FEDRUL IN VANET

Parameter	Value
Size of map	2.5 km x 2.0 km
Number of nodes	50 (including 36 RSUs)
Application layer protocol	WAVE Short Message Protocol (WSMP) 0x88dc
Physical layer protocol	IEEE 802.11p (physical and MAC layers)
Channel selection	CCH ch176 as control channel
Channel bandwidth	10 MHz
Data transmission rate	6 Mb/s
OFDM	48 subcarriers

The purpose of the experiment

There are two main approaches to RUL (Remaining Useful Life) prediction: model-based prediction and data-driven prediction. Model-based prediction relies on mathematical or physical models of machine degradation and requires a deep understanding of the physical failures of the system. On the other hand, data-driven prediction methods rely on statistical and machine learning models trained on historical data.

Typical data-driven prediction methods for RUL are based on deep learning techniques. However, in many practical cases, transferring data to a central server for training may expose data privacy issues. By training models directly on vehicle nodes, the risk of data leakage can be avoided, and data privacy can be ensured. Training models directly on the vehicle nodes enables real-time data processing and decision-making because data does not need to be transmitted to a central server for processing and can be processed and trained directly on the node. This enables faster decision-making and response when processing real-time data.

Moreover, training models directly on vehicle nodes can reduce data transfer and communication overhead. Only model updates need to be transferred from the nodes to the server, rather than the entire dataset. This reduces network bandwidth and communication overhead, making the overall system more efficient.

Problem Description

In this task, the goal is to use sumo and ns3 to generate a simulation environment consisting of mobile vehicles and a Roadside Unit (RSU). The RSU is located at position 14-49, while nodes 0-13 are mobile vehicles generated using sumo. The total simulation time is 81 seconds. The WAVE protocol is used for communication between the nodes, which is mainly composed of 802.11p and IEEE1609.

The task involves sending a data packet containing a trained neural network model from a vehicle node to the RSU every 0.4 seconds. Each data packet carries a tag, which contains the current speed and position information of the sending node. Only the RSU within communication range can receive the data packet and send it to the server for model aggregation. The RSU then sends the data packet to all mobile vehicles within communication range in the next time interval. Similarly, only vehicles within communication range can receive the data packets and update their local model. The vehicle nodes perform model training again to enter the next round of federation aggregation stage.

In the 802.11 protocol, three channel bandwidths are defined: 20MHz, 10MHz, and 5MHz. 802.11p uses a bandwidth of 10MHz, and the data transmission rate can reach 27Mb/s. The communication range can reach 300-1000m. The 802.11p protocol provides four types of devices with different transmission powers, namely: Class A devices, suitable for point-to-point communication close to the vehicle. Class B equipment, suitable for point-to-point communication over long distances, such as communication between vehicles on a highway. Class C equipment, suitable for short-distance communication in dense areas such as cities. Class D equipment, suitable for communication in short-distance communication scenarios such as parking lots.

Type	Maximum Output Power	Maximum Communication Range
A	23 dBm	About 300 meters
B	20 dBm	About 1 kilometer
C	14 dBm	About 100 meters
D	8 dBm	About 10 meters

Since the positions and speeds of different vehicles are constantly changing in the Internet of Vehicles scenario, this task aims to simulate the communication heterogeneity problem in federated learning. In addition, the task also simulates the packet loss problem in the real vehicle network, which further affects the training effect of the model. The transmit power of all nodes is set to the same value to further simplify the simulation.

Notes:

- The default RSU and the base station are wired connections, ignore this part
- Due to computer CPU and memory limitations, there is not a one-to-one correspondence between simulation time and model training time.

Main difficulty

- The data is non-iid, which affects the performance of the final model training.
- Since the car node sends data packets to 36 RULs at the same time at each moment, channel conflicts cause packet loss, which affects the final training performance of the model.
- In the downlink phase, due to packet loss, the models of some nodes are not updated.

Solution

A. On the client side, choose whether to send data packets through the current model parameters and speed and position.

- State:

$$\{w_1, w_2, \ldots. w_N\}, wn = \{NeuralNetworkParameters, position, speed\}$$

- Action: N-dimensional vector, if the packet is sent, the corresponding position is 1, otherwise it is 0
- Reward: Encourages the accuracy of the final model and penalizes packet loss

B. On the server side, directly use RL to select the client that can send out the packet.

note:

- Scheme A can effectively reduce the amount of data transmission, alleviate the packet loss problem caused by channel conflicts, and improve the performance of the model,
- Option B simply does the combination

Experimental design

Scheme A

- Main challenge:
    1. It is necessary to use ns3-ai to control whether the vehicle node sends data packets in real time to observe the packet reception of RSU
    2. Build a federated learning framework for Internet of Vehicles in ns3
- Main process:

  (1)ns3 outputs the speed and position information of the node

  (2)Control which nodes send packets through reinforcement learning in ns3-ai

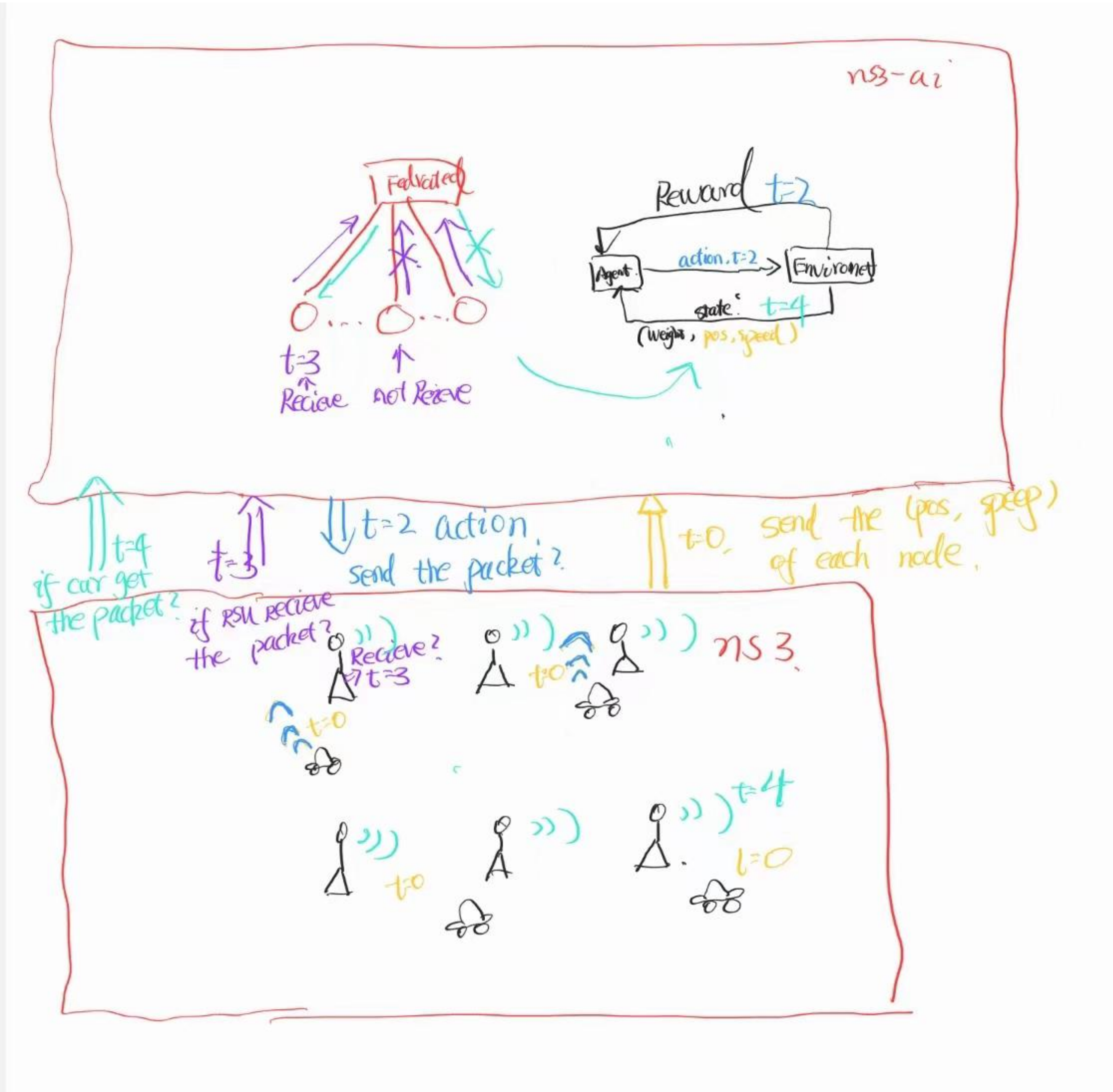
  (3)Nodes in ns3 start to send data packets, and return which nodes sent packets received by RSU. Return information to ns3-ai

  (4)Global aggregation on ns3-ai side

  (5)The ns3 terminal simulates the RSU sending data packet node, and outputs the receiving information

  (6)Update the corresponding client model in ns3-ai.

  (7)repeat the first step



Scheme B

Main challenge:

1. Build a federated learning framework for Internet of Vehicles in ns3

- Main process:

  (1) Use ns3 to build the FL framework, simulate sending data packets at a certain time interval, and simulate packet loss in the uplink and downlink

  (2) Read the simulation results of ns3 directly and modify them in the previous code.

Algorithm comparison:

(1) FedAvg
(2) FedProx
(3) Fed Adam
(4) Self-designed federated reinforcement learning algorithm scheme (A or B) + Adam

(5) Self-designed federated reinforcement learning algorithm scheme (A or B) + SGD

(6) Self-designed federated reinforcement learning algorithm scheme (A or B) + PGD

Compare the above algorithms

Under the same model structure (including CNN and LSTM), learning rate, batch size, training round

Metric

Accuracy, mae, communication overhead represented by the number of communication rounds

Experimental procedure

- Set different transmission powers for vehicle nodes and observe the impact of various indicators
  - Determine transmit power and model(CNN)

    1.FedAvg

    2.FedProx

    3.Fed Adam

    4. Self-designed federated reinforcement learning algorithm scheme (A or B) + Adam
    5. Self-designed federated reinforcement learning algorithm scheme (A or B) + SGD
    6. Self-designed federated reinforcement learning algorithm scheme (A or B) + PGD
  - Determine transmit power and model(LSTM)

    1.FedAvg

    2.FedProx

    3.Fed Adam

    4. Self-designed federated reinforcement learning algorithm scheme (A or B) + Adam
    5. Self-designed federated reinforcement learning algorithm scheme (A or B) + SGD
    6. Self-designed federated reinforcement learning algorithm scheme (A or B) + PGD

References

1.https://www.nsnam.org/wiki/GSOC2013WAVE_MAC#Vehicluar_Networks_and_VANET_simulations