Privacy Preserving Misbehavior Detection in IoV using Federated Machine Learning

Aashma Uprety, Danda B. Rawat and Jiang Li
Data Science & Cybersecurity Center, EECS Dept., Howard University, Washington DC, USA

***Abstract*—Data falsification attack in Vehicular Ad hoc Networks (VANET) for the Internet of Vehicles (IoV) is achieved by corrupting the data exchanged between nodes with false information. Data is the most valuable asset these days from which many analyses and results can be drawn out. But the privacy concern raised by users has become the greatest hindrance in performing data analysis. In IoV, misbehavior detection can be performed by creating a machine learning model from basic safety message (BSM) dataset of vehicles. We propose a privacy-preserving misbehavior detecting system for IoV using Federated Machine Learning. Vehicles in VANET for IoV are given the initial dull model to locally train using their own local data. On doing this we get a collective smart model that can classify Position Falsification attack in VANET using the data generated by each vehicle. All this is done without actually sharing the data with any third party to perform analysis. In this paper, we compare the performance of the attack detection model trained by using a federated and central approach. This training method trains the model on a different kind of position falsification attack by using local BSM data generated on each vehicle.**

***Index Terms*—VANET, Federate Learning, Data Falsification Attack, Privacy**

I. INTRODUCTION

In IoV, VANET is a self-managed network for vehicles whose main motive is to improve the safety of driver and passenger. Intercommunication within the network helps to pass safety and traffic related information among the vehicles. For instance, VANET transforms accident related information to the network so that other vehicle drivers get aware of that situation and can plan the safety accordingly. In general, VANET provides two types of service i.e incident based services and cooperative knowledge sharing services. In incident based services, nodes in the network communicate information when an incident occurs. And in cooperative knowledge sharing applications, nodes aware the neighboring nodes about any dangerous situation ahead. VANET can help the driver make travel plan according to traffic congestion. Additionally, drivers are cognizant about any road accidents and unfavorable conditions ahead of time, and hence driving is safer. However, many security issues are associated with this network. The information communicated by beacons to each other contains the position of the vehicle, their speed, and much other critical information. The position of vehicle generating the broadcasted messages has a significant role in realizing this self managed network. Therefore, attackers manifest these position features to disrupt the VANET network. Creating an illusion of a false position to gain the desired attacking goal is called the Position Falsification Attack. The misbehaving node or the attacker node can create multiple illusion of false position and can severely take control of the traffic in VANET.

A. VANET Model in IoV

The main unit in VANET is On-Board Unit (OBU) which is present in each vehicle and static unit called Road Side Unit (RSU). Each vehicle in VANET, called nodes, are installed with multiple sensing and computing devices. These units facilitate the exchange of information between nodes in this environment. Each vehicle broadcasts information periodically as Basic Safety Messages (BSM). Messages include warning related to collision, accidents, lane-change information, navigation information, and other traffic related messages. Vehicles communicate with each other (V2V) and to other infrastructures like roadside unit (V2I). V2V communication is the most notable communication to realize this network. Vehicles communicate using Dedicated Short Range Communication (DSRC) standardized by IEEE 802.11p [4].

B. Data Falsification Attack in VANET

Cyber attacks in any domain are classified as a node-centric attack and a data-centric attack. The node-centric attack is performed on the source of information itself. On the other hand, in the data-centric attack, the information is attacked. In the context of VANET, trust in data is more important than trust in the vehicle generating that data. Alert data and beacons broadcasted by each vehicle through BSM carries critical safety information. Therefore, false BSM data can jeopardize the whole VANET network. Position falsification attack is one of the serious data-centric attacks in VANET. The attacker node broadcasts false position information in BSM to fool other vehicles about their real locations.

C. Federated Machine Learning

Google proposed a novel training approach to collaboratively learn from multiple devices [6]. This approach called federated learning can jointly train the model on multiple nodes without actually collecting data. The overall benefit of this approach is privacy preservation and learning from a large number of nodes. This training approach is beneficial because of the following reasons.

1) Privacy Preserving: Data privacy is a serious issue at this time. Users are hesitant to share their data with any third party. Many regulations and restrictions appear to collect data to perform any analysis. As a solution, federated training supports data training on multiple nodes by locating models to data sources. No one ever sees the data from nodes.

2) Reduced Communication Overhead: In the FL setting, only model parameters and gradients should be communicated with the central server. Also, it can choose clients based on bandwidth availability. This facilitates efficient communication [2].

3) Train on large datasets: Because of no requirement to transfer data from nodes to the server, federated training can train on large datasets by sending the model to data sources. Training on large datasets results in higher model accuracy.

II. Related Work

Data falsification attack is one of the highly researched cyber attacks in VANET. The proper functioning of this network relies on the integrity of data transmitted. So, different detection methods for this kind of attack has been well proposed by researchers. In the paper, the authors proposed a technique to detect a data falsification attack using hash chains by adjusting the contention window size which transfers accurate information to the neighboring vehicles in a timely manner [7]. Malicious data can badly affect the VANET environment. Early detection of such data can save the network from future damage and other costs. Here in work [3], the authors proposed malicious data detection and correction method. The VANET node scores the collected data based on possible explanations. The node then validates the data with the best scoring explanations. Machine learning is a promising method to detect patterns from data to develop a detection system. Because VANET can have numerous data due to a highly scalable network, machine learning techniques are highly studied in literature to find attacks in VANET. Authors in work [9] integrate plausibility check with machine learning to detect misbehaving vehicles in VANET. Here authors train the model using SVM and KNN algorithm to get a detection model. For the model to be trained more accurately, data size plays a high influence. The more the dataset, the more is the learning capacity of these algorithms. However, centrally collecting huge datasets is infeasible in the context of communication cost and privacy issues. The aforementioned issue is resolved by our approach of federated training on BSM data generated by nodes locally.

III. Problem Statement

Privacy issue these days is the most raised issue to the authority that perform machine learning using users dataset. In the VANET environment, there is always a trade-off between privacy and security. The fact is users are more privacy concerned than security. VANET is a highly mobile and dynamic networked environment and it generates a huge number of data. Machine learning can find meaningful information using these datasets. In our case, using the BSM dataset, we can learn a model to detect misbehaving vehicles in the network. However, privacy issues arise with the centralized learning approach. The goal of our research is to learn a misbehavior detection model that can find attacker vehicles by analyzing BSM data received from them. The main requirement is to do all this by keeping the users' data secure without any privacy breach. Researchers have proposed a misbehavior detection system using a simulated dataset. Due to the high communication cost to collect an enormous number of data generated by real VANET, researchers use simulated data. Also, data sharing is strictly monitored by law to protect user privacy. Using a centralized training approach, it is quite difficult to train on real VANET data. Additionally, bandwidth consumption and privacy is another problem. As a solution to this problem, we propose a federated learning-based misbehavior detecting system.

A. Assumptions

We assume the training process to be synchronous. This means all the local nodes updates their local model and simultaneously send it to the central authority [8].

IV. System Model

A misbehavior detection system using Federated Machine Learning is obtained by local training of the model using BSM data generated on vehicles. This training approach is used to make the model smarter without actually sharing the data with any central authority. Our model aims to detect any misbehaving vehicles in VANET by training on local BSM data generated by each vehicle. In this scenario, both the attacker vehicle and the legitimate vehicle are broadcasting BSM to communicate over the network. A large number of the vehicle generates a large amount of BSM data even in a shorter duration of time. These huge number of data generated by mobile nodes gives a greater possibility for the analyst to perform data analysis.

A. Entities Involved

Our proposed Federated training approach includes a Central Authority (CA) and vehicles (V) as nodes for local training. CA can be any third party that is training a misbehavior detecting model. This entity holds the initial dull model which is later sent to local nodes for training. Initially, there is no availability of training data. The initial model is trained by using proxy data or the model parameter is set which is later trained using local data. CA sends this model to local vehicles that are involved in local training. In our detection scenario, each vehicle uses the local BSM data received from all neighboring vehicles. BSM data received by each training vehicles are locally used to update the model parameters. These all updated models are sent to the averaging entity or CA which performs federated averaging of all local weights and gives a more smarter model. This whole process is performed in a single round. Multiple rounds are repeated until the desired model accuracy is obtained.

B. Federated Learning Approach

In the context of VANET, each vehicle broadcasts BSM (Basic Service Message). BSM carries critical and private information like current speed, location, etc. Using this BSM data, machine learning technology can learn smart decisions like routing decisions, safety increasing guidance, and many more. With these advantages, there come issues like privacy and security threat. Sharing the BSM data generated by vehicles for training the model is like somehow sharing the personal information to potential attackers. Attackers can also exploit the machine learning techniques to draw some patterns and conclusions about the vehicle owner by playing with the BSM data. Cyber Attacks like Sybil Attack, Denial of Service Attack, Jamming Attack in VANET are some of the well-studied topics. In our work, we look at the data falsification attack. In a position falsification attack, attackers send false data to mislead the vehicles running on the roads. Detection of this attack and the removal of attacking vehicles from the network is a critical step.

As a solution, we apply federated learning approach that lets the data reside on the vehicle and can locally train the ML model in the vehicle without sharing the information to central training authority. In figure [1], steps involved in the training process are shown.

Steps involved are:

1) The main initial model is initialized by using some parameters.
2) This model is then shared to certain group of vehicles which are randomly selected for locally updating the weights of the model.
3) Selected vehicles use their local BSM data to train the model and update the gradients of the model. On doing this no vehicles should share the real BSM data to server or central training authority.
4) All the vehicles send their individual local models and a federated averaging is performed on all local models which gives a central aggregated smarter model.
5) These steps are repeated for certain rounds until a desired or acceptable accuracy is reached.

This mechanism is performed without actually looking at BSM data hence protecting the privacy of the data. Local model training is done in each participating vehicles without requiring any outsider look at the data.

C. Federated Averaging

After local training of the detection model using BSM data in local nodes, all model weights are aggregated to form an updated model.This averaging is achieved by using Federated Averaging algorithm [5]. In the presence N clients, p_k is the weight of client k and n_k is the number of training data present in client k The optimization function can be expressed as below,

$$f(w) = \sum_{k=1}^{N} p_k F_k(w) \text{ where } F_k(w) = \frac{1}{n_k} \sum_{i=1}^{n_k} f_i(w)$$

Here $f_i(w)$ is the loss on prediction made by model with weight w on data (x_i, y_i).

Each training node performs a step of gradient descent weight update on the initial model by using its local data. The server then calculates the weighted average of the local updated models. The main parameters of this algorithm are the number of clients selected for participation in training process, epoch used by local client to train in its local dataset. And, the minibatch size used for client updates. Generally, minibatch size is chosen to be 1.

TABLE I
VeReMi Dataset

Attack ID	Type of Attack	Description
1	Constant Attack	Fixed false location is transmitted by attacker
2	Constant Offset Attack	A fixed offset added to real position is transmitted
4	Random Attack	A random position inside the simulation area is transmitted uniformly by attacker
8	Random Offset	A random position in a preconfigured rectangle around the vehicle is transmitted by attacker
9	Eventual Stop	Attacker vehicle acts normal for some time, later transmits a constant current position

V. Experimentation

We use a publicly available dataset VeReMi to perform our experimentation. In this work, we train on this dataset by using both federated and traditional centralized training approach. The results of this training experimentation will be discussed in next section.

A. VeReMi Dataset

We use VeReMi dataset for our experimentation, which is publicly available dataset to analyze misbehavior detection mechanism in VANET. It is a simulated dataset which contains message logs of GPS data of the vehicle and BSM data received from other vehicles communicated through DSRC. Ground truth file for every message is present in this dataset. A total of 225 individual simulations are performed to generate this dataset in the presence of 5 types of attackers, varying attacker densities and traffic densities and also few repetitions on each parameter. Five types of position falsification attack is simulated.

These are the constant attacker, the constant offset attacker, the random attacker, the random offset attacker, and the eventual stop attacker. A short description of these attacks is given in the table below. We preprocess the dataset by performing some data cleaning. The dataset consists of message logs for each receiving vehicle from senders within the range of 300 meters during its entire journey. ML-friendly VeReMi [9] is the preprocessed version of the original VeReMi dataset to perform machine learning evaluation on the data. This dataset is made available to perform other machine learning analysis. Feature vector are included in the dataset by performing plausibility checks. This feature vector includes information about the behavior of the sender during the entire journey of it when it was inside the communication range of the receiver. We use 4 feature vectors as columns in the dataset. Feature 1 and 2 are the difference of calculated average velocity and predicted velocity based on reported velocity by sender in x and y direction respectively. Feature 3 is the magnitude of features 1 and 2. Last feature is the difference between calculated total displacement and the predicted total displacement based on average velocity.

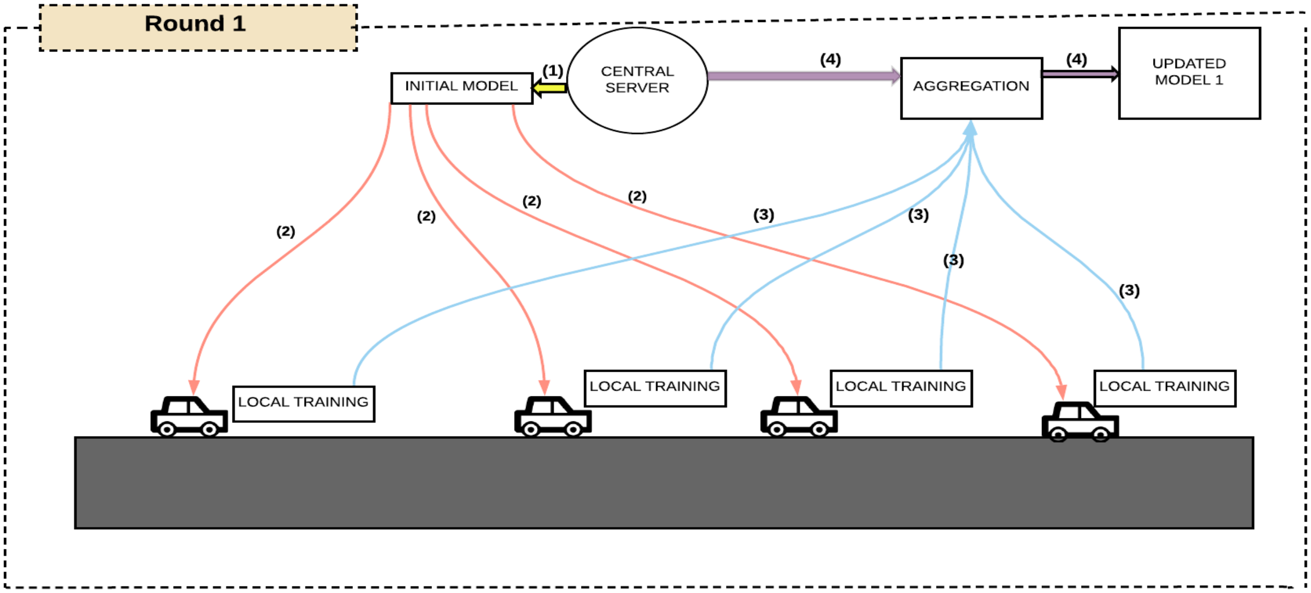

Fig. 1. Single Round Federated Training in VANET for IoV.

TABLE II
FEATURES IN THE DATASET

Features	Type	Detail
1	Numerical Feature	Difference of calculated average velocity and predicted velocity based on reported velocity by sender in x direction
2	Numerical Feature	Difference of calculated average velocity and predicted velocity based on reported velocity by sender in y direction
3	Numerical Feature	Magnitude of features 1 and 2. Represents Constant Offset Attack
4	Numerical Feature	Difference between calculated total displacement and the predicted total displacement based on average velocity.

B. Federated Training

We have used Tensorflow Framework to perform this experiment. Initially, we centrally get the dataset and distribute it among the number of nodes. This is for experimentation purposes only. In a real scenario, no central data collection and distribution is required. Each vehicular node has its own local data set. But here we simulate the federated scenario by distributing the dataset to vehicular nodes. Here each vehicular node selected for training will be simulated virtually. They act as virtual workers and simulated as if they are training separately. All these simulations are performed in a single machine by the virtual assumption of separate workers.

First, we create a set of vehicles to act as Virtual Worker. For the experimentation purpose, we initialize 10 vehicles as local nodes.

A different algorithm like support vector machine (SVM), KNN, LSTM have been used as a training algorithm in federated learning. Here, we perform the experimentation using an Artificial Neural Network (ANN). In the federated simulation, data distribution should follow the non-iid property. To give non-iid property to local data on each vehicle, we distribute the data of different sizes to the local vehicles. This mimics the unbalancedness category of non-iid data.

1) Federated Learning Parameter: During our analysis we have used following learning parameter to train in federated manner.

R (Rounds number) = 500
t (Number of nodes selected for a round) = 10
B (Batch size used at local nodes) = 16
N (Epoch number at local nodes) = 1
LR (Local learning rate) = 0.01

Each local vehicles selected, train the initial ANN model using their local BSM data. The number of the local model update is equal to the chosen epoch size. Federated averaging of all local weights gives a global updated model. We repeat this step for 500 rounds. In the coming section, we have discussed the result of this experimentation.

C. Traditional Centralized Training

We have performed the experimentation on a traditional centralized training manner to perform some comparative analysis. We trained an ANN model using the same parameter as in federated training. Here we split the entire dataset into test data and training data. Using the training data, we trained ANN model for 500 epochs. Comparative analysis between two approaches will be discussed below.

VI. RESULTS AND EVALUATION

A. Evaluation Metrics

We use Precision, Recall and Accuracy to evaluate the performance of these algorithms. Precision is the proportion of positive findings that are actually correct. Recall is the proportion of actual positive that are identified correctly. And accuracy is the measure of number of correctly predicted data out of all data points.

Precision = $\frac{TP}{TP+FP}$ and Recall = $\frac{TP}{TP+FN}$

The TP called True Positive implies that attacker vehicle are detected as attacker. FP means legitimate vehicle are detected as attacker and FN means attacker was not detected as attacker. We use mean Average Precision (mAP) as the metrics to evaluate the detection accuracy. In this approach, AP is calculated for each attack classes and all these APs are averaged. Average Precision is the area under the precision-recall curve.

$mAP = \frac{\sum_i^n AP_i}{n}$, n is the number of attack classes.

1) Accuracy, Precision and Recall: Here we analyze the overall accuracy of attack detection achieved by training using two approaches. The accuracy plot in [2], clearly shows that the overall accuracy achieved by federated training is higher than centralized training

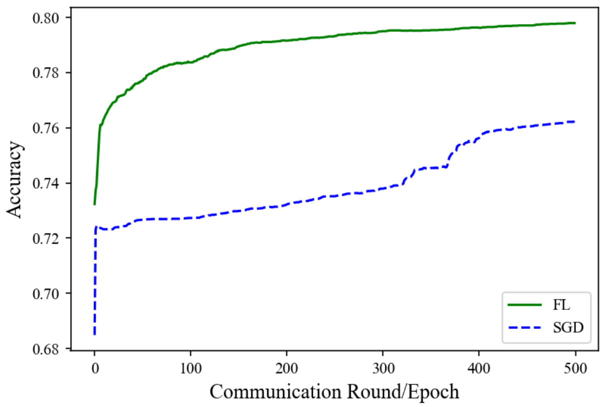

Fig. 2. FL vs SGD Accuracy

Also we have analyzed the precision and recall value achieved for each attack using two training approaches in given table [III]. The result shows that for Attack1, federated trained model showed good precision than SGD Trained model. However, recall value is good for SGD trained model. Similarly, for other attack type, Federated Trained model has better precision value. Recall value for both models is same for Attack 4 and Attack 8. Yet for other types of attack, recall is slightly less in federated trained model as compared to SGD trained model. Precision and recall curve of the detection model for federated training (FL) vs central training using Stochastic Gradient Descent (SGD)is shown in Figure 3. Precision for different attack type is nearly equal in both federated and central training. However, centralized training outperformed the federated model in recall value.

TABLE III
PRECISION AND RECALL FOR DIFFERENT ATTACKS TRAINED USING TWO METHODS.

Precision	Attack1	Attack2	Attack4	Attack8	Attack16
Federated Trained Model	0.9455	0.6758	0.8675	0.8674	0.9294
SGD Trained Model	0.9454	0.6809	0.8675	0.8615	0.9373
Recall					
Federated Trained Model	0.81	0.63	0.61	0.63	0.73
SGD Trained Model	0.94	0.67	0.61	0.63	0.89

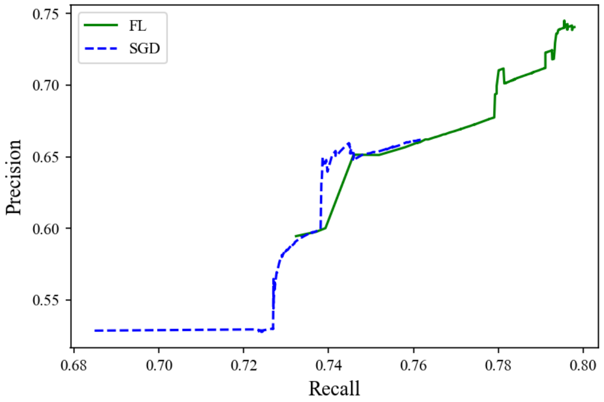

Fig. 3. Precision-Recall Curve

2) Communication Cost: In VANET environment, the central authority and local nodes communicate wirelessly to participate in training process. Local vehicles transmit the updated weights over wireless channel. Performance of this federated training process depends on the availability of network bandwidth and processing capacity of OBUs in vehicles. Communication cost is the cost associated with uploading and downloading the models between vehicular nodes and central authority. We compare the communication cost based on the amount of data exchanged between local vehicles and central authority. In central training, all BSM data are transferred to the CA. In contrast, FL setting requires locally learned weights to be transferred to the CA.

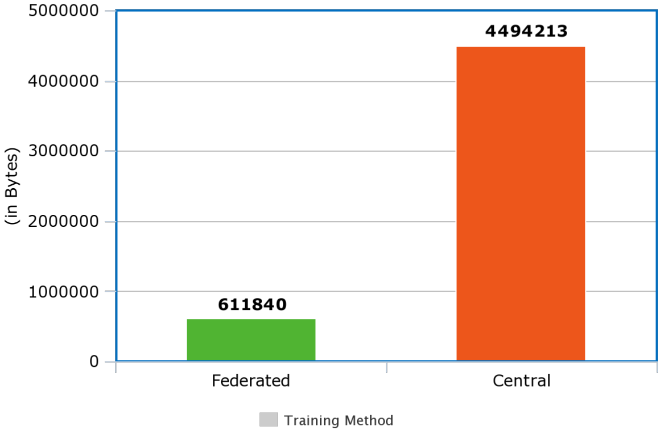

Fig. 4. Size of data communicated

Let d_g, d_l and D be the size of global model, local model and local data (in bits) for each vehicle respectively. Total size of bits to be uploaded and downloaded (T) varies in federated and centralized training approach. Total size in bits is the size of global model to be downloaded by each client plus size of each local model for federated setting. On the other hand, in centralized setting it is the total size of local data generated by vehicle. This size for federated (T_{FL}) and centralized (T_{CL}) setting can be expressed as following:

$$T_{FL} = \sum_{1}^{R} \sum_{j=1}^{N} (d_{g(i)} + d_{l(i)}) \tag{1}$$

$$T_{CT} = \sum_{i=1}^{N} (D_i) \tag{2}$$

where, N is the total number of local vehicle nodes and R is the communication round number. As the VANET network becomes more dense, N will be high. At this time, data generated by each vehicle nodes increases significantly. It is obvious that size of BSM data generated by a vehicle exceeds the weights of the model trained by that data. More the vehicular density more will be the data transfer cost in centralized training approach.

We perform the experimentation to see the data size in bits for training the models in both training setting. For central training, the total data size is equal to the size of all VeReMi dataset that we used for training. In real scenario, for central training, data from each node is collected by central server. Sum of all these data is the data size for central training. In our experimentation, the total data exchanged between server and all nodes during entire communication round (500) is noted. We observed comparatively fewer data communication in federated setting as compared to central approach. In given Figure 4, the data size communicated to achieve same accuracy is comparatively larger in central training than federated approach. Total size communicated using uplink and downlink to converge was observed to be 611840 Bytes. While the total size of data to be collected for central training was 4494213 Bytes. This gap further increases with the increase in vehicular densities. This shows that training the misbehavior detection model in a federated fashion saves communication cost.

VII. Challenges and Future Works

Our experimentation depends on the publicly available labeled dataset. This dataset does not show the properties required for federated learning. We have used the dataset that lacks the true non-IID property and non balanced data distribution. Another challenges is the training error that can occur due to wireless resource limitation [1] when communicating the weights. The local training nodes selected for a round may get disconnected at the middle of the round. This possibility is not considered in this work. We performed the experimentation by using public dataset which is labeled for each attack type mentioned before. However, in real scenario BSM data are not labelled. Labelling the data locally using some pre-processing unit is out of scope of this paper. This part can be looked in future works. The whole goal of this research is to detect attacks without actually looking at BSM data. However, attackers can see the weights sent over the network which violates the privacy preserving property of federated learning. In future, we can use encryption algorithms like homomorphic encryption, differential privacy to make the training process more secure and privacy preserving.

VIII. Conclusion

Data falsification attack in VANET is performed by misbehaving vehicles to disrupt the VANET environment. We use federated machine learning approach to identify the position falsification attack. In this paper, we used publicly available dataset VeReMi to train misbehavior detection model. On training for certain communication rounds, the model converged with acceptable accuracy, precision and recall value. We also compared these metrics of the model when training is done centrally. Federated training showed acceptable performance to centrally trained model. We conclude that without sharing BSM data to third party for model training, we can achieve significant high performing model. Also for federated training, only model weight are to be transferred which highly reduces the data communication overhead as compared to central training. Each vehicle node at last is deployed with the trained global model and can detect position falsification attack on getting a new BSM data from attacker vehicles.

Acknowledgments

This work was supported in part by the US NSF under grants CNS/SaTC 2039583, CNS 1650831, IIS-1924092 and HRD 1828811, and by the DoD Center of Excellence in AI and Machine Learning (CoE-AIML) at Howard University under Contract Number W911NF-20-2-0277 with the U.S. Army Research Laboratory.

References

[1] Mingzhe Chen, Zhaohui Yang, Walid Saad, Changchuan Yin, H Vincent Poor, and Shuguang Cui. A joint learning and communications framework for federated learning over wireless networks. *arXiv preprint arXiv:1909.07972*, 2019.

[2] Z. Du, C. Wu, T. Yoshinaga, K. Yau, Y. Ji, and J. Li. Federated learning for vehicular internet of things: Recent advances and open issues. *IEEE Open Journal of the Computer Society*, 1(01):1–1, may 5555.

[3] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in vanets. In *Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks*, VANET '04, page 29–37, New York, NY, USA, 2004. Association for Computing Machinery.

[4] D. Jiang and L. Delgrossi. Ieee 802.11p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2036–2040, 2008.

[5] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics*, pages 1273–1282, 2017.

[6] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Ag "u era y Arcas. Federated learning of deep networks using model averaging. *CoRR*, abs / 1602.05629, 2016.

[7] D. B. Rawat, B. B. Bista, and G. Yan. Securing vehicular ad-hoc networks from data falsification attacks. In *2016 IEEE Region 10 Conference (TENCON)*, pages 99–102, 2016.

[8] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah. Distributed federated learning for ultra-reliable low-latency vehicular communications. *IEEE Transactions on Communications*, 68(2):1146–1159, 2020.

[9] 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)*, pages 564–571, 2018.