FedMix: A Sybil Attack Detection System Considering Cross-layer Information Fusion and Privacy Protection

1st Jing Zhao
School of Software Technology
Dalian University of Technology
Dalian, China
zhaoj9988@dlut.edu.cn

2th Ruwu Wang
School of Software Technology
Dalian University of Technology
Dalian, China
wrw@mail.dlut.edu.cn

***Abstract*—Sybil attack is one of the most dangerous internal attacks in Vehicular Ad Hoc Network (VANET). It affects the function of the VANET network by maliciously claiming or stealing multiple identity propagation error messages. In order to prevent VANET from Sybil attacks, many solutions have been proposed. However, the existing solutions are specific to the physical or application layer's single-level data and lack research on cross-layer information fusion detection. Moreover, these schemes involve a large number of sensitive data access and transmission, do not consider users' privacy, and can also bring a severe communication burden, which will make these schemes unable to be actually implemented. In this context, this paper introduces FedMix, the first federated Sybil attack detection system that considers cross-layer information fusion and provides privacy protection. The system can integrate VANET physical layer data and application layer data for joint analyses simultaneously. The data resides locally in the vehicle for local training. Then, the central agency only aggregates the generated model and finally distributes it to the vehicles for attack detection. This process does not involve transmitting and accessing any vehicle's original data. Meanwhile, we also designed a new model aggregation algorithm called SFedAvg to solve the problems of unbalanced vehicle data quality and low aggregation efficiency. Experiments show that FedMix can provide an intelligent model with equivalent performance under the premise of privacy protection and significantly reduce communication overhead, compared with the traditional centralized training attack detection model. In addition, the SFedAvg algorithm and cross-layer information fusion bring better aggregation efficiency and detection performance, respectively.**

***Index Terms*—Information fusion, Federated learning, Sybil attack detection, Privacy protection, ANN**

I. Introduction

VANET is a mobile ad hoc network that supports vehicle-to-vehicle communication and vehicle-to-infrastructure communication. It can provide timely and effective safety messages and traffic information for drivers or related agencies. So that they can know the traffic accidents and adverse environment around them in advance and then make timely decisions to ensure their own safety. Many applications are deployed in VANET to provide intelligent services such as road safety, congestion warning, and audio-visual entertainment. Most of these services involve cooperation between multiple vehicles. Suppose the identity of the vehicle involved in the cooperation is false, or it is interfered with by the attacker to send wrong messages. In that case, these applications will make wrong responses and affect the driver's decision-making. Therefore, the correctness and completeness of the data in VANET are essential. Once there is an error in the information in VANET, it will lead to immeasurable harm.

The highly mobile distributed nodes, frequently changing topology environment, and self-management characteristics of VANET make it face more threats and vulnerabilities than traditional wired networks. The existing traditional security mechanisms, such as asymmetric encryption based on public-key infrastructure (PKI) [1], can only eliminate external attacks that do not have key material. However, because the attacker may be an insider of the network (i.e. have valid key materials), this method cannot be detected. The Sybil attack is such an attack launched by insiders. Attackers create multiple virtual identities through forgery or theft and use them to release malicious messages to affect traffic and seek their own benefits. This is highly fatal to VANET. If information related to road safety is interfered, it may lead to serious traffic accidents. Even if the current self-driving vehicles are equipped with sensors such as lidar and cameras, some fake vehicles can be identified. However, the above sensor only works under the condition of the line of sight. In the case of non-line-of-sight (NLOS) and obscured line of sight (OLOS) [2], the above sensors will fail and can only rely on messages or perceived radio signal strength broadcast through DSRC or C-V2X communication technology. Sybil attackers take advantage of this point, using virtual identities to fabricate several actively involved vehicles to send basic safety messages (BSM), falsely reporting their movement, which will directly lead to road safety problems.

Therefore, deploying a Sybil attack detection system in vehicles and infrastructure is essential. At present, the detection of Sybil attack is mainly realized by developing the detection mechanism of abnormal behaviour and performing the analysis of VANET physical-layer data (e.g. received signal strength) or application-layer data (e.g. BSM). These detection schemes are carried out for the single-level information of VANET, and there is a lack of research on the fusion of the two-level information. At the same time, this information contains many private data, such as the vehicle's location and speed. However, the existing detection schemes need to access these private data directly and require vehicles to share them with third-party organizations (e.g. Road Side Unit (RSU), Central Agency (CA)) for centralized analysis, which will lead to privacy leakage. The behaviour of sharing these vehicles' private data is like sharing personal information with potential attackers somehow. Attackers can also use machine learning technology to analyze these data to draw some behavior patterns and conclusions about car owners. Although encryption can alleviate this problem to a certain extent, tens of thousands of vehicles in VANET keep broadcasting BSM messages, which will produce a large amount of data in a short time. Whether these data are transmitted encrypted or not will take up a large amount of network bandwidth, affect the performance of VANET, and will be accompanied by extensive time and economic costs.These limitations and the increasingly perfect laws and regulations on privacy protection make it impossible to implement the existing Sybil attack detection schemes.

Our goal is to design a Sybil attack detection scheme that considers cross-layer information fusion and provides privacy protection. It prevents local vehicle data from being directly accessed by third organizations. Moreover, each vehicle can participate in the training of the attack detection model without transmitting local data. Each vehicle then gets a federated Sybil attack detection model trained cooperatively to perform a joint analysis of the physical and application layer data. Compared with the traditional centralized training Sybil attack detection model, this federated detection model will not reduce the detection performance but also significantly reduce the communication overhead in the training process. Our contributions are summarized as follows.

- A federated Sybil attack detection model with privacy protection: We introduce FedMix: the first Sybil attack detection system that provides privacy protection. FedMix uses the machine-learning algorithm and federation training to generate a federated Sybil attack detection model for attack detection. It provides privacy protection by avoiding mass transmission of private information and direct access by third organizations. We test the federated detection model on several scenarios of the F2MD simulator. The test results show that it can achieve the detection performance equivalent to the centralized training model. At the same time, we calculate the communication overhead of the two models, and the training of the federated machine-learning model has less communication overhead than the traditional machine learning model.
- Cross-layer information fusion detection: We have carried out the research on cross-layer information fusion detection. FedMix will analyze VANET physical-layer data and application-layer data simultaneously. Multi-level data can provide more helpful information for Sybil attack detection. We compare the difference in detection performance between information fusion and non-fusion. The experimental results show that information fusion can achieve better detection performance. In addition, to facilitate the implementation of physical-layer data analysis, we expand the VeReMi [3] data set output by the F2MD simulator, adding the location field and signal strength field when the vehicle receives the message.
- An improved federated aggregation algorithm: We design a new aggregation algorithm called SFedAvg to alleviate the problem of slow convergence speed and low aggregation efficiency caused by the uneven data quality of each vehicle in VANET. Compared with the standard FedAvg aggregation algorithm in federated learning, SFedAvg can improve the aggregation efficiency without reducing the model's performance, so that the number of communication rounds required to achieve the same model performance is less, and the communication overhead is further reduced.

II. Background

A. Federated Learning

Federated learning (FL) has emerged and been promoted to solve the problem that standard deep learning solutions are challenging to implement in privacy scenarios. Federated learning is composed of local devices and global servers. The global server connects many local devices through the network to train the deep neural network model together. Unlike the centralized data collection scheme in standard deep learning, the data in federated training is widely distributed on different local devices. At the same time, the global server only specifies the initial training model and related aggregation algorithm, and does not collect any training data. Moreover, only the model-related parameters are transmitted during their communication. The transmission of raw data and its critical statistical information is prohibited.

At present, the standard aggregation algorithm in federated learning is FedAvg [4]. The design of this algorithm assumes that the data is uniformly distributed in each local device. The parameters of each local model are averaged with a fixed aggregation weight, which is set to be proportional to the size of the client data set, and finally a global aggregation model is obtained. In addition, FedAvg requires that stochastic gradient descent (SGD) must be used as the optimization algorithm. Obviously, such a setting is difficult to achieve optimal in all scenarios. In this work, we focus on Sybil attack detection in VANET. In this scenario, vehicles act as local devices, and the traffic environment in which these vehicles are located is always changing and different. Therefore, it is

almost impossible for the local data set collected by each vehicle to be uniformly distributed.

B. Sybil Attack

The Sybil attack was first introduced by Douceur in [5]. The vehicles communicate via DSRC technology (also known as ITS-G5 technology) or V2X Mode 4. Basic Safety Message (BSM) is broadcast periodically by all vehicles. Each message contains the vehicle's pseudonym (temporary identity) and several kinematic information (e.g. position, velocity, orientation, etc.). A public key infrastructure (PKI) can be used to manage cryptographic certificates, providing a vehicle with one long-term certificate and several short-term certificates, called pseudonym certificates. These certificates are used to sign BSM messages. Vehicles change pseudonyms frequently to avoid tracking and protect privacy. In order to ensure the vehicle's ability to send BSM messages continuously, several valid pseudonyms must be provided at the same time, and the European Commission recommends the use of up to 100 valid pseudonym certificates [6]. However, PKI is an effective countermeasure against external attackers (i.e. attackers without valid encryption certificates). An internal entity with a valid encryption certificate can still launch an attack. We cannot assume that all internal entities are trustworthy. And researchers have shown that internal security threats are real in VANET, which has been proven through field operational testing (FOT) [7]. Such attacks initiated by internal entities are often referred to as insider attacks. Sybil attacks are one of them. A Sybil attack occurs when a vehicle with a valid cryptographic certificate intentionally uses multiple valid pseudonyms simultaneously (regular vehicles cannot use more than one pseudonym certificate to sign BSM messages within a certain period of time).

F2MD [2] is the latest framework for full-element research on simulation and security analysis of insider attacks in VANET. The framework is dedicated to the analysis of application-layer BSM messages. It is designed to be easily extended to develop new attack and anomaly detection algorithms. There are four forms of Sybil attacks implemented in this framework, which are described in detail in [6]. Here, we briefly recapitulate:

1) Traffic Congestion Sybil: Attackers use multiple valid pseudonyms to simulate multiple ghost vehicles intelligently, make them reasonably distributed on the road, and publish false messages with credible content, creating the illusion of traffic congestion.
2) Data Replay Sybil: The attacker selects a victim's vehicle, and whenever it receives a message from it, it immediately creates a message with the same essential content for replay. Meanwhile, it changes the pseudonym for each replay. Making the victim's vehicle more likely to be mistaken for the attacker by the detection system.
3) Dos Random Sybil: The attacker increases the beacon frequency of the vehicle, uses different pseudonyms, and sends a large number of messages filled with random data (e.g. random locations). As a result, the detection system is in a busy state for a period of time and can not process the latest information in time.
4) Dos Disruptive Sybil: This attack is a combination of 2) and 3). The attacker still uses a different pseudonym in each message, but instead of randomly filling in the message content, it is based on real messages received from nearby vehicles. At the same time, different from 2), it is not only for a single vehicle but for multiple nearby vehicles, making the messages exchanged by all surrounding vehicles become unreliable in a short time.

III. Related Work

Sybil attack detection has been explored by extensive research. They can be roughly divided into detection research on physical-layer data and detection research on application-layer data. Specifically, in the research on detecting application-layer data, Hao et al. [8] proposed a security protocol to detect Sybil nodes cooperatively by checking the rationality between the vehicle's location and the location of its neighbors. When a vehicle detects a potential Sybil node in a neighbor, it broadcasts to other neighbors to confirm whether an attack is occurring. When it is confirmed that the number of vehicles attacking at this time is greater than the threshold, the vehicle will refuse to receive messages sent by the identified attacking vehicle for a period of time. However, it relies on the strong assumption of the honest majority principle. When the number of attackers in the neighborhood of the vehicle is larger, the cooperative detection will fail. However, the detection method based on machine learning technology does not rely on the strong assumption of the honest majority principle and can automatically extract some potential detection rules. It is therefore heavily studied in the literature. Gu et al. [9], [10] proposed the detection method of Sybil attack based on the k-nearest neighbor and support vector machine (SVM) algorithm, respectively. Subsequently, Quevedo C et al. [11] proposed using the extreme learning machine (ELM) for Sybil attack detection to achieve lower computational complexity. They all classify vehicles based on the similarity of vehicle driving patterns and then identify Sybil nodes. In addition, Eziama et al. [12] used a Bayesian network combined with probabilistic modeling to establish a trust model to identify honest and malicious nodes. In the above methods based on machine learning, to obtain a model with higher detection performance, a large size of data is often needed for model training. However, in VANET, due to the consideration of communication cost and the limitation of privacy protection, it is not feasible to collect a large size of data centrally.

In the research on detecting physical-layer data, Lv et al. [13] proposed a Sybil attack detection scheme based on RSSI of received signal strength and node cooperation. This method does not calculate the exact position but calculates the distance between different nodes. The identities with similar signal strength are combined and then broadcast to other nodes. Each node comprehensively determines the Sybil node according to the multiple identity sequences received. This method uses mutual trust between nodes as the premise of cooperation,

while Sybil nodes can not be trusted. In addition, Zhang et al. [14] proposed an intrusion detection mechanism based on LSTM. The method uses the LSTM algorithm to self-learn the difference between actual and estimated RSSI sequences and establishes a trust list to identify malicious nodes. Yao et al. [15] also proposed a Sybil attack detection method based on RSSI sequence. The method takes the RSSI sequence as the vehicle identifier and compares the similarity between all the received sequences. It does not rely on the radio propagation model and neighbor cooperation, but an attacker can evade detection by controlling the signal transmitter. Therefore, the detection method that only depends on the single-level data of the physical layer is unreliable.

IV. System Design

The Sybil attack detection system FedMix designed by us consists of two parts: the FedMix client installed in the vehicle and the FedMix server installed in the central agency. The specific design is shown in Fig. 1. The system uses the proposed cross-layer information fusion detection method for attack detection. Based on the ANN classification algorithm, this method distinguishes malicious nodes from benign nodes by driving patterns (from application-layer data) between vehicles and the difference between actual and estimated RSSI sequences (from physical-layer data) during the driving process. At the same time, the system adopts the way of federated training. FedMix client will perform data preprocessing and feature extraction steps to generate the difference between RSSI sequences and the characteristics of driving patterns and carry out local model training. Then, the FedMix server will constantly aggregate local models and distributes the latest federated aggregation model. When new messages are received, the client will generate the features of the message sequence and then use the federation detection model to identify them to determine whether they are sent by malicious nodes. The system is described in detail below.

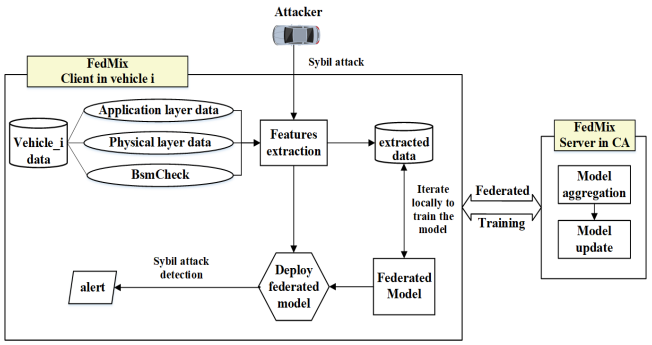


Fig. 1. FedMix Sybil attack detection system.

A. Data Preprocessing

The FedMix system will preprocess the application and physical layer data, respectively. First, the basic plausibility and consistency checks (provided by F2MD) are performed for the application-layer data. Moreover, for physical-layer data, we calculate the distance $Dist$ between the receiving and sending vehicles according to their positions and estimate the received signal strength corresponding to the current distance by using the wireless signal attenuation model in [14]. And then calculate the difference $rDiff$ between it and the actual physically measured receive signal strength of the vehicle. The higher the $rDiff$ value, the more likely it is that the message is sent by a malicious node. Of course, this may also be due to accidental errors because the empirical infinite fading model is not always suitable for signal estimation in all cases, especially when the vehicle is traveling at a high speed or the distance between vehicles is long. Therefore, it is necessary to combine more dimensions of information for further judgment.

B. Feature Extraction

Each message is cached after preprocessing and accumulated for some time. When the set time window is reached, the time series data in the window will be extracted. The extracted features represent the characteristics of different dimensions of the data stream during this period. The types of features extracted from the application-layer data and the physical-layer data are shown in Table I below.

TABLE I: Feature Types

Feature Type	Detail
agg_autocorrelation	This feature computes descriptive statistics of time series autocorrelation [16]
benford_correlation	This feature calculates the correlation between the first-digit distribution of time series and the distribution of Newcomb-Benford's Law [16].
variation_coefficient	This feature calculates the coefficient of variation of the time series, that is, the relative value of the variation around the mean. [16]
maximum	This feature calculates the maximum value in the time series
minimum	This feature calculates the minimum value in the time series

In addition, we customize a feature extraction rule called checkScore for the results of plausibility and consistency checking. The specific design is as follows. We think that for a time series, the current time's basic plausibility and consistency check results are more important than those of the historical time. Therefore, to get the comprehensive score of each basic plausibility and consistency check at each moment, we balance the historical average score and the current score by setting weights. The specific calculation formula is as follows. Assuming that the length of the time series data is n, m represents the number of basic plausibility and consistency checks, and C_{ij} represents the j-th check of the i-th data, then the checkScore is calculated as:

$$\text{checkScore} = \frac{\sum_{j=1}^{m} \text{check}_j}{m} \tag{1}$$

where the comprehensive score for the j-th basic plausibility and consistency check can be calculated according to the following formula:

$$\text{check}_j = \sum_{i=2}^{n} \left(C_{ij} \times 0.6 + \left(\sum_{k=1}^{i-1} C_{kj} \right) / (i-1) \times 0.4 \right) \tag{2}$$

C. Federated Training

Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Random Forest, and LSTM machine learning models have been tried to use in Sybil attack detection. However, most of them are not suitable for federated training, such as Support Vector Machine. It needs to traverse all the data to get the support vector, which contradicts the fact that federated training does not collect raw data. In the FedMix system proposed in this paper, we choose the artificial neural network (ANN) as the basic model, which can be used to perform federation training. This model will classify the received information flow according to the extracted time series message features.

We describe the process of the FedMix system performing federated training in VANET as follows. First, the central agency initializes a blank global model $model_{(0)}$ using the FedMix server. Then perform multiple rounds of aggregation to obtain more intelligent models continuously. Fig. 2 shows the specific process of each round of aggregation, taking the i-th round of aggregation as an example. This process includes the following steps:

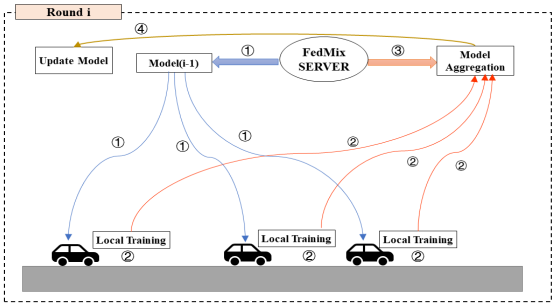


Fig. 2. Round i of federated training in VANET.

1) The FedMix server in the central agency sends the model $model_{(i-1)}$ to each vehicle entity (client) through the network.
2) Each vehicle entity trains the model by using its own local data, updates the weight of the model, and sends the updated model to the FedMix server.
3) The FedMix server executes the aggregation algorithm to aggregate all local models into a new model $model_{(i)}$.
4) The FedMix server updates the stored global model.

D. Model Aggregation Algorithm

The model aggregation algorithm belongs to the federated optimization problem. In the standard FedAvg algorithm, at each round of model aggregation update, the aggregation weight of the current participant in the aggregation is given by calculating the proportion of the size of the participant's training dataset to the sum of the sizes of all participant's training datasets. Then, the models of each participant are aggregated according to the weights to obtain a single global model w. Moreover, the local client adopts a stochastic gradient descent (SGD) algorithm to update the obtained global model w in multiple rounds. The updated model parameters and training dataset size are then uploaded. The update of the model parameters comes from the SGD gradient $\nabla F_k(w)$ generated after several rounds of training with the client's private data. Let w_i represent the model parameters of the client after the i-th round of local training, and let η represent the learning rate of the client. Then, the parameter update of the client model in i-th round can be expressed as $w_i = w_{i-1} - \eta \nabla F_k(w)$.

Because the SGD gradient of the client is generated after several rounds of training, this is originally intended to speed up the federated learning and reduce the number of communications, but it also brings problems. The different distribution and quantity differences of client data (e.g. label imbalance, non-iid data) will cause huge differences in SGD gradients between clients. In this case, the FedAvg algorithm still simply averages the aggregation weights according to the size of the training set. So the impact of label imbalance in the training data of each client will be completely ignored. This will reduce the convergence rate of the global model and increase the communication rounds for federated training.

Therefore, we propose SFedAvg algorithm. It is mainly divided into three steps: initial aggregation, screening and final aggregation. In the initial aggregation step, we aggregate the models of each participant (client) more reasonably to get a preliminary overall model. We require participants to upload the attack label proportion of the training set in addition to model parameters and training set size. And this does not defeat the original purpose of privacy protection. Because privacy issues in VANET mainly focus on protecting vehicle location and other travel information, the proportion of labels will not reveal the owner's private information. We co-determine the aggregate weight for each model participating in the aggregation based on the training set size and the label proportion. Their contribution to the weight allocation is 0.7 and 0.3, respectively. This proportion is adjustable, but the training set size proportion is not recommended to be less than 0.5 because it is still the first factor to be considered in the aggregation weight allocation. We let the size of the training set of K clients be represented by the set $S = \left[s^1, \ldots, s^K\right]$, the attack label proportion is represented by the set $R = \left[r^1, \ldots, r^K\right]$. Then the aggregation weight α^k of the k-th client's model in the aggregation is:

$$\alpha^k = 0.7 \times \frac{s^k}{\text{sumS}} + 0.3 \times \frac{r^k}{\text{sum}\, R} \tag{3}$$

where $\text{sum}\, S = \sum_{k=1}^{K} s^k$ and $\text{sum}\, R = \sum_{k=1}^{K} r^k$.

The proportion of attack label in training data of Sybil attack detection model is usually low. This is because the attack sample itself is not easy to collect in a real scenario. The weight allocation strategy we proposed can improve the influence of attack label.

In the screening step, we first calculate the update gradient of the preliminary global model and each participant model obtained in the previous step. Then the cosine similarity between the update gradient of each participant model and the update gradient of the preliminary global model is calculated,

and the obtained cosine similarity is sorted from large to small. Participants behind the third quartile are eliminated because they are considered to be too different from the overall update gradient direction, which will reduce the effectiveness of the federated model.

In the final aggregation step, the participants not eliminated by the screening step are selected to perform the first aggregation step again. The eliminated participants will not be able to participate in the aggregation. That is, they will receive an aggregation weight of 0. At the end of this step, the final model of the current round of aggregation is obtained.

It is assumed that there are T rounds of federal training, the set of clients participating in the training is $C = \left[c^1, \ldots, c^K\right]$, w_t represents the aggregation model of round t, E represents the number of local training epochs, and η represents the learning rate. The model uploaded by each client in round t aggregation is represented as set $W_t = \left[w_t^1, \ldots, w_t^K\right]$. The corresponding aggregation weight for each client model is $\alpha = \left[\alpha^1, \ldots, \alpha^K\right]$. rn represents the number of clients eliminated in each round of aggregation, and its value is $K - \mathrm{int}(K \times 0.75)$. The corresponding set of clients eliminated in round t is represented as $rm_t = \left[rm_t^1, \ldots, rm_t^{rn}\right]$. Then the detailed description of the SFedAvg algorithm is shown in Algorithm 1.

Algorithm 1 The SFedAvg Algorithmic Framework

Initialize w_0
for $t = 1, ..., T$ **do**
  for each client k $\in C$ in parallel **do**
    $w_t^k = w_{t-1}$
    for $e = 1, ..., E$ **do**
      Sample batch p from client k 's training data.
      Compute loss $l\left(w_t^k; \mathrm{p}\right)$.
      Compute gradient of w_t^k and update w_t^k.
    end for
    Send w_t^k, s^k, r^k to server.
  end for
  At FedMix Server:
    Receive $w_t^k, s^k, r^k (k \in C)$.
    For initial aggregation step: calculate the aggregation weight α^k for each client according to formula (3) $(k \in C)$.
    Compute the global model based on all local models $w_t = \sum_{\mathrm{k} \in C} \left(\alpha^k \times w_t^k\right)$
    For screening step: calculate the gradient of global model and each local model update $diff_{\text{global}} = w_t - w_{t-1}$, $diff_{\text{local}}^k = w_t^k - w_{t-1} (k \in C)$.
    Compute the cosine similarity for each $diff_{\text{local}}^k$ and $diff_{\text{global}}$, and sort them in descending order.
    $rm_t \leftarrow$ Obtain the rn clients with the lowest cosine similarity.
    For final aggregation step: perform the initial aggregation step for clients that do not belong to the rm_t set to obtain the final model w_t.
    Broadcast w_t to each client.
end for

V. Experimentation

A. Data Set

To evaluate our proposed solution FedMix, We use the publicly available F2MD [2] simulation framework to provide simulations of realistic scenarios and generate datasets. Datasets are important for supervised machine learning techniques. Datasets in VANET can be obtained through accurate scene testing and simulation. Currently, the research in the literature is usually based on the dataset generated by the simulator to verify the detection scheme because it is challenging to capture sufficient Sybil attack detection cases in the real world. Moreover, there are few publicly available datasets, and none of them are suitable for detecting Sybil attacks. For example, the VeReMi [3] dataset is the first public misbehavior detection dataset in VANET. Nevertheless, it only contains five location tampering attacks, unsuitable for advanced attacks such as Sybil attacks or DoS attacks. However, the recently released F2MD simulation framework is a promising one that provides simulations of advanced attacks, which the public can access for research purposes. F2MD is an extension of VEINS [17]. VEINS is an open-source framework for vehicular network simulation, consisting of an event-based network simulator (OMNeT++) [18] and a road traffic simulator (SUMO) [19].

F2MD can generate two kinds of datasets. One is the VeReMi dataset. It records the BSM messages received by each vehicle from neighboring vehicles during the entire journey and whether the vehicle itself is an attacker. In addition, for physical layer data analysis, we extend the VeReMi dataset output by F2MD by adding two fields: the location and the received signal strength when the vehicle receives the message. Another dataset generated by F2MD is the BSMList, which contains the results of the basic plausibility and consistency checks (provided by F2MD) on each message received.

We preprocess the above two datasets to make them more friendly to attack detection methods based on machine learning. First, we will calculate the $Dist$ and $rDiff$ values (see Section IV) and add them to each message record in the VeReMi dataset. Then we merge the VeReMi dataset and the BSMList dataset. Specifically, we will extract the basic plausibility and consistency check of each message in the BSMList dataset, and make them correspond to the messages in the extended VeReMi dataset one by one. Finally, the merged data is sorted according to the ID of the vehicle receiving the message, the pseudonym ID of the vehicle sending the message, and the receiving time. Table II shows a brief summary of each piece of data in the merged dataset.

The merged dataset contains the local dataset for all vehicles. For experimental purposes, we screened all vehicles' datasets, filtering out vehicles that received very little data. Then 500ms is used as the extraction window (five consecutive data from the same vehicle) to extract the features of the time series data received by each vehicle. The dataset after feature extraction will be used for model training and testing, with a division ratio of 4:1. We used the tsfresh [16] open source

TABLE II: Merged dataset

Field	Detail
rcv_veh_id	ID of the vehicle receiving the message
sd_veh_id	Pseudonym ID of the vehicle sending the message
rcv_time	Time when message was received
application-layer data	Includes BSM data (position, speed, acceleration, etc.) received by each vehicle from surrounding vehicles
physical-layer data	Including the received signal strength, the distance between vehicles sending and receiving messages, and the difference between the estimated signal strength and the actual physical measurement of signal strength rDiff
bsmCheck	Results of basic plausibility and consistency checks performed on the BSM (provided by F2MD)

library to extract interpretable features related to time series.

B. Experimental Setup

The simulation runs under the Ubuntu/Linux operating system. We validated our scheme in the Luxembourg SUMO Traffic (LUST) [20] and Ulm SUMO Traffic (ULM) scenarios. Both scenarios are publicly available from F2MD. In order to carry out Sybil attack detection, we inject four kinds of Sybil attacks provided by F2MD into the above two simulation scenarios. The density of attacker vehicles introduced in LUST and Ulm scenarios is 0.2 and 0.1, respectively. Each attacker vehicle randomly selects a Sybil attack type. This setting can verify our proposed solution more comprehensively.

In addition, we use the Keras framework and TensorFlow framework in the simulation to build and train the model and simulate the federated and centralized training process. We conducted multiple experiments on the two training methods and comparative analysis. In federated training, selecting the client to participate in the training is usually necessary before it starts. In order to improve efficiency, the number of clients should not be too large in general. In our simulation experiment, we selected ten vehicle entities as clients. In order to make the data more sufficient, we collect the data of each vehicle after feature extraction, randomly shuffle them, and then redistribute them to the ten vehicle entities. In addition, considering the real scene, data distribution among vehicle entities should be Non-Independent Identically Distribution(Non-IID). Therefore, we assign different data sizes to vehicle entities and ensure that there is at least one vehicle entity with only data for a single category label, thus assigning them non-iid attributes. Note that the above operations are only for experimental purposes. The data in the real world are all local to the vehicle, so there is no need to aggregate and distribute the data. When the federated simulation starts, these selected virtual local vehicle entities can only access their own allocated data for model training, just as if the real vehicles were trained independently locally. The parameter setting of federation training is shown in Table III.

For the traditional centralized training, we use the same parameters as the federated training to train the neural network model. The neural network model at this time will be trained for 3000 epochs. In order to facilitate the comparison with the federated training, the results of every 5 epochs in the centralized training are recorded once to correspond to the results of 1 round of aggregation in the federated training. Both training methods use the same training set and test set. At the same time, the architecture of the ANN model used by them is the same. The model consists of an input layer, two tightly connected hidden layers, a dropout layer, and an output layer. We set the model's hyperparameters as follows: the number of neurons in each hidden layer is 300, the hidden layer activation function is ReLU, the output layer activation function is sigmoid, and the optimization algorithm is SGD, and the loss function is Binary cross-entropy.

TABLE III: Federation training parameters

Param	Detail	Value
R	Round number	600
C	Client number	10
s	Number of clients selected for a round	10
B	Batch size used at local clients	16
E	Epoch number at local clients	5
LR	Learning rate of local client	0.01

VI. Evalution and Discussion

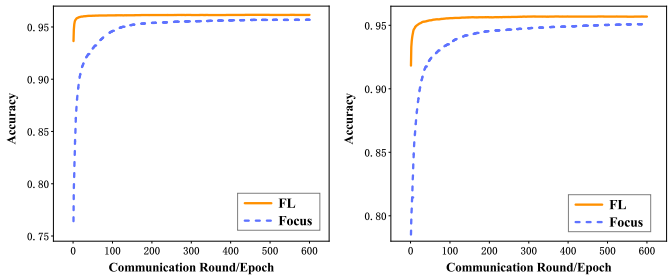
We evaluate the performance of these models using accuracy, precision, recall and f1-score. The accuracy refers to the ratio of correct predictions (positive and negative) over all data points. The precision refers to the proportion of the correct prediction of the positive class to the total prediction of the positive class. Moreover, recall refers to the proportion of correctly predicted positive classes to all actual positive classes. The f1-score is the harmonic mean of the computed precision and recall values. Precision and recall affect each other and hold each other down. To balance the precision and recall estimation of the model, consider using the f1-score. It can define the overall performance of the model in a single value.

A. Results of Performance Comparison Between Federated and Centralized Model

Our first group of experiments is to test the overall performance of federated and centralized training models and conduct a comparative analysis. We look at the model's overall performance from the following two perspectives.

a) Evaluation metrics: We first compare the overall accuracy of the models trained using the two methods for attack detection. The Lust scenario and the Ulm scenario results are shown in Fig. 3(a) and Fig. 3(b), respectively. It can be seen from the figure that the overall accuracy achieved by federated training is not lower than that of centralized training.

We also analyzed the overall precision, recall, and f1-score achieved by the two training methods, as shown in Table V. The experimental results show that the precision and recall of the federated model are higher than those of the centralized model in the Lust scenario. However, the centrally trained model in the Ulm scenario performs better in precision. As can



(a) Lust scenario (b) Ulm scenario

Fig. 3. FL vs Focus Accuracy in two scenario.

be seen from the f1-scores of the two scenarios, the federated training performed better overall than the centralized training.

TABLE IV: Precision, recall and f1-score of the model trained using two methods

Scenario	Model	Precision	Recall	F1-score
Lust	federated	**0.985**	**0.851**	**0.913**
	centralized	0.983	0.832	0.902
Ulm	federated	0.952	**0.842**	**0.894**
	centralized	**0.961**	0.807	0.878

b) Communication cost: In the VANET environment, the central agency and the local vehicle entity participate in the training process by communicating wirelessly via DSRC or V2X technology. The communication cost of federated training is the cost associated with the local vehicle uploading or downloading the model as it interacts with the central agency throughout the training process. The communication cost of centralized training is the cost associated with the vehicle uploading all local data to the central agency, including the vehicle's BSM data and physical layer data. We compare the communication costs in terms of the size of data exchanged between the local vehicle entity and the central agency.

In our experiments, the upload cost for a single round of federated training is set to the sum of the sizes of all local models and the sizes of the dataset attributes that need to be uploaded. Its download cost is set to the sum of the global model sizes obtained by all local vehicle entities. We recorded the total size of data exchanged between all vehicle entities and the central authority over the entire communication cycle (600 rounds). On the other hand, the communication cost of centralized training is set to the total size of the VeReMi dataset and the BSMList dataset. This is because centralized training requires the central agency to collect local data of all vehicles, and these two data sets store all local data of vehicles. We take the Lust scenario as an example to make a statistical comparison of their communication costs, as shown in Fig. 4 below.

We observe that the total size of data communicated in the federated training is relatively small compared to the centralized training approach. Fig. 3 and Fig. 4 show that the size of data communicated to achieve a similar accuracy rate

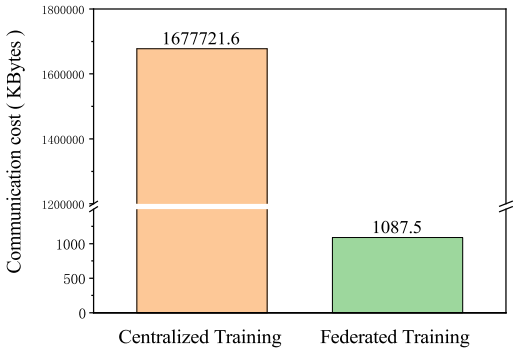


Fig. 4. Size of data exchanged in Lust scenario.

in the centralized training is relatively larger than in the federated training method. The communication cost of federated training is 1087.5K bytes, while the communication cost of centralized training is 1677721.6K bytes. It is worth noting that when the density of vehicles in VANET increases, the data collected locally by vehicles will increase significantly. The communication costs of centralized training will also increase dramatically, while federated training will not. This shows that the communication cost can be saved by using the federated training method to train the Sybil attack detection model. Moreover, the greater the density of vehicles, the more obvious this advantage will be.

B. Results of Comparison Between Cross-layer Information Fusion Detection and Single-layer Information Detection

Our proposed FedMix uses cross-layer information fusion detection, which collects both application-layer and physical-layer data. In the second group of experiments, we compare the overall detection performance of this fusion detection with that of collecting only a single layer of data at the physical or application layer for detection. In order to conduct a comparative experiment, for the method of collecting only the physical-layer data for detection, only the data of the physical-layer part will be used. Meanwhile, the method that only collects application-layer data for detection will only use data from the application-layer part and the bsmCheck part for the application layer. Table I shows the specific contents of each part of the data. We tested the accuracy and f1-score of the model obtained by cross-layer information fusion and only using single-layer information under centralized training and federated training (using the SFedAvg aggregation algorithm). In order to make the experiment more convincing, we deployed this group of experiments in both Lust and Ulm scenarios, and the experimental results are shown in Table V below.

The experimental results show that the model trained by the data after cross-layer information fusion in the two scenarios, whether centralized or federated training, achieves the best detection accuracy and f1-score. The model trained only with physical-layer data achieves the lowest detection accuracy and f1-score.

TABLE V: Comparison results of cross-layer information fusion detection and single-layer information detection

Scenario	Matrix	Centralized Training			Federated Training		
		Multi	Phy	App	Multi	Phy	App
Lust	accuracy	**0.957**	0.926	0.946	**0.962**	0.939	0.951
	f1-score	**0.902**	0.822	0.874	**0.913**	0.858	0.886
Ulm	accuracy	**0.951**	0.907	0.942	**0.956**	0.923	0.951
	f1-score	**0.878**	0.747	0.852	**0.894**	0.797	0.879

C. Comparison Results of Aggregation Efficiency Between SFedAvg and FedAvg

Our third group of experiments is to test the aggregation efficiency of the SFedAvg aggregation algorithm and the baseline algorithm FedAvg. We conducted experiments in Lust and Ulm scenarios, respectively, and compared the number of communication rounds required by the models to achieve similar accuracy and f1-score using two different aggregation algorithms. In the Lust scenario, we observe the number of communication rounds required when the detection accuracy of the model reaches or exceeds 95% and 96% for the first time and when the f1-score reaches or exceeds 90% and 91% for the first time. In the Ulm scenario, we choose to observe the number of communication rounds required when the detection accuracy of the model reaches or exceeds 94% and 95% for the first time and when the f1-score reaches or exceeds 88% and 89% for the first time. The experimental results are shown in Table VI.

TABLE VI: Communication rounds required to achieve the same detection accuracy and f1-score using two algorithms

Lust Scenario		acc=95%	acc=96%	f1=90%	f1=91%
	FedAvg	3	25	7	39
	SFedAvg	**2**	**16**	**2**	**24**
Ulm Scenario		acc=94%	acc=95%	f1=88%	f1=89%
	FedAvg	4	17	36	129
	SFedAvg	**1**	**3**	**5**	**93**

The experimental results show that the SFedAvg algorithm requires fewer communication rounds and has higher aggregation efficiency when the model achieves the same detection accuracy and f1-score in the two scenarios.

VII. Conclusion

In this paper, we propose a Sybil attack detection system, called FedMix, which considers cross-layer information fusion and privacy protection. The system can train the federated Sybil attack detection model without requiring vehicles to upload private data such as BSM, and then performs joint analyses on VANET physical-layer and application-layer data. We conducted simulation tests in several scenarios provided by the F2MD simulation framework to verify the effectiveness of FedMix. The results show that FedMix can provide a significant high-performance model with low communication overhead and provide privacy protection.

References

[1] S. Tangade, S. S. Manvi, and P. Lorenz, "Decentralized and scalable privacy-preserving authentication scheme in vanets," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2018.

[2] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6631–6643, 2020.

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

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, pp. 1273–1282.

[5] L. M. Surhone, M. T. Tennoe, and S. F. Henssonow, *Sybil Attack*. Sybil Attack, 2010.

[6] J. Kamel, F. Haidar, I. B. Jemaa, A. Kaiser, B. Lonc, and P. Urien, "A misbehavior authority system for sybil attack detection in c-its," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2019, pp. 1117–1123.

[7] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con*, vol. 21, no. 260-264, pp. 15–31, 2013.

[8] Y. Hao, J. Tang, and Y. Cheng, "Cooperative sybil attack detection for position based applications in privacy preserved vanets," in *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*, 2011, pp. 1–5.

[9] P. Gu, R. Khatoun, Y. Begriche, and A. Serhrouchni, "k-nearest neighbours classification based sybil attack detection in vehicular networks," in *2017 Third International Conference on Mobile and Secure Services (MobiSecServ)*, 2017, pp. 1–6.

[10] P. Gu, R. Khatoun, Y. Begriche, and A. Serhrouchni, "Support vector machine (svm) based sybil attack detection in vehicular networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, 2017.

[11] C. H. O. O. Quevedo, A. M. B. C. Quevedo, G. A. Campos, R. L. Gomes, J. Celestino, and A. Serhrouchni, "An intelligent mechanism for sybil attacks detection in vanets," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[12] E. Eziama, K. Tepe, A. Balador, K. S. Nwizege, and L. M. S. Jaimes, "Malicious node detection in vehicular ad-hoc network using machine learning and deep learning," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–6.

[13] S. Lv, X. Wang, X. Zhao, and X. Zhou, "Detecting the sybil attack cooperatively in wireless sensor networks," in *2008 International Conference on Computational Intelligence and Security*, vol. 1, 2008, pp. 442–446.

[14] Y. Y. Zhang, J. Shang, X. Chen, and K. Liang, "A self-learning detection method of sybil attack based on lstm for electric vehicles," *Energies*, vol. 13, no. 6, p. 1382, 2020.

[15] Y. Yao, B. Xiao, G. Wu, X. Liu, Z. Yu, K. Zhang, and X. Zhou, "Multi-channel based sybil attack detection in vehicular ad hoc networks using rssi," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 362–375, 2019.

[16] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh – a python package)," *Neurocomputing*, vol. 307, pp. 72–77, 2018.

[17] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, and M. Segata, *Veins: The Open Source Vehicular Network Simulation Framework*. Recent Advances in Network Simulation, 2019.

[18] A. Varga, "Using the omnet++ discrete event simulation system in education," *IEEE Transactions on Education*, vol. 42, no. 4, pp. 11–pp, 1999.

[19] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using sumo," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.

[20] L. Codeca, R. Frank, and T. Engel, "Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research," in *2015 IEEE Vehicular Networking Conference (VNC)*, 2015, pp. 1–8.