plausibility checks for misbehavior detection in v2x networks. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '19, page 84–93, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Sohan Gyawali and Yi Qian. Misbehavior detection using machine learning in vehicular communication networks. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, 2019.

[8] Secil Ercan, Marwane Ayaida, and Nadhir Messai. Misbehavior detection for position falsification attacks in vanets using machine learning. *IEEE Access*, 10:1893–1904, 2022.

[9] Pranav Kumar Singh, Shiv Prakash Gupta, Ritveeka Vashistha, Sunit Kumar Nandi, and Sukumar Nandi. Machine learning based approach to detect position falsification attack in vanets. In *ISEA Asia Security and Privacy Conference*, 2019.

[10] Faisal Hawlader, Abdelwahab Boualouache, Sébastien Faye, and Thomas Engel. Intelligent misbehavior detection system for detecting false position attacks in vehicular networks. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2021.

[11] Aekta Sharma and Arunita Jaekel. Machine learning based misbehaviour detection in vanet using consecutive bsm approach. *IEEE Open Journal of Vehicular Technology*, 3:1–14, 2022.

[12] Sara Chadli, Mohamed Emharraf, Mohammed Saber, and Abdelhak Ziyyat. The design of an ids architecture for manet based on multi-agent. In *2014 Third IEEE International Colloquium in Information Science and Technology (CIST)*, pages 122–128, 2014.

[13] Rens W. van der Heijden, Thomas Lukaseder, and Frank Kargl. Veremi: A dataset for comparable evaluation of misbehavior detection in vanets, 2018.

[14] Lara Codeca, Raphael Frank, Sebastien Faye, and Thomas Engel. Luxembourg sumo traffic (lust) scenario: Traffic demand evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63, 2017.

[15] Christoph Sommer, David Eckhoff, Alexander Brummer, Dominik Buse, Florian Hagenauer, Stefan Joerer, and Michele Segata. *Veins: The Open Source Vehicular Network Simulation Framework*, pages 215–252. 05 2019.

[16] Daniel Krajzewicz, Georg Hertkorn, Christian Feld, and Peter Wagner. Sumo (simulation of urban mobility); an open-source traffic simulation. pages 183–187, 01 2002.

[17] András Varga and Rudolf Hornig. An overview of the omnet++ simulation environment. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Simutools '08, Brussels, BEL, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).