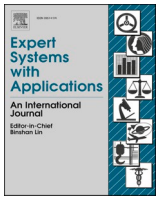
Expert Systems With Applications 227 (2023) 120295

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Bandit-based data poisoning attack against federated learning for autonomous driving models

Shuo Wang [a], Qianmu Li [a,*], Zhiyong Cui [b,*], Jun Hou [c], Chanying Huang [a,1]

[a] School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei Street, 210094 Nanjing, China
[b] School of Transportation Science and Engineering, Beihang University, 37 Xueyuan Road, Haidian District, 100191 Beijing, China
[c] School of Social Science, Nanjing Vocational University of Industry Technology, 210046 Nanjing, China

ARTICLE INFO

Keywords:
Poisoning attack
Federated learning
Model security
Autonomous driving

ABSTRACT

In Internet of Things (IoT) applications, federated learning is commonly used for distributedly training models in a privacy-preserving manner. Recently, federated learning is broadly applied to autonomous driving for training intelligent decision models without disseminating local data remotely. Although federated learning provides a safer training manner for protecting data privacy in autonomous driving, the model training process is still vulnerable to poisoning attacks from vehicle client ends. It is beneficial to study poisoning attacks for enhancing the robustness of the training process to generate reliable decisions for safe driving. Until now, a few researches on poisoning attacks against classification models under federated learning scenarios have been proposed. However, those poisoning attacks against classification tasks cannot be directly applied to regression tasks in a federated learning framework, especially autonomous driving tasks such as steering angle control and brake control. The biggest challenge is that the output of non-linear regression models in federated learning is a dynamic sequential value decided by an online updated non-linear function. Thus, minor attacks can affect the overall non-linear function inference outputs leading to a failed stealthy attack. Based on existing challenges, this paper proposes an **ATT**ack against **F**ederated **L**earning based **A**utonomous **V**ehicle framework(**ATT_FLAV**) to evaluate and enhance the robustness of the federated learning-based autonomous driving models and take the steering angle control task as a representative non-linear regression task to illustrate the methodology. In the proposed framework, a bandit-based **A**ttack**R**egion-**UCB** (**AR-UCB**) algorithm is designed for dynamic data poisoning attacks against the non-linear regression model. This is a black-box attack strategy that chooses the target attack label region dynamically in each federated learning round based on historical attack experiences. Compared with the attack performance of baseline poisoning attacks and the robustness under defense schemas, the proposed poisoning attack strategy can achieve superior attack performance via continuous data poisoning attacks against the federated learning framework.

1. Introduction

Recent years have witnessed a rapid development of the Internet of Things in many industry fields. Autonomous driving as one of the most critical fields has been developed rapidly and has attracted much research attention in smart transportation. Autonomous driving aims to imitate human driving actions via object detection, path planning, steering control, etc. Steering control is one of the key components aiming to decide the driving directions the same as a human.(Feng, Yan, Sun, Feng, & Liu, 2021; M P, A., R, G., Panda, M., 2021; Zhang et al., 2021). Until now, machine learning based autonomous driving steering models can provide steering assistance with high accuracy by extracting features on the image and learning a regression mapping between the image and the steering angle. In IoT scenarios, most autonomous driving model is trained offline based on centralized learning on the cloud server (Chi & Mu, 2017; Fernando et al., 2017; Gidado et al., 2020; Rausch et al., 2017). In this way, heterogeneous data on various vehicles will be sent to the cloud server for training a global shared steer model.

* Corresponding authors at: School of Computer Science and Engineering, Nanjing University of Science and Technology, 200 Xiaolingwei Street, 210094 Nanjing, China (Qianmu Li); School of Transportation Science and Engineering, Beihang University, 37 Xueyuan Road, Haidian District, 100191 Beijing, China (Zhiyong Cui).
E-mail addresses: sharon_wang@njust.edu.cn (S. Wang), qianmu@njust.edu.cn (Q. Li), zhiyongc@buaa.edu.cn (Z. Cui), houjunnjust@163.com (J. Hou), hcy@njust.edu.cn (C. Huang).
[1] ORCID: 0000-0002-1314-4949.

https://doi.org/10.1016/j.eswa.2023.120295
Received 12 August 2022; Received in revised form 22 March 2023; Accepted 25 April 2023
Available online 28 April 2023

However, centralized training on cloud computing meets several challenges: (1) Transferring raw data (including images and other sensor data) of each vehicle client to the server exposes data at risk of privacy leakage. Self-driving data contain front view images from left, center, and right views captured by cameras on autonomous vehicles, latitude, longitude, gear, brake, accelerator, steering angle, and speed captured by other sensors. Specifically, steering angles are usually predicted based on captured front-view images and the corresponding steering angles which can reflect sensitive information including the driver's routine and driving behaviors. Thus uploading sensitive images and steering angles to the server could reveal privacy; (2) Disseminating raw images from vehicle clients to the server puts much more pressure on the network transmission bandwidth and is possible to cause packet loss and processing delay; (3) Images captured by different vehicles are all heterogeneous since those vehicles are moving in different scenarios. A shared pre-trained autonomous steering model trained offline on the cloud server cannot fit heterogeneous data on various edge ends well. Thus, in edge computing, it is necessary to train online with real-time data from each vehicle edge end to improve the detection performance of the global steer model. Federated learning is an efficient online learning way for training a shared model safely for isolated data islands.

Many federated learning based autonomous driving frameworks in edge computing environments are proposed recently (Savazzi, Nicoli, Bennis, Kianoush, & Barbieri, 2021; M P, A., R, G., Panda, M., 2021; Nguyen et al., 2021; Zhang et al., 2021). The basic Federated Learning based Autonomous Vehicle framework (FLAV) is shown in Fig. 1. In this framework, autonomous driving vehicles as edge devices play the role of client ends in federated learning. Each vehicle is equipped with cameras for capturing the driving road images and an autonomous steering angle control module based on real-time collected front-view images. Several autonomous vehicles are gathered in a group with one central server end and exchange the model gradient or the model weight information between the central server end. Each vehicle owns its private data and does not share data with other vehicles and the central server. As vehicles collect more images with variable views from respective cameras in each round, the steering model is updated by training on each client end.

Even though the federated learning for autonomous driving models is more stable than traditional centralized training, the edge end which directly collects image data from data sources is still vulnerable to being invaded, especially poisoning attacks (Jiang et al., 2019). Specifically, once the training process on one of the client ends is interfered with dirty data or model, the maliciously trained model will contaminate the global model by aggregation on the server end, and other honest clients (i.e normal clients, who do not provide the malicious updates proactively) by global model distribution respectively. Since FL learning is vulnerable to the poisoning attack on the edge client end, untrusted devices should be carefully added to the system. Hence, the study of the attack on the FLAV reveals the weakness of the framework and ultimately helps enhance the robustness of the FLAV framework.

Existing poisoning attacks on federated learning concentrate on classification tasks such as object detection (Xie, et al., 2020; Huang, 2020; Jagielski et al., 2021; Sun et al., 2021). Only a few research paid attention to the attack against regression task(Li et al., 2021; Şuvak et al., 2022). Nevertheless, none of these attacks are adapted for the federated learning-based non-linear regression learning task with the following several challenges.

Firstly, most poisoning attacks against federated learning focus on model poisoning which is the White-box attack with large attack costs and is not time-sensitive. Poisoning attack includes data poisoning attack and model poisoning attack (Lyu et al., 2020). On one hand, model poisoning attacks need to know the model structure and training details. However, in the real-world federated learning framework for autonomous driving, only limited data can be accessed by attackers, i.e. black-box attacks. On the other hand, model poisoning is usually conducted offline based on optimization methods. Thus model poisoning usually costs unlimited optimization times and cannot be applied to real-time scenarios in the real world. Nonetheless, a data poisoning attack is more feasible to be implemented in reality due to the simple data manipulations on the edge end(Wang et al., 2022; Zhuang et al., 2022a; Zhuang et al., 2022b). Specifically, label-flipping attack is one of the basic data poisoning methods with fewer attack costs. Considering the data privacy security and the limited data storage space on the vehicle client ends, data captured in each round on each client end will not be preserved in the long term and not be shared between different client ends or different training rounds. Attackers can only access and utilize the data in the current attack round and the updated global model shared by the server end. Hence, data information that can be used by attackers in real-world federated learning is more constrained.

Secondly, the output of the regression model is continuous rather than discrete in the classification task. The poisoning attacks on classification tasks usually change the decision boundary which divides the feature vectors into multiple regions to decide the classification category of each input data point. In classification tasks, a slight perturbation in the training data may change the decision boundary easily to mislead the classification decision of the targeted attacked category. However, the output of non-linear regression tasks is decided by the non-linear function. Tampering on a small portion of data in the non-linear regression task may cause the overall non-linear function to change in uncertain directions, especially for those untampered data points. Thus, choosing target labels casually to attack may not produce expected

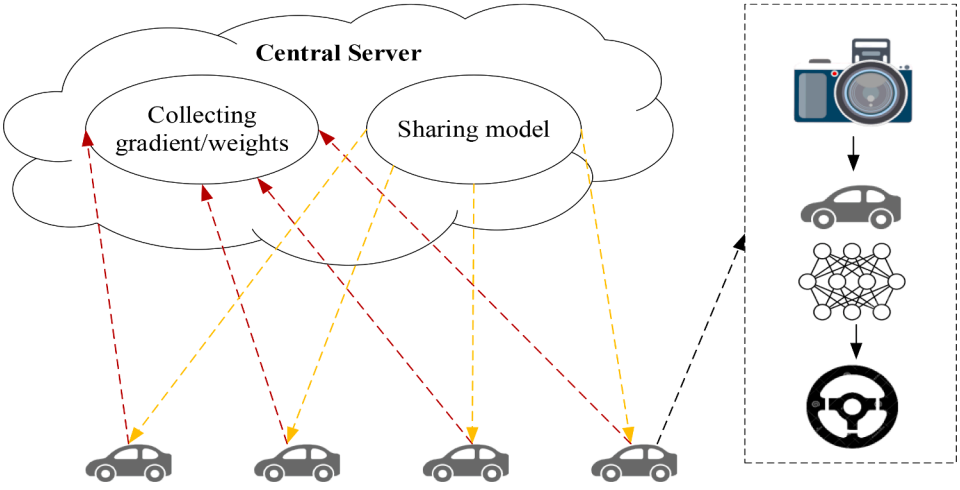


Fig. 1. The overview of the federated learning based autonomous steering control system.

attack effects.

Thirdly, the FLAV is a sequential learning and inference process with defense aggregation schemes on the server end. The aggregation schemes merge the honest models and malicious models on the server end, thus can discount the attack from the attack on client ends in each round. When the attack target of common poisoning attacks on FLAV is fixed in each round, the malicious models can be easily found after several rounds and cannot obtain excellent attack effects through the whole sequential federated learning process under the aggregation and anomaly detection schemes.

In summary, designing an efficient and enduring black-box attack against federated learning based autonomous driving still meets multiple challenges: 1) Poisoning attacks on real-world federated learning based autonomous driving models lack research on Black-box attacks. However, the data information that can be accessed by attackers is limited and time-sensitive which needs Black-box attacks. 2) Some poisoning attack methods can be applied to the linear regression model but it is hard to carry out an efficiently targeted attack on the non-linear regression model. 3) The common aggregation schema on the federated learning server end can discount the attack effect from a small amount of malicious client ends, leading to failed attacks. Considering these challenges, existing poisoning attacks cannot be transferred directly to attack non-linear regression models based on a federated learning framework. It is necessary to propose a new data poisoning attack that can come out with persistent attack effects on non-linear regression models with limited data information for these special applications.

In recent years, the bandit strategy has been used in several online black-box attacks by exploiting prior information (Guan et al., 2020; Ilyas et al., 2019; Sundar et al., 2020). Bandit is commonly used for dynamic sequential decisions with historical statistical experience to maximize the accumulated rewards. Hence, black-box attacks can benefit from a bandit strategy with only short-term data and limited model information. However, existing bandit-based attacks are all applied in generating online adversarial samples or selecting attack targets, especially for inference attacks for classification models Ilyas et al., 2019; Sundar et al., 2020), but do not poison the training process. Considering the advantages of the bandit, it is expected to be used to choose target attack labels in each federated learning round dynamically for label-flipping poison attacks. Bandit-based dynamic attack label chosen strategy can help improve label-flipping attack effects and the robustness, especially for the attack against federated learning based non-linear regression models, and would only need limited data and model feedback information.

Based on the above analysis, we propose a black-box attack framework for federated learning based autonomous driving model and a bandit-based data poisoning method against the FLAV framework for improving the sequential data poisoning attacking effect. The ultimate research objective of the attack is to help enhance the robustness of the autonomous driving model and complement the research on the stability of the non-linear regression task in the federated learning framework. In a word, the main contributions are summarized as follows:

1. The **ATT**ack against **F**ederated **L**earning based steering angle control of **A**utonomous **V**ehicle framework (**ATT_FLAV**) is proposed. It targets to discover the vulnerability of federated learning for autonomous driving applications, especially for non-linear regression tasks. This framework ultimately aims to evaluate and improve the robustness of the federated learning based autonomous driving models.

2. A black-box data poisoning attack algorithm in the ATT_FLAV framework is proposed for injecting malicious training data to the malicious client ends with the least-cost malicious manipulations. This algorithm is used to attack the FL training process of non-linear regression models and generate sequential attacks on the global model and other honest client models in each round.

3. A bandit-based target attack label region selection strategy named **AttackRegion-UpperConfidenceBoundBandit (AR-UCB)** is proposed for maximizing the attack performance on the online non-linear regression tasks under the FLAV framework. AR-UCB is used to choose the best target attack label region dynamically in each round of non-linear regression tasks based on the attack experiences without storing history data and models for the long term on each client end.

4. The attack performance of the proposed attack algorithm is compared with baseline and other label-flipping attack strategies and verified to obtain more attack rewards. Moreover, the robustness of the proposed attack strategy is examined under different defense aggregation schemes on the server end.

The rest of the paper is organized as follows. Section 2 introduces the background and related works of federated learning based autonomous driving systems and poisoning attack methods on regression tasks. Section 3 states the proposed methodology in detail. Section 4 exhibits the experiment settings and experiment results. Section 5 concludes the work of the paper and proposes future work directions.

2. Background and related works

2.1. Autonomous driving system based on federated learning

As autonomous driving models develop greatly on inference accuracy, many researchers have paid attention to the security and efficiency of autonomous model training. The traditional autonomous driving model is trained based on a centralized training mode. However, centralized training challenges the computation capacity of the server, the security of the data, and the transmission overheads of the network. Recently, the federated learning framework is gradually adopted for training autonomous driving models.

Distributed federated learning can help vehicles fuse their multi-dimension sensor data from various vehicles and improve real-time decision accuracy(Savazzi et al., 2021). Elbir et al. (Elbir et al., 2020) investigate the federated learning application in an intelligent vehicular network. They show the basic federated learning process in a vehicular network. M P et al. (M P et al., 2021) proposed a federated learning based steering angle prediction framework by transmitting model parameters between the central server and edge devices on vehicles. Zhang et al. (Zhang et al., 2021) implement an end-to-end federated learning framework and verify it in the autonomous driving scenario. Both of these methods input the image frames into CNN on edge devices and exchange the model between the server and edge vehicle ends. The training time and transferred bytes based on federated learning are greatly reduced with better model accuracy than centralized machine learning. Nguyen et al. (Nguyen et al., 2021) designed a Federated Autonomous Driving network for improving the stability and accuracy of the autonomous driving model. In their framework, there are several server ends that can communicate with each other. However, multi-party communications increase the risk of data privacy exposure.

Therefore, we consider the federated learning based autonomous driving framework shown in Fig. 1 where only client ends and the only server end can exchange model information in each round. In each round of federated learning, each edge client end trains the global model based on its own local data and uploads the updated model or the gradient to the federated learning server. Then the collected edge client models are aggregated into a new global model on the server end. The training scheme avoids the raw data transmission between the server and the edge end. Only the gradient or the model weight is exchanged to avoid the security and computation load problems mentioned in centralized training.

2.2. Poisoning attack on FL

Even though federated learning provides privacy-preserving training which enhances data security greatly, the model on the edge end is still vulnerable to be attacked. Model robustness attacks can be divided into poisoning attacks and inference attacks. An inference attack tampers the model output during the inference phase by distorting or falsifying the

input inference data. For example, adding noise or the backdoor trigger on the inference data can mislead the model to output a false inference result. However, in federated learning systems, inference attack can only mislead the client that is being invaded but not all federated member clients and is not easy to be implemented in reality. The robustness of federated learning can be easily weakened by poisoning attacks in real-world scenarios. A poisoning attack manipulates the training process by contaminating the data or the model gradient, divided into data poisoning attacks and model poisoning attacks. Once the attacker invades at least one of the edge clients in the federated learning system, the data or the model is fragile to be tampered maliciously and the poisoned model on the malicious client end will be uploaded to the server to contaminate the global model. The malicious model and the benign model collected from malicious and benign clients separately will be aggregated together on the server end. Then other clients can be terminated by distributing the aggregation model to other clients in the next model update round. In reality, model poisoning attack is still limited in implementation. They are usually based on optimization methods with enormous efforts and computation complexity. Moreover, model poisoning attacks need much model privacy information for optimization. Thus, we pay more attention to data poisoning attacks in real autonomous driving scenarios.

Data poisoning is more feasible and can undermine the model's robustness in real-world federated learning based autonomous driving systems. Specifically, there are several data poisoning attacks that focus on label-flipping attacks which are easier to implement with fewer manipulation costs among all data poisoning attacks. Most label-flipping attacks are based on dirty labels i.e. falsifying the labels directly. Biggio et al.(Biggio et al., 2011) first implemented the heuristic-based adversarial label-flipping attack against the SVM classifier and verified the better performance compared with the random label-flipping. They repeat flipping those labels with non-uniform probabilities until the farthest attack samples from the SVM hyperplane to achieve the maximum classification error. Xiao et al. (Xiao et al., 2012) developed an optimization framework for adversarial label-flipping. They describe the problem of finding the best label flips as two minimization problems, i.e. minimizing the loss of training on the clean dataset and the contaminated dataset. Paudice et al. (Paudice et al., 2019) introduced an algorithm to craft label-flipping attacks under the worst security scenario with less computation cost. They greedily select the flipped labels that maximize the validation error desired by the attacker in each iteration. Taheri et al. (Taheri et al., 2020) use silhouette and k-means clustering methods and select samples with lower inner-cluster values to flip labels to attack the Android malware systems by deep learning model. Zhang et al. (Zhang, Cheng, Zhang, & Li, 2021) proposed two label-flipping attacks based on entropy and k-medoids separately for the attacking Naive Bayes based spam filter system.

However, these attacks are all applied to classification tasks and CNN based autonomous driving model is a non-linear regression task. The output of the non-linear regression is a continuous value rather than a discrete value in the classification task. Transferring the attack on the classification task to federated learning based non-linear regression task meets challenges.

2.3. Poisoning attack on regression

Some researchers study the attack on the regression task. Jagielski et al. (Jagielski et al., 2018) first proposed the poisoning available attack on linear regression task. They resolve the problem by formulating the regression attack as a bi-level optimization problem. The outer level optimization is to select the poisoned point to maximize the loss function on the clean validation dataset. The inner-level optimization is to maintain the regression algorithm on the training dataset including both clean and poisoned datasets. However, this optimization is not suitable for the deep neural network model which is a non-linear regression task considering the high-complex computation costs.

Ghafouri et al. (Ghafouri et al., 2018) linearize the neural network at each operating point and find the optimal attack solution around the point on regression tasks in cyber-physical systems. Then, a regression-based defense is proposed to mitigate the attack. Suvak et al. (Şuvak et al., 2022) designed a bilevel optimization equation to model the attack and the original regression task. Meng et al. (Meng et al., 2019) designed an adversarial sample generation method of evasion attack for regression tasks. Gupta et al. (Gupta et al., 2021) proposed a Jacobian-based adversarial attack generation method for regression tasks. They find the worst perturbation that leads to the maximum deviation of attack results and evaluates the deviation by Lp-norm. Liu et al. (Liu et al., 2019) proposed to transform the poisoning regression task to find the solution in a non-convex trust region. Muller et al. (Muller et al., 2020) proposed a flipping attack for a non-linear regression task and a defense mechanism. The attack algorithm picks the candidate poisoned data point by calculating the instance distance to the upper or lower end of the feasible field and flipping it to another side of the feasible field. Nonetheless, none of these regression attacks are applied in federated learning based non-linear regression tasks.

3. Proposed methodology

This section introduces the proposed attack framework and the data poisoning attack algorithm. Section 3.1 describes the problem in formulation, including the basic federated learning for the autonomous steering control model, the poisoning attack against the federated learning framework, and the specific attack object of the poisoning attack against the federated learning based steering control model. Section 3.2 states the detailed structure of the proposed attack framework named ATT_ FLAV, based on federated learning for training autonomous driving framework, and Section 3.3 explains the bandit-based data poisoning algorithm applied in ATT_FLAV in detail.

3.1. Preliminary

This section describes the background of ATT_FLAV and the problem to be solved in ATT_FLAV. The basic federated learning framework includes distributed parallel gradient descent and centralized model aggregation demonstrated in Section 3.1.1. Next, for discovering the vulnerability of the FLAV framework, a data poisoning attack from the client ends against FLAV is considered and described formally in Section 3.1.2. Moreover, for enhancing the attack performance against federated learning for a non-linear regression model, a bandit strategy is exploited for dynamic label-flipping data poisoning attack and is formulated in Section 3.1.3. The preliminary notations used in this paper are listed in Table 1.

3.1.1. Federated learning for training an autonomous steering control model

This study primarily focuses on the data poisoning attack against federated learning (FL) under autonomous steering control scenarios. Under the autonomous driving scenarios, the FL architecture attempts to learn and improve the autonomous steering control model based on crowdsourcing data collected from private vehicles. Due to communication costs and privacy issues, private vehicles cannot upload collected data to the server end, especially the images with sensitive.

information collected from vehicles' front cameras. Hence, in this study, the FL architecture incorporates a global autonomous driving model f^g on the central cloud server end and a set of client models $\left\{f^1, \ldots, f^i, \ldots, f^n\right\}$ on n vehicles. To describe the steering control procedure in simplicity, on the client vehicle end i, the collected image data X^i are fed to f^i to generate the steering angle y^i such that

$$y^i = f^i\left(x^i; w^i\right) \tag{1}$$

Table 1
Preliminary notations.

w_t	Global weight shared with client ends in round t
w_{t+1}^i	The updated weight of client i for sending to the central server in round t + 1
$\widetilde{w}_{t+1}^l$	Malicious model weight on client i trained by malicious dataset for sending to the central server in round t + 1
$T_{a,t}$	The number of times that action a is selected in the past t times in UCB bandit
$Q_t(a)$	The estimated reward of action a at time t in UCB bandit
C	The confidence value that control the exploration ability in UCB bandit
A_t	The chosen action at time t in UCB bandit
γ	Learning rate for model training
L	Loss function for model training
$D_t^i(X,Y)$	Training dataset on client i in round t with image set X and corresponding label set Y
$\widetilde{D}_t^l(X,\widetilde{Y})$	Malicious data set on client i in round t with image set X and corresponding malicious label set $\widetilde{Y}$
n	Number of client ends
def	Target deflection angle
$Agg(w_t^1,\cdots,w_t^i,\cdots,w_t^n)$	Aggregation scheme on the central server
m	Number of images in each sub-dataset
$\widetilde{m}$	Number of images in test dataset
$Y = \{y_j \mid j = 1,2,\cdots j\}$	Ground truth steer angle of image j
$\widehat{Y} = \{\widehat{y}_j \mid j = 1,2,\cdots j\}$	Output from the trained model of image j
$U = \{u_1,\cdots,u_v,\cdots,u_V\}$	Divided v unit label regions
$U_{att}{}^{(t)}$	Target attack label region at round t
$R_v{}^{(t)}$	Step reward of attacking label region u_v at round t
$R_{acc}{}^{(t)}$	The average accumulated step reward for all attack regions u_v of all attack rounds t
$BRP^{(t)}$	The probability of obtaining the best reward for all candidate label region at round t

where w^i is the weight of model f^i. During the FL process, in round t, each client i trains the client model f^i copied from the server end with their own dataset $D_t^i = \{x^1, x^2, \cdots, x^i\}$. But only the model weights can be uploaded to update the global model to avoid privacy issues, and the client training procedures can be described as

$$w_{t+1}^i = w^i{}_t - \gamma \nabla L\left(D_t^i\right) \tag{2}$$

where w_{t+1}^i is the updated weight on client i, γ is the learning rate, L is the loss function while training f^i, and $\nabla L(\bullet)$ is the gradient. Please note that since the images on vehicle client ends will not be uploaded to the server, they cannot be stored on vehicles permanently either. Thus, the dataset on each client will be updated periodically, and that's why D_t^i has the t subscript. Then, the trained model weights are collected by the server end and aggregated to update the global model for sharing in the next round $t + 1$:

$$\mathrm{w}^g{}_{t+1} = \mathrm{Agg}\left(w_t^1, \cdots, w_t^i, \cdots, w_t^n\right) \tag{3}$$

where Agg is the aggregation scheme for updating the global model, which is normally an average function. After the global model is updated, it will be shared with all the clients in the next round. In other words, in the next round t + 1, all the client models are copies of the global model.

3.1.2. Poisoning attack in federated learning

In this study, we consider the attack on the FL-based autonomous driving model. Although the data collected by vehicles can be fully utilized under the FL framework, the autonomous steering model deployed on a vehicle client end is vulnerable to data poisoning attacks when it is iteratively trained and updated with the new coming training data. The malicious client is an attacked client given a poisoned dataset $\widetilde{D}_t^l$. The attacking effect can be reflected in the modeling updating process:

$$\widetilde{w}_{t+1}^l = w_t - \gamma_t \nabla L\left(\widetilde{D}_t^l\right) \tag{4}$$

where $\widetilde{w}_{t+1}^l$ is the malicious model weight after the model's training dataset is poisoned. The malicious client uploads the contaminated weight trained with $\widetilde{D}_t^l$ to the server end. Other honest clients still update their model weights based on Equation (2) and send the clean weights to the server. Then, the malicious weight and other clean weights are aggregated together on the server end to generate the new global model weight to be shared in the next round, described as Equation (5), where Agg is normally an average function.

$$w_{t+1}^g = \mathrm{Agg}\left(w_{t+1}^1, \cdots, \widetilde{w}_{t+1}^l, \cdots, w_{t+1}^n\right) \tag{5}$$

3.1.3. Black-box targeted attack in a Non-linear regression model

Data poisoning is one of the black-box attacks by attacking the training data and does not need to access the model structure and parameters. Furthermore, a label-flipping attack is one of the data poisoning attacks that can be easily deployed in real-world attack scenes with fewer attack costs. Thus, a label-flipping attack is commonly used in existing research. In other research, a stealthy poisoning attack normally only affects a small portion of the model's output region and retains the model's overall performance. Especially under the FL framework, if the training labels of the model on the malicious client are all or widely flipped, it is easy to be detected by defending components on the server end. However, none of the existing works propose the data poisoning attack against federated learning based non-linear regression task, which is commonly carried out in real-world scenes, such as online training for autonomous driving steering control and autonomous vehicle speed detection. Moreover, attack parts of regression inputs also meet challenges as stated in Section 1. Thus, it is necessary to use a confined label-flipping attack method for federated learning based non-linear regression tasks.

In this study, the steering control model is taken as the representative of the non-linear regression model whose output should be a continuous steering angle value. The steering angle values can be normalized ranging from −1 to 1 which is symmetrically centered on 0. Here, the targeted region label-flipping attack on the steering control model refers to moving the model's output steering angle y into a target region $u_j = (u_{j\_start}, u_{j\_end})$ where $-1 \le u_{j\_start} < u_{j\_end} \le 1$, by adding target deflection to the target attack labels. In other words, only the samples in $\widetilde{D}_t^l$ with the labels within the target continuous region u_j are contaminated. However, selecting a proper target label region to attack FL based non-linear regression model is a hard problem for several reasons: 1) Since the output of the non-linear regression model is continuous, the attacking effect of any one of the continuous regions is uncertain so that the attack label region need to be chosen carefully; 2) the vehicle clients collect real-time images from varying driving environments to train the steering control model, and thus, the effect of attacking a fixed target region may differ when time varies. Therefore, choosing the target region U_{att} of the label-flipping attack from a set of potential regions $U = \{u_1, \cdots, u_v, \cdots, u_V\}$ to maximize the average accumulated attack effect $R_{acc}{}^{(t)}$ during t rounds in FL process is the goal of the targeted attack on the FLAV, which can be described as

$$\max_{U_{att}} R_{acc}{}^{(t)}, U_{att} \in U \tag{6}$$

The detailed computation of $R_{acc}{}^{(t)}$ is described in Section 3.3. In summary, this paper focuses on maximizing the attack effect in the continuous online FLAV by designing a strategy to choose the U_{att}. The detailed federated learning based autonomous driving framework and the label-flipping attack strategy is stated in the following sections.

3.2. Proposed framework

This subsection introduces the framework of **ATT**ack against **F**ederated **L**earning based steering angle control of **A**utonomous **V**ehicle (**ATT_ FLAV**). In the real-world, experimental autonomous vehicles gathered in a union with other private vehicles are assigned to collect data on various roads for training a shared model. These experimental vehicles are connected to the internet so they may suffer vulnerability attacks such as cyber security attacks and then be deliberately manipulated by the attacker or be the attacker itself. (Kim et al., 2021). Once the attacker takes control of the vehicle, data and the model training process can be deliberately attacked to mislead the inference results of the intelligent model. Due to the characteristics of federated learning, one of the poisoned models will spread to the global model on the server end and local models on the other vehicle end as well. The influence is broad in the federated learning framework and endangers driving safety. Thus, we further explore the weakness of this federated learning framework, especially the vulnerability under data poisoning attack, to enhance the robustness of the framework for evading driving accidents.

ATT_FLAV is proposed in this section for attacking the training process of federated learning based non-linear regression model. This framework can be applied to discovering the risk of federated learning for non-linear regression tasks, such as traffic speed monitoring(Pu et al., 2022), traffic flow density estimation(Pu et al., 2021), and steering angle control tasks(Kong et al., 2020). ATT_FLAV is beneficial for finding out the weakness and improving the robustness of FLAV. Moreover, ATT_FLAV is a dynamic low-cost black-box attack framework that exploits limited data and parts of model feedback, especially can be applied in real-world edge computing environments. Here, the steering control model, which is a non-linear regression model, is regarded as the use case model of the proposed attack framework. We suppose parts of the participant vehicle ends are malicious. The attack objective of this framework is to obtain the poisoned global model by attacking parts of vehicle ends and then spreading the poison to all participant vehicle ends. Specifically, the poisoned global model is expected to output the target deflection angle away from the correct angles in the target steering angle region in sequential learning processes. The framework of **ATT_ FLAV** is shown in Fig. 2.

Fig. 2 displays two consecutive rounds of the framework when one of the vehicles is poisoned. It also shows how the poisoning effect diffuses from the poisoned vehicle client to the global model and other honest client ends. The framework consists of three main components, poisoning attacks on the malicious vehicle ends, local model training on edge client ends, and global model aggregation on the server end, as shown in different colors and step numbers in Fig. 2 separately. In the first step, a label-flipping data poisoning attack is imposed on the malicious vehicle ends to tamper parts of the training data labels. The attack objective is the steering angle control model which is representative of the non-linear regression model. Attacking a federated learning-based non-linear regression model may fail due to the dynamic training data in federated learning and the aggregation schema on the server end (as stated in Section 1). Hence, the target label region to be attacked for label-flipping is chosen dynamically based on the bandit strategy as shown in the bottom right in Fig. 2. After the label-flipping attack is finished in the first step, the malicious vehicle ends training the model based on the poisoned training data and honest vehicle ends train their own models with normal data, referring to the medium part numbered Block (2). Then all client models are uploaded to the server end and aggregated to the new global model for initializing client models in the next federated learning round, as shown in the upper numbered Block (3) in Fig. 2. At this time, the malicious model has tampered the global model and the poisoned global model will spread the poison to other honest client ends in the next training round. More detailed framework and poisoning attack processes are described as follows.

The ATT_FLAV process: In the ATT_FLAV framework, the central server collects model weights from all participant client ends at each training round t as the conventional federated learning framework. The collected weights are aggregated into the global weight w_t on the server end, and it is shared with all the vehicle clients. Those client ends retrain the model based on their own collected data and update the model weight w_t into w_{t+1}. Suppose the client l is the only malicious client end that trains the received global model w_t and generates a malicious model with the weight $\widetilde{w}_{t+1}^{l}$. The malicious weight is then uploaded to the server end to poison the global model. At the same time, all other clients upload their own updated model weights to the central server for aggregation in the next round $t+1$. Once the global model is aggregated based on all the weights $\{w_{t+1}^{1}, \cdots, \widetilde{w}_{t+1}^{l}, \cdots, w_{t+1}^{n}\}$ collected from honest clients and malicious clients, the poison can be distributed to other client ends by sharing this updated global model in the next round. Consequently, the new round of model updates and poisoning attacks are

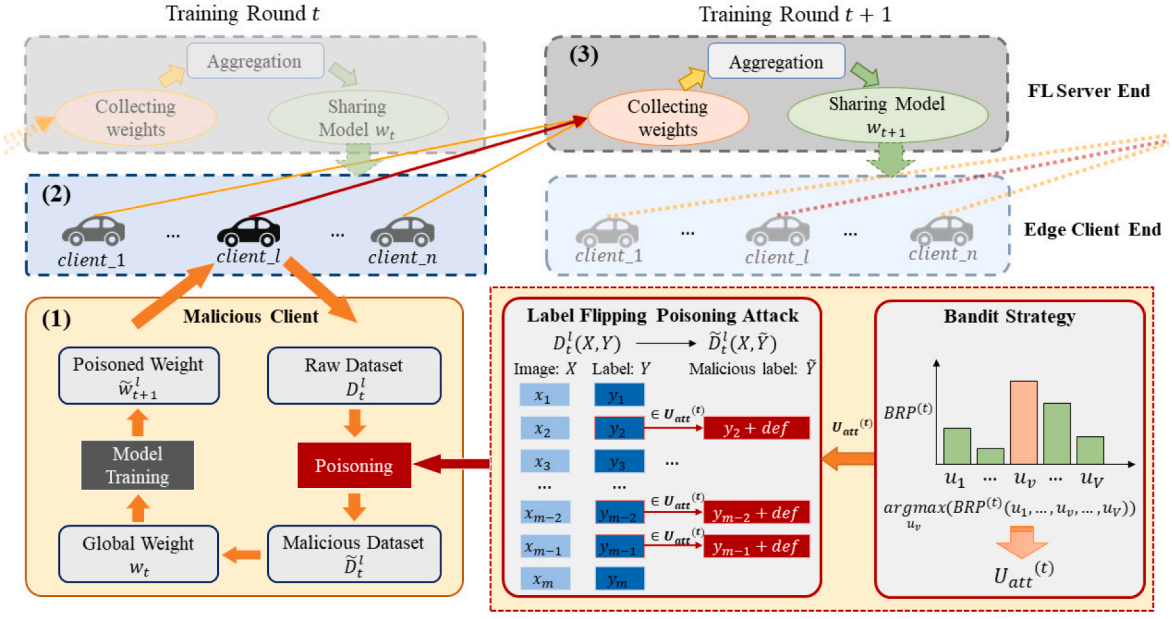


Fig. 2. Framework of Data Poisoning Attack on Federated Learning for Autonomous Steering Angle Control System.

proceeding iteratively as the process described above. At the same time, the poison caused by the malicious client end is injected into all participant client ends in each round.

Label-flipping Poisoning Attack: More specifically, the intentionally generated malicious input on the malicious client end is shown in the bottom right with the red dotted box in Fig. 2. Malicious input contains two parts, normal samples with clean labels and target attack samples with target labels. Normal samples with clean labels refer to the raw training data without any malicious manipulation. These samples have the raw image data X and the corresponding raw label Y. Target attack samples are composed of raw image data with the corresponding flipped label $\widetilde{Y}$ which is generated by adding the *def* to Y whose value locates in the U_{att}.

Bandit-based Label-flipping Attack: In the sequential data poisoning attack in the ATT_FLAV framework, a bandit strategy is proposed to choose one of the label regions from the candidate label region set as the target attack region ${U_{att}}^{(t)}$ at round t. The objective of choosing ${U_{att}}^{(t)}$ is to obtain the greatest attack gain (reward) in a sequence of learning rounds. The attack reward is evaluated by Best Reward Probability (BRP) score of each label region, introduced in the next section in detail. In this attack framework, the label region set is predefined, such as $U = \{u_1, \cdots, u_v, \cdots, u_V\}$, representing V divided label regions to be the candidate attack targets. Once the target attack region ${U_{att}}^{(t)}$ is decided in each round, the malicious input dataset can be reconstructed for training the local poisoning model.

Finally, the malicious vehicle ends upload the malicious model weight to the central server round by round with the maximum accumulated attack reward which is tested on the global model.

3.3. Poisoning attack on federated learning

In the proposed ATT_FLAV framework, we mainly illustrate the label-flipping data poisoning attack algorithm in federated learning and the bandit-based label selection strategy for label-flipping attacks against non-linear regression tasks. The steering angle control model is regarded as the use case of autonomous driving for demonstrating the proposed poisoning attack.

3.3.1. Poisoning attack in FL

The proposed framework has three types of major components, including one server end, honest clients, and poisoned clients. The corresponding data poisoning attack algorithm on the ATT_FLAV composed of three functions is detailed in Algorithm 1. The *Server_Aggregate* function describes the weight update and aggregation on the server end of the FL framework. The *Client_Update* and *Malicious_Client_Update* functions describe the detailed weight update procedures on honest and attacked vehicle clients, respectively. In Algorithm 1, we assume there is only one server end and n vehicle clients in the federated learning aggregation group including one malicious vehicle client.

Before the federated learning-based steering control algorithm starts, a pre-trained steering control model is deployed onto the server end and all vehicle ends. The model weight on each vehicle client is updated round after round based on the new data samples in parallel. For example, the new training dataset at round t on the i-th is represented by $D_t^i = D_t^i(X, Y)$, where X represents the input image frames collected by the autonomous vehicle's front camera and Y is steering angle label set of the corresponding frames. It should be noted that the data collected by client vehicles are heterogenous, i.e. $D_t^i \neq D_t^j$, since they are assumed to be traveling on different roads or environments. All the vehicle clients update their model weights as Equation (2) with the clean datasets.

Algorithm 1: Data Poisoning Attack on the Edge End

Initialization pre-trained steering control model with model weight w_0
Input: initial global model w_t at round t, training samples D_t^i from the i-th vehicle client, the pre-defined deflection value *def*.

(*continued on next column*)

(*continued*)

Algorithm 1: Data Poisoning Attack on the Edge End

Output: malicious model $\widetilde{w}_{t+1}^l$ trained by poisoned data
Function Malicious_Client_Update(w_t, D_t^l, *def*):
  w_t = **Server_Aggregate()**
    collect new training samples D_t^l from the attacked vehicle client
  ${U_{att}}^{(t)} \leftarrow$ **AR-UCB**(D_t^l, w_t, U)
  for x, y in $D_t^l(X, Y)$ **do**:
    if $y \in {U_{att}}^{(t)}$:
      $y \leftarrow y + def$
  end for
  $\widetilde{D}_t^l(X, \widetilde{Y}) \leftarrow D_t^l(X, Y)$
  $\widetilde{w}_{t+1}^l = w_t - \gamma \nabla L(\widetilde{D}_t^l)$
return $\widetilde{w}_{t+1}^l$ to server
end Function

Function Server_Aggregate():
  for each round t = 0, 1, 2, …**do**
  for each vehicle client $i \in \{1, 2, ..., n\}$ $(i \neq l)$ ***in parallel*** **do**
    $w_{t+1}^i \leftarrow$ **Client_Update**(w_t^i)
  end for
  $\widetilde{w}_{t+1}^l \leftarrow$ **Malicious_Client_Update**(w_t^l)
  $w_{t+1} = \frac{1}{n}(\sum_{i=1, i \neq l}^{n} w_{t+1}^i + \widetilde{w}_{t+1}^l)$
  end for
end Function

Function Client_Update(w_t):
  receive w_t from server
  collect new training samples D_t^i from the i-th vehicle client
  $w_{t+1}^i = w_t - \gamma \nabla L(D_t^i)$
return w_{t+1}^i to server
end Function

The weight update has a data poisoning procedure in this proposed framework. The malicious vehicle client l follows the *Malicious_Client_Update* function to renew the model weight w_{t+1}^l. Since the non-linear regression property of the steering control model greatly affects the poisoning attack performance in the FL framework, the first step is to choose a target region to maximize the attacking performance as well as to let the poison attack evade the FL's defending component on the server end. In this study, we propose a bandit-based label-flipping attack region selection method, i.e. the AttackRegion-UCB(AR-UCB) function, which is further detailed stated in Algorithm 2.

After the attack region U_{att} is determined at round t, the collected data on the malicious client D_t^l will be poisoned by the label-flipping method. All the data samples' labels in the attack region $(y \in {U_{att}}^{(t)})$ are tempered from y to $y + def$, where *def* is a predefined target deflection angle. In this way, the dataset on the malicious client $D_t^l(X, Y)$ is poisoned and converted into $\widetilde{D}_t^l(X, \widetilde{Y})$. Finally, the model on the malicious client is trained based on $\widetilde{D}_t^l$, and the model weight on the malicious client w_t is updated into $\widetilde{w}_{t+1}^l$ based on Equation (4). After all the clients update their model weights, the central server collects weight from both honest client ends and malicious client ends. The collected weights are averaged and aggregated into w_{t+1} for the new round $t + 1$. Then, w_{t+1} is shared with all clients for the next learning round.

3.3.2. Bandit-based Label-flipping attack

The mainstream autonomous driving steering models are based on neural networks which are non-linear regression models to predict continuous steer values. Considering the property of the targeted non-linear regression attack, bandit algorithms are taken into consideration for choosing the target steer region to attack dynamically. To this end, the AR_UCB algorithm is designed especially for non-linear regression models with continuous-valued inputs and outputs. For presenting the AR-UCB algorithm step by step, we introduce the basic UCB

bandit algorithm in sub-section A. Then we clarify the role and the changes made on the basis of the basic UCB used for target attacks in AR-UCB in sub-section B. At last, sub-section C introduces the algorithm process of the proposed AR-UCB attack based on the revised UCB strategy in ATT_FLAV.

3.3.2.1. UCB bandit. The bandit algorithm is one of the online learning algorithms commonly used in reinforcement learning.(Kuleshov & Precup, 2014; Lattimore & Szepesvári, 2020). Bandit is used to resolve the multi-armed bandit problem which aims to gain the maximum reward by exploring environments and exploiting the best statistical experience. Exploring is to find out the arm that can gain the maximum reward with the greatest probability by testing each arm and the corresponding reward distribution. Exploit is to gain as many rewards as possible by taking the best action. The most important thing is to decide on the selection strategy. Upper Confidence Bound (UCB) algorithm is one of the classical algorithms for choosing the arm to achieve the best reward in the long-term. In the early rounds, each action is performed once for obtaining the initial reward distribution of each action. In the later round, UCB chooses one action a from the potential actions that satisfies the following equations.

$$BRP^{(t)}(\mathrm{a}) = Q_t(a) + \sqrt{\frac{C \bullet lnt}{T_{a,t}}} \tag{7}$$

$$A_t = \underset{a}{argmax}\left[BRP^{(t)}(\mathrm{a})\right] \tag{8}$$

In Equation (7), $T_{a,t}$ is the number of times that action a is selected in the past t times. Moreover, C is the confidence value that controls the exploration ability. The larger C means the more probability of choosing other potential actions. $Q_t(a)$ is the estimated reward of action a at time t, representing the exploitation effect of action a based on experiences. The root term in Equation (7) calculates the upper confidence bound of the reward so that the reward value is below with bound $BRP^{(t)}(\mathrm{a})$. A larger $T_{a,t}$ gives a smaller upper bound leading to a larger probability to choose other actions rather than action a. Finally, the UCB algorithm chooses the action A_t at time t as Equation (8) to balance exploration and exploitation. In UCB, the action that maximizes the upper confidence bound is chosen. Thus, UCB gives more opportunities to explore other actions with a strong potential to obtain optimal rewards instead of selecting the best-estimated reward all the time.

3.3.2.2. UCB in target label attack. The action selection strategy of UCB can be applied in the dynamic selection of the target attack regions in the target attacks in each round of the federated learning framework. In the non-linear regression task, taking the steering control task as an example, considering the model's output is a continuous value, it requires splitting the range of the output value into discrete sub-regions as the potential attacking targets to fulfill the target label attack.

In this study, the ultimate goal of the dynamic target label attack is to achieve the maximum accumulated attack reward in the long-term and against the defense on the server end at the same time. Specifically, for each attack region v, the estimated attack performance, i.e. attack reward, at each round t is defined as

$$R_v{}^{(t)} = \sum_{j=1}^{m} \frac{1}{m}\left(\left|y_j{}^{(t)} - \widehat{y}_j{}^{(t)}\right|\right), y_j{}^{(t)} \in u_v{}^{(t)} \tag{9}$$

where $u_v{}^{(t)}$ is the target attack label region, $y_j{}^{(t)} \in u_v{}^{(t)}$ is the ground truth steer angle in round t, and $\widehat{y}_j{}^{(t)}$ is the model output whose labels are in the same range $u_v{}^{(t)}$ at round t. m is the number of samples whose labels locate in range u_v at round t. Here, the step attack reward $R_v{}^{(t)}$ is defined as the average deviation between the ground truth steering angle and the detection value in the target attack region based on the data on the malicious client at each attack round. Then, the average step reward of action u_v during t rounds $\overline{u_v}(t)$ is updated based on $R_v{}^{(t)}$

$$\overline{u_v}(t) = \sum_{1}^{t} \frac{1}{T_{v,t}} R_v{}^{(t)} \tag{10}$$

where $T_{v,t}$ is the test times of the action u_v in t rounds. $\overline{u_v}(t)$ is used to update the best reward probabilities ($BRP^{(t)}$) score that instructs the probability of gaining the best attack reward for each candidate label region. $BRP^{(t)}$ is updated as the manner in Equation (7) but replacing $Q_t(a)$ with $\overline{u_v}(t)$ as Equation(11).

$$BRP^{(t)}(u_v) = \overline{u_v}(t) + \sqrt{\frac{C \bullet lnt}{T_{v,t}}} \tag{11}$$

The square root term is the bonus of the action u_v. On one hand, the more times action u_v is chosen, the smaller the square root is, so that other actions are more likely to be explored. On the other hand, if the $\overline{u_v}(t)$ is large enough, i.e. the average reward of u_v during t rounds performs well, UCB tends to exploit the high-quality action u_v for gaining the best accumulated reward. Then in each round t, the attack label region u_v with the greatest reward probability will be selected as the target attack label region $U_{att}{}^{(t)}$ by Equation(12). Specifically, $BRP^{(t)}(u_v)$ is a list initialized by attack label region in $(u_1, \cdots, u_v, \cdots, u_V)$ once again to gain the initial reward for each label region.

$$U_{att}{}^{(t)} = \underset{u_v}{argmax}\left(BRP^{(t)}(u_1, \cdots, u_v, \cdots, u_V)\right) \tag{12}$$

Furthermore, the accumulated reward of each action in each round is shown as Equation (13). The average accumulated attack reward $R_{acc}{}^{(t)}$ is the average accumulated step reward for all attack regions u_v of all attack rounds, which intuitively is used to evaluate the overall attack performance. Since $R_{acc}{}^{(t)}$ is calculated based on previous $R_{acc}{}^{(t-1)}$, it is initialized as 0 at the first step, i.e. $R_{acc}{}^{(0)} = 0$, and will be assigned values based on R_{step} and R_{acc} from step 1.

$$R_{acc}{}^{(t)} = R_{acc}{}^{(t-1)} + \frac{1}{t}\left[R_v{}^{(t)} - R_{acc}{}^{(t-1)}\right], R_{acc}{}^{(0)} = 0 \tag{13}$$

3.3.2.3. AttackRegion-UCB. AttackRegion-UCB(AR-UCB) strategy is designed to choose the target label region to be poisoned in each round for maximizing the accumulated attack reward throughout the federated learning iterations as shown in Algorithm 2. Since the steering angle values are distributed symmetrically centered on 0 normalized ranging from −1 to 1, the attack effect on (-1, 0) and (0, 1) are similar. Here, we take the positive label region, i.e. steering angle label in (0,1) as the representation to test the poisoning attack effect. For example, we divide the label region (0,1) into four regions (i.e. actions in the bandit algorithm).

In Algorithm 2, in the first V rounds, each label region in U is attacked once a time to initialize the $BRP^{(t)}$ for each region as Line 1–8. That is to say, the attack label region is fixed in the first V rounds. The labels of the selected region in each round is tampered with *def* which is the target steer deviation defined by the attacker. Then the steering angles $\widehat{Y}$ of the client's own dataset D_t^i is inferenced based on the shared global model w_t. Note that w_t represents the poisoned global model attacked in the round (t-1) and we test the w_t with the dataset in round t for evaluating the attack effect of attack in round (t-1). Hence the inference outputs $\widehat{Y}$ are used to indicate the attack reward of the label region chosen in the round (t-1) as described in Lines 5–8. The step reward of the chosen label region in the last round is calculated in Line 5 and update the average reward $\overline{u_v}(t)$ of region v is in Line 6. Then $BRP^{(t)}$ is updated in Line 7. After all regions are attacked traverse, BRP is initialized completely with the initial BRP of each candidate attack region. At the same time, the average accumulated step reward $R_{acc}{}^{(t)}$ is computed in line 8 for evaluating the general attack reward until now.

After initialization, $U_{att}{}^{(t)}$ is chosen based on $BRP^{(t)}$ updated in each

round as Lines 10–14. Lines 10–13 are similar to the initialization. The main difference is that after the $BRP^{(t)}$ is updated, ${U_{att}}^{(t)}$ is chosen for the most probability to gain the greatest attack reward. Finally, the chosen target label region to attack in round t is returned to data poisoning attacks and the training label in ${U_{att}}^{(t)}$ can be poisoned. Moreover, the step reward for evaluating the attack effect of the last round (t-1) choice, and the average accumulated reward for reflecting the general attack effect until round t are also generated for monitoring the attack in the continuous federated learning.

Algorithm 2: AttackRegion-UCB

Input: D_t^i, w_t, U
Output: ${R_v}^{(t)}$, ${R_{acc}}^{(t)}$, ${U_{att}}^{(t)}$
1 **if** attack from round $t \le V$: **// *BRP initialization for each action***
2 $v = t$**//*Traverse label region U in the first V rounds***
3 ${U_{att}} = u_v$
4 $\widehat{Y} \leftarrow f^i(D_t^i; w_t)$
5 ${R_v}^{(t)} = \sum_{j=1}^{m} \frac{1}{m}\left(\left|y_j^{(t)} - \widehat{y}_j^{(t)}\right|\right), y_j^{(t)} \in u_v{}^{(t)}$**//*Update step reward of label region U***
6 $\overline{u_v}(t) = \sum_1^t \frac{1}{T_{v,t}} {R_v}^{(t)}$
7 $BRP^{(t)}(u_v) = \overline{u_v}(t) + \sqrt{\frac{C \bullet lnt}{T_{v,t}}}$**//*Update BRP*$^{(t)}$**
8 ${R_{acc}}^{(t)} = {R_{acc}}^{(t-1)} + \frac{1}{t}[{R_v}^{(t)} - {R_{acc}}^{(t-1)}]$**//*Update average accumulated rewards***
9 **else: // *choosing* ${U_{att}}^{(t)}$ *based on* $BRP^{(t)}$**
10 $\widehat{Y} \leftarrow f^i(D_t^i; w_t)$
11 ${R_v}^{(t)} = \sum_{j=1}^{m} \frac{1}{m}\left(\left|y_j^{(t)} - \widehat{y}_j^{(t)}\right|\right), y_j^{(t)} \in u_v{}^{(t)}$
12 ${R_{acc}}^{(t)} = {R_{acc}}^{(t-1)} + \frac{1}{t}[{R_v}^{(t)} - {R_{acc}}^{(t-1)}]$
13 $BRP^{(t)}(u_v) = \overline{u_v}(t) + \sqrt{\frac{C \bullet lnt}{T_{v,t}}}$
14 ${U_{att}}^{(t)} = \underset{u_v}{\operatorname{argmax}}(BRP^{(t)}(u_1, \cdots, u_v, \cdots, u_V))$ **//Choose action based on** $BRP^{(t)}$
15 **return** ${R_v}^{(t)}$, ${R_{acc}}^{(t)}$, ${U_{att}}^{(t)}$

4. Experiments

This section introduces the dataset, autonomous steering angle control model used in the FLAV framework, baseline attack methods for comparison, metrics for evaluating the model performance, parameter configuration used in our experiments, and the results of four sets of experiments in the following sections separately.

4.1. Dataset

The federated learning based autonomous steer model is implemented based on the public autonomous driving dataset Udacity[2]. Udacity is a dataset prepared for the autonomous driving algorithm competition and is commonly used for autonomous driving research. The driving data is captured from San Mateo to Half Moon Bay including the curvy and highway driving, with the resolution of 640 x 480 for each frame. In addition to the images taken by the vehicle, the data set also includes the attributes and parameter information of the vehicle itself, such as latitude and longitude, brake, accelerator, steering degree, and speed. This dataset provides the real scenario of autonomous driving for training and testing the proposed models. Existing steering angle control models are almost based on this dataset.

The image data is captured by cameras deployed on the experimental vehicles from left, center, and right cameras and divided into six sub-datasets, i.e. HMB1, HMB2, HMB3, HMB4, HMB5, HMB6. For steering angle control, researchers usually use the images captured from the center view with manual labels. There are 404,916 frames for training (including HMB1, HMB2, HMB4, HMB5, and HMB6) and 5614 frames for testing (HMB3) with standardized steering labels ranging from −1 to 1. Fig. 3 displays four sets of original autonomous driving steering samples and the example of the corresponding attacked steering results. Fig. 3(a) and (b) present the original and the malicious right deflection results. In Fig. 3(a) and (b), the second and the third images are the malicious steering angle results with deflection of 0.2 and 0.4 to the right in the green and blue curves, respectively. Fig. 3(c) and (d) present original and malicious left deflection results similar to Fig. 3(a) and (b). A single steering control attack on one frame can affect the steering angle and even push the steering angle in the opposite direction. Furthermore, continuous steering control attack increases driving safety significantly. We randomly distribute the training data to all 5 vehicle clients and 500 training rounds of the federated learning autonomous driving system without repetition. In other words, for imitating the heterogenous data distribution, each vehicle client contains a unique image subset different from each client end and each round in federated learning.

4.2. Autonomous driving model

Considering the limited storage and computation resources on the edge end, we choose a CNN model with less model size and high accuracy, the SteerCNN[3] as the autonomous steering control model. This model is supposed to be deployed on the edge client end in the federated learning autonomous driving system. The model performs well on Udacity Challenge[4], for steering angle detection and is implemented based on (Deng et al., 2020). The test RMSE is 0.10 evaluated based on the Udacity dataset.

4.3. Baseline models

To the best of our knowledge, there is no existing poisoning attack on federated learning based non-linear regression task on the edge end. Thus, we take the conventional label-flipping from Xiao et al. (Xiao et al., 2018) as the Random-attack to show the attack effect of our proposed algorithm. Besides, we compare the proposed AR-UCB with other label region selection strategies, i.e. ∈-greedy strategy(Kuleshov & Precup, 2014). Zhang et al. (Zhang et al., 2020) studied reward-poisoning attacks against reinforcement-learning agents by ∈-greedy strategy. Ma (Ma, 2021) studied the adversarial attacks in sequential decision-making, including the ∈-greedy strategy. In our experiment, we reuse ∈-greedy from Zhang et al. and Ma and make a small adjustment to choose the attack region in the federated learning, named ∈-greedy-attack. In addition, we compare our attack method with another data poisoning attack method against a non-linear regression model proposed by Muller et al. (Muller et al., 2020), abbreviated as Flip2Corner in this manuscript. Flip2Corner is the first to evaluate poisoning attacks on non-linear regression models. Since it does not consider the federated learning situation, we adjust the attack details for adapting to the federated learning situation. The detailed settings of the baseline models are described as follows:

Random-attack: In each iteration round, the attacker chooses the attack region randomly following the conventional label-flipping rules.

∈-greedy-attack: ∈-greedy is to choose the action in each round formulated as Equation (14). In each round, a random value α is generated for choosing the action. If the random value is less than the pre-defined ∈ value, a random region U is selected to be the attacked one in our proposed scenarios. Otherwise, the region with the max accumulated reward is chosen to be attacked. In other words, smaller ∈ value

[2] *Self-Driving-Car/Steering-Models/Community-Models/Rambo at Master · Udacity/Self-Driving-Car*, 2017.

[3] *Self-Driving-Car/Steering-Models/Community-Models/Cg23 at Master · Udacity/Self-Driving-Car*, n.d.

[4] *Udacity. Udacity challenge 2: Steering angle prediction. https://bit.ly/2E3vWyo, 2017.*

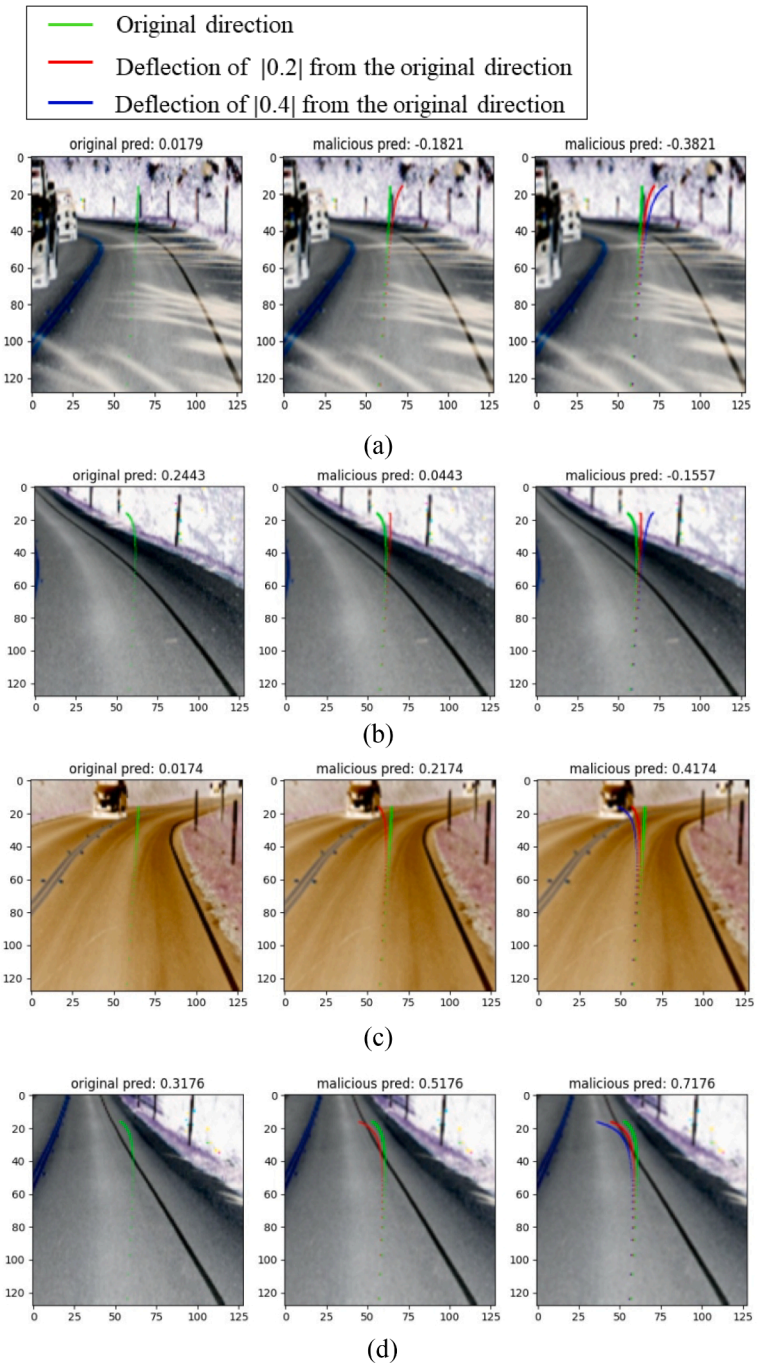


Fig. 3. Data example. All steer angles are shown in a standardized range. In each set of data examples, from the left to the right, the image presents the original direction, the direction that deflects |0.2| from the original directions and deflects |0.4| from the original curve in green, red, and blue curves respectively. The first two sets and the last two sets display the right and left deflections with negative and positive deflection values respectively.

has larger probability to choose the region with the greatest accumulated reward based on the previous selection experiences. In comparison experiments, we choose two ϵ values 0.1 and 0.05 separately for comparison.

$$U = \begin{cases} argmax(U(v)) if \alpha \geq \epsilon, \\ randomchoice if \alpha < \epsilon \end{cases} \tag{14}$$

Flip2Corner: In this study, we follow the basic idea of (Muller et al., 2020) to generate poison training data in each updating round in federated learning. Firstly, we take all training data in each updating round as the candidate substitute data and take 75% of them to poison. The feasibility domain is [-1,1] in our steering angle control task. Then we calculate the maximum distance to the lower or upper end of each data point and choose the data with 75%-highest distance as the candidate poison data. Next, flip each candidate poison data label to the feasible domain boundaries that are farther away. At last, the poisoned data will be concatenated with clean data as the training data for updating the local model in the current federated learning round. Based on this poison data generation methodology, we repeat the operation in each updating round before training the local model.

4.4. Evaluation metrics

4.4.1. Attack effect

The autonomous driving model performance can be evaluated by Rooted Mean Square Error (RMSE) described as Equation (15), where y_i and $\widehat{y}_i$ are the ground truth steering angle of the i-th frames separately, $\widetilde{m}$ is the total number of test frames. The smaller RMSE represents better detection accuracy. We verify the attack effect based on the average accumulated attack reward. The average accumulated attack reward is the average accumulated step reward of all attack rounds. Step attack reward is defined as the average deviation between the ground truth steering angle and the detection value in the target attack region based on the data on the malicious client in each attack round. The formal definition is shown in Equation (10). In particular, Random-attack and ∈-greedy-attack are the targeted attack against the federated learning framework so that these two methods can be compared by calculating average accumulated step reward in the exact target label region selected in each federated learning updating round. Since Flip2Corner does not select the target label region but selects the label according to the distance in each round, we compare Flip2Corner with AR-UCB based on the test RMSE of the aggregated global model on the server end.

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{\widetilde{m}} \|y_i - \widehat{y}_i\|^2}{\widetilde{m}}} \tag{15}$$

4.4.2. Attack against defense schemes

We also evaluate the attack effect against the typical defense schemes applied in poisoned federated learning. Federated learning usually uses an aggregation scheme on the central server end to exclude the collected malicious client ends. The most common aggregation scheme in federated learning is the FedAvg which is to aggregate client models by averaging model weights. This aggregation scheme in federated learning can counteract the effect of uploaded malicious model weights to some degree. FedAvg is the basic aggregation schema in FLAV on the central server as Equation (16). With the development of attacks on federated learning, other heuristic-based aggregation schemes are also proposed against strong attacks.

Blanchard et al.(Blanchard et al., 2017) designed Krum to aggregate the weight based on the Euclidean distance between collected client weights. Krum filters out a model weight that is closest to (n-q-2) adjacent weights in the squared Euclidean norm space among all the collected weights as shown in Equation (17–19). Where q is the max number of malicious client end among all client ends. In our experiment, q is set as 1 among all 5 participant client ends. Krum calculates the Euclidean distance between each client model weight (w_t^i) and other weights (w_t^j) as $d_{ij} = \|w_t^i - w_t^j\|^2, (i,j = 1,2,\cdots n)$. Then for each client weight, sort d_{ij} and calculate the sum of the closet (n-q-2) weight distance($d_{ij}{}'$) to w_t^i as the score of the w_t^i ($score_i$), where n is the total client number, q is the maximum malicious client number.

Yin et al. (Yin et al., 2018) proposed the Trimmed Mean defense aggregation schema. Trimmed mean sorts the uploaded parameters of client models and drops the largest and smallest model parameters. Then the mean of the remaining model parameters is taken as the updated global model for sharing with other clients ends in the next federated learning round. Trimmed Mean can be formulated as Equation (20). S is the set of remaining models after removing the largest and smallest model parameters and s is the s-th remaining model.

$$\textbf{FedAvg}: \ w_{t+1} \leftarrow \sum_{i=1}^{n} \frac{1}{n} w_t^i \tag{16}$$

$$\textbf{Krum}: d_{ij} = \|w_t^i - w_t^j\|^2, (i,j = 1,2,\cdots n) \tag{17}$$

$$score_i = \sum_{i=1}^{n} d_{ij}{}', (j = 1,2,\cdots, n-q-2) \tag{18}$$

$$w_{t+1} \leftarrow w_t^i = \mathrm{argmin}(score_i), i \in (1,2,\cdots,n) \tag{19}$$

$$\textbf{TrimmedMean}: w_{t+1} \leftarrow w_t^i = \frac{1}{n-2} \sum_{s \in S} w_t^s \tag{20}$$

4.5. Parameter setup

In this study, the experiments set up one malicious vehicle client end among all 5 vehicle ends. All the steering angles are standardized into (-1,1). Since the image data is horizontally symmetric, we only attack the steering angle region between (0,1). The steering angle region is divided into four regions, (0,0.2), (0.2, 0.4), (0.4, 0.6), (0.6,1), and only one of the regions is chosen to be attacked in each iteration round. The attack target *def* is −0.4 away from the ground truth steering angle. Both centralized and federated learning based steering angle model training set the learning rate as 1e-4, batch size as 32 and Adam is chosen as the optimizer. The image is resized to 128*128 uniformly as the input of both models. Each attack is carried out from the 30th round to the 500th round. All the experiments are tested five times with five independent random seeds for generating different sub-dataset order in each training round, and all rewards in the experimental results are averaged given the five different random seeds to avoid testing noises.

4.6. Numerical results

This section shows the numerical results of four sets of experiments to evaluate the performance of our proposed label-flipping attack. Firstly, we verify the performance of the steering angle control model trained by federated learning. Secondly, we show the successful and failed attack examples based on the verified robust model. In the third part, we compare our UCB-based attack with baseline models to illustrate the attack performance of AR-UCB attack. The fourth set of experiments is under different parameters in AR-UCB and compares the attack rewards of multiple target deflection steering angles and attack beginning rounds. In the last set of experiments, we display the attack results against typical federated learning defense schemas based on AR-UCB attack.

4.6.1. FLAV performance

A robust model trained by federated learning is the base of the attack framework. There is no need to attack and explore the vulnerability of a poor model. Hence, before the attack, it is necessary to verify the performance of the steering angle control model trained by federated learning.

In this section, we verify the effect of the model trained by federated learning by comparing its inference results with the results of the centralized training. We train the SteeringCNN model based on the Udacity training dataset by both centralized training and federated learning. In centralized training, we divide the training data into 300 subsets average and train one of the sub-dataset in each training round. The total training round is 300 rounds and the training batch in each training round is 32. We test the trained model on the test dataset after each training round. The test results are shown by RMSE in Fig. 4 in blue. From the test results, the model trained by centralized training obtain 0.6 RMSE after the first training round and then RMSE quickly reduces to 0.2 after 25 training rounds. Then the model begins to converge from the 100th round. The test RMSE after converged keep around 0.12. In federated learning, we do the same splitting on the training dataset. There are five client ends for federated learning and each client end trains the local model based on one of the subdataset. The total federated learning training round is 300 rounds, i.e. the global model is aggregated by 300 times on the server end. The aggregated global model is tested based on the test dataset on the server end after each aggregation. The test results are also shown by RMSE in Fig. 4 in red. From Fig. 4, the RMSE of the first federated learning training round is 0.2 less than 0.6

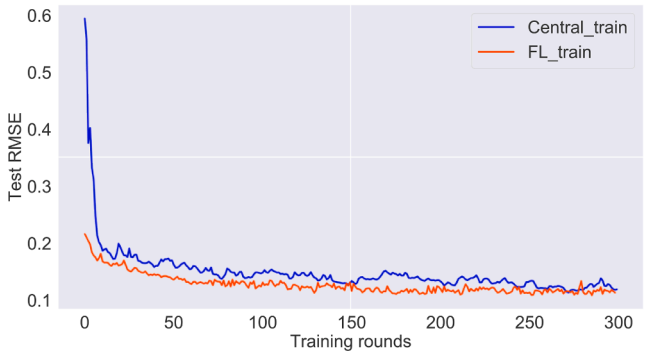


Fig. 4. Comparison between the test RMSE of the SteerCNN model trained by centralized training and federated learning.

for centralized training. The model trained based on federated learning converges to a similar RMSE level of 0.11 after 100 rounds.

Fig. 5 displays the ground truth steering angle and inference steering angle of test images based on the 300th round aggregated global model on the server end separately. The blue line represents the ground truth steering angle and the orange line is the inference angle of the aggregated model with test RMSE 0.11. The two lines are similar in general with only minor inference errors on a few images. Overall, the model trained by federated learning can achieve almost the same accuracy as the centralized training will little acceptable error. The model trained by federated learning can be the target attack model.

4.6.2. ATT_FLAV performance

Based on the analysis of the model trained by federated learning, we further show examples of attacks against the FLAV to illustrate the successful and unsuccessful attack performance. Figs. 6 and 7 exhibit the different inference results of the successful poisoned global model with two attack targets separately. Fig. 6 is the successful attack results when the target attack label range is (0.5,1) and the target deflection is −0.5. The attack object is to encourage the model to output inference results −0.5 away from the ground truth for samples with labels greater than 0.5. From Fig. 6, the inference steering angles of samples with ground truth greater than 0.5 are decreased by −0.5 in general. However, the poisoned model still performs high detection accuracy on samples with non-target label ranges, i.e. labels less than 0.5. The test RMSE is 0.13 a little higher than 0.11 of normal federated learning in Fig. 5. Fig. 7 is the attack results when the target attack label range is (0.25,1) with target deflection –0.5. The difference between attacks in Figs. 6 and 7 is their different attack label range (0.5,1) and (0.25,1). If the attack is successful as shown in Fig. 7, the inference steering angle of samples with ground truth greater than 0.25 will be decreased by −0.5 in general and keep the performance of the model on samples with other label ranges. Comparing Figs. 6 and 7, the attacked samples in Fig. 7 are more than Fig. 6 so the overall RMSE 0.18 is also much higher than 0.13 in Fig. 6. Thus, attack label ranges in concentrated distribution can reduce the overall inference performance more significantly, and the attack is more easily to be tested based on verifying the test RMSE of the global model. Nevertheless, attack data labels distributed in the tail can keep the general performance of the model greatly and is more stealth for the attack. Hence, attacking a minority of samples instead of the majority samples or the whole data set is more possible to achieve a successful and stealth attack.

The failed attack includes two situations. On one hand, the attack from parts of malicious client ends is defended after the aggregation on the server end. In this case, the attack is invalid so the test results of the global model on the server end will be similar to the normal federated learning as Fig. 5. On the other hand, the attack is valid and the global model detection accuracy is reduced greatly but the model is not available on all samples as Fig. 8. The target flipping label region is (0, 1) and the target deflection is −0.5, meaning that detection results of samples with labels in range (0, 1) are expected to output steering angles in (-0.5, 0.5). From Fig. 8, the inference output (red line) of the majority of samples with ground truth labels (blue line) in (0, 1) are indeed move by −0.5 average. However, since samples with labels near 0 occupy most of the portion in the dataset, the attack on this label region with −0.5 deflection will crash the model on general inferences. In other words, although the model inference results of the target samples are maliciously manipulated in the wrong direction, the detection performance of the whole model is also disturbed extremely. The attack can be easily found soon by monitoring the global model test RMSE on the server end or the inference results on other client ends. Thus, this attack is also considered a failure.

4.6.3. Comparison with baseline attack strategies

This section compares the proposed attack with the targeted Random-attack, ∈-greedy-attack, and untargeted data poisoning attacks against a non-linear regression model.

4.6.3.1. Comparison with baseline bandit-based targeted attacks. Fig. 9 shows the average accumulated reward of the proposed attack (AR-UCB) in the blue solid line and the Random-attack in the gold dotted line, respectively. The lines in the middle of the colored regions are the average of testing results with five different random seeds mentioned in Section 4.5. The upper bound and the lower bound of the blue and gold regions are the largest and the smallest attack rewards of the five times attacks in each round.

Since the attack is carried out from the 30th round, the attack reward of the Random-attack increases suddenly from the 30th round as shown in the golden dotted line. However, the attack reward of AR-UCB rises gradually in the blue solid line. Because the Random-attack selects the label region to poison randomly in each round, the upper and lower bounds of the attack reward change arbitrarily, especially before 100 rounds. While the selected region of the AR-UCB attack in the early attack rounds are relatively fixed with few attack history experiences, the fluctuation of the average accumulated reward is concentrated in the range of less than 0.05.

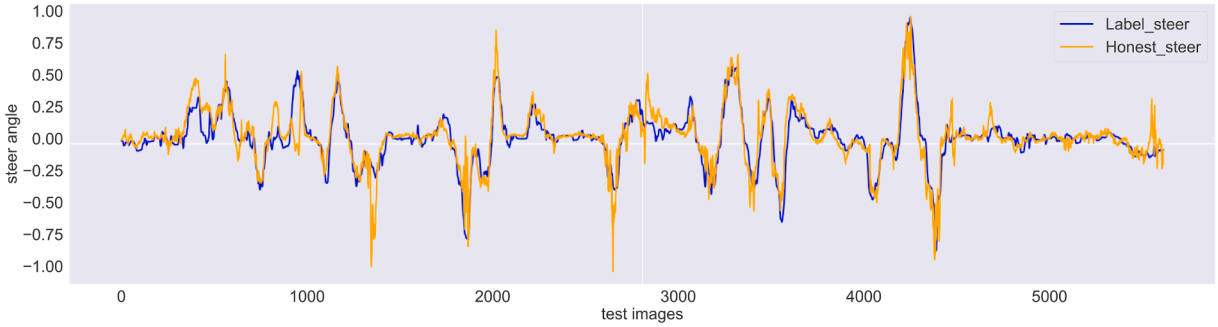


Fig. 5. The test results of the 300th round global model aggregation on the server end in federated learning. The test RMSE of the 300th round is 0.11.

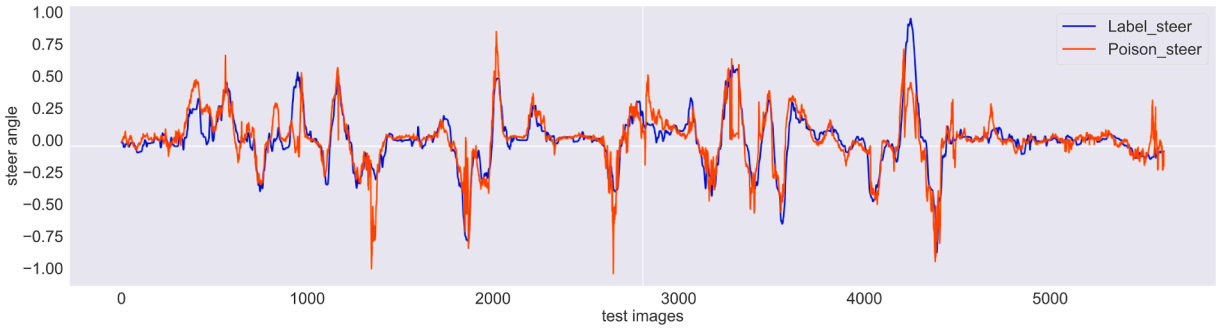


Fig. 6. The test results of the successfully attacked aggregated global model on the server end when attack one of the client ends in federated learning. The attacked label range is (0.5,1). The attack object is to make labels in the target label range deviate from the ground truth steering angle by −0.5.

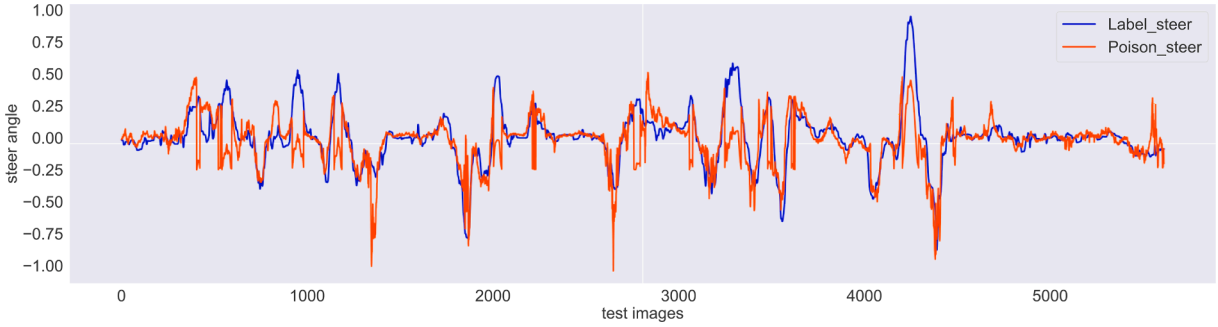


Fig. 7. The test results of the successfully attacked aggregated global model on the server end when attack one of the client ends in federated learning. The attacked label range is (0.25,1). The attack object is to make labels in the target label range deviate from the ground truth steering angle by −0.5.

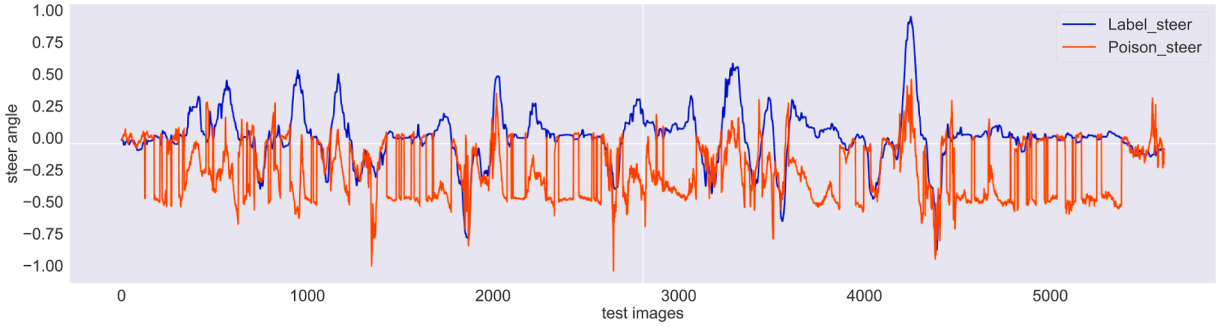


Fig. 8. The test results of the unsuccessfully aggregated global model on the server end when attack one of the client ends in federated learning. The attacked label range is (0, 1). The attack object is to make labels in the target label range deviate from the ground truth steering angle by −0.5.

Furthermore, the fluctuation of the upper and lower bounds is stabilized gradually after 100th rounds for both baseline attack and AR-UCB attack. The range change of the Random-attack is shrunk and stable. In other words, the stochastic label selection cannot introduce targeted influence on the target label region. The magnitude of the fluctuation of the AR-UCB is around 0.05 much more than the baseline. The main reason is that in the first V rounds, AR-UCB attacks each label region once a time for initializing the upper bound probabilities of each candidate attack label region. Afterward, AR-UCB updates the BRP score of each label region after each training round and chooses the specific label region with the greatest reward probability to attack.

Generally, after around 50 rounds, i.e after attacking 20 rounds, the average accumulated attack reward of the Random-.

attack keeps around 0.15. After around 200 rounds (attack 170 rounds), the average accumulated attack reward of the proposed AR-UCB maintains greater than 0.2. AR-UCB is almost 0.05 higher than the Random-attack. Therefore, AR-UCB achieves more attack rewards, especially the accumulated reward which can cause serious steering angle misleading on the sequential autonomous driving decision.

Fig. 10 displays the average accumulated reward of the proposed AR-UCB in the blue line and the ϵ-greedy-attack in the gold dotted line(eps = 0.05) and green solid line(eps = 0.1). For accelerating the optimal selection in fewer iterations, ϵ is set as 0.5 before 200 rounds for both two ϵ-greedy-attack. The ϵ values are set as 0.05 and 0.1, respectively, after 150 rounds. Thus, the attack rewards of two ϵ-greedy-attack are almost the same before 200 rounds. Since when ϵ is 0.5, there is 50% probability of choosing a random label region as the attack target and 50% probability to choose a label region with the best history reward. Therefore, the fluctuation of the upper and lower bounds before the first 200 rounds is larger than that after the 200-th round. After 200 rounds, the label regions with the best history attack experiences are selected with greater probabilities. Thus, more average accumulated attack rewards are achieved to almost 0.2 in each round after 200 rounds. Moreover, the fluctuation concentrates around 0.03, and the upper and lower bounds are between 0.18 and 0.21 after 200 rounds.

The average accumulated reward indicates that ϵ-greedy-attack

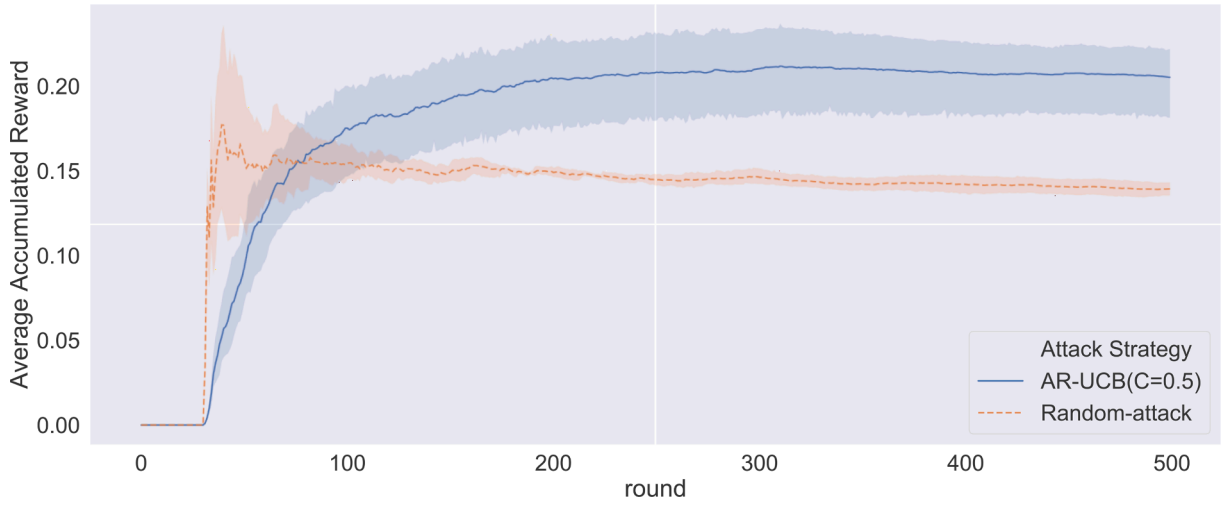


Fig. 9. The average accumulated reward compared with Random-attack.

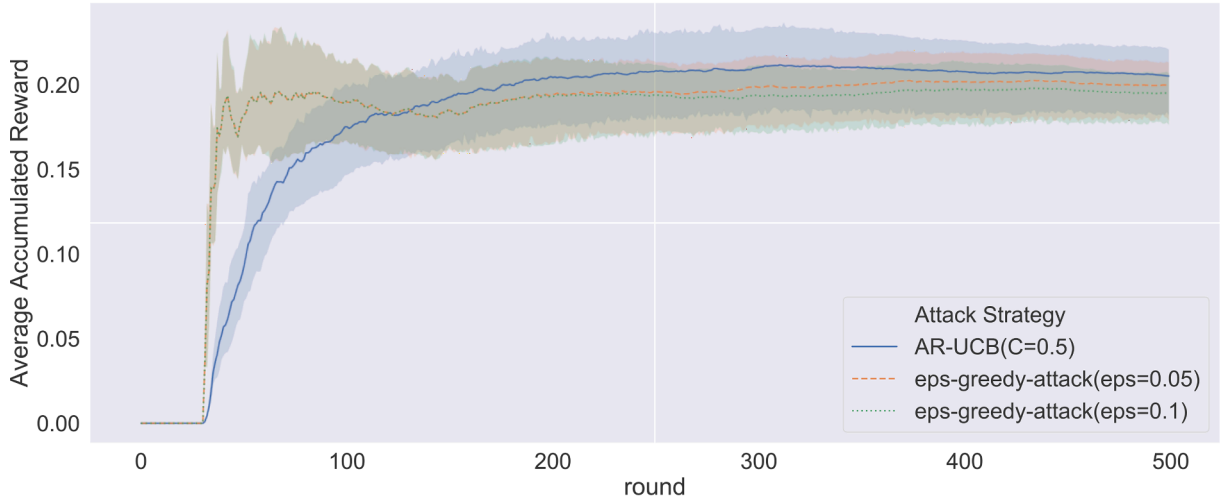


Fig.10. The average accumulated reward compared with eps-greedy attack.

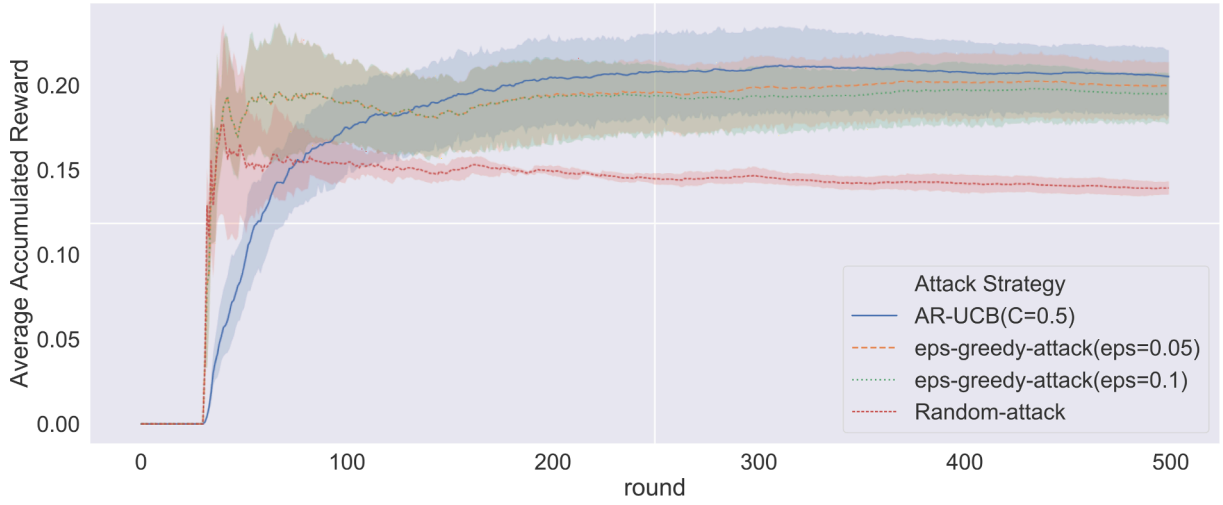


Fig. 11. The average accumulated reward of different label region selection strategy.

keeps almost close but less than 0.2 throughout the attack rounds for both eps as 0.05 and 0.1. The difference between two ∈-greedy-attack is tiny. While the average accumulated reward of the AR-UCB is apparently better than the eps-greedy attack after 120 rounds and maintains a little higher than 0.2 throughout the rest attack rounds. Thus, the proposed AR-UCB still performs better on the accumulated attack reward in the continuous federated learning than ∈-greedy-attack.

Fig. 11 compares AR-UCB, Random-attack and ∈-greedy-attack together. It can be seen AR-UCB, and ∈-greedy-attack all achieve more attack rewards than the Random-attack. In other words, attacking the target label region with the bandit strategy can achieve a better attack effect in the target label region than selecting the label region randomly. Moreover, AR-UCB gains more attack rewards than ∈-greedy-attack in continuous federated learning.

4.6.3.2. Comparison with untargeted attacks on non-linear regression model. Flip2Corner is the representative of attacks against the non-linear regression model but flips the label in all domains. Thus we can only compare the test RMSE based on the test dataset on the server end to compare the attack performance. Fig. 12 shows the comparison of the test RMSE between AR-UCB and Flip2Corner in 200 training rounds. As illustrated in the before sections, the untargeted attack is easier to be found out in the early attack rounds due to the overall corruption of the model. Therefore, if the overall model performance decreased obviously, the attack is still a failure. The attack is added from the 30th training round. The test RMSE of Flip2Corner in the 30th round increased from 0.14 to 0.15 suddenly, obviously higher than AR-UCB RMSE of 0.14. Before 110 rounds, the test RMSE of Flip2Corner is generally higher than AR-UCB. In other words, the stealthy of Flip2Corner is worse than AR-UCB and the model's overall performance is affected more significantly. After 100 rounds, the test RMSE of both attacks fluctuates around 0.13. In general, compared with Flip2Corner, our proposed attack methodology keeps better overall performance on the whole domain so that the attack can survive longer in the continuous attack and achieve a stronger attack effect in the later attack rounds.

4.6.4. Parameter evaluation

This section evaluates the attack parameters from three aspects, including the effect of UCB parameters, target deflection angles, and the beginning round of the attack on the average accumulated rewards. All parameter evaluation experiments are implemented based on the same attack strategy.

4.6.4.1. UCB parameter evaluation. Fig. 13 displays the average accumulated rewards of the proposed AR-UCB with different confidence values C compared with the baseline attack. The confidence value C controls the level of exploration and may affect the label region selection and the attack reward in each round. Hence, we assign two confidence values, 0.5 and 2, and the corresponding models are named AR-UCB(C = 0.5) and AR-UCB(C = 2), respectively. AR-UCB(C = 0.5) and AR-UCB (C = 2) achieve almost the same attack rewards. Both of them achieve higher attack rewards than the Random-attack after the 70th round. The average accumulated reward of AR-UCB grows up since the attack is implemented from the 30th round to more than 0.2 and keeps the same level after the 180th round. The stable attack rewards of both AR-UCB are significantly higher than the baseline attack at the same time.

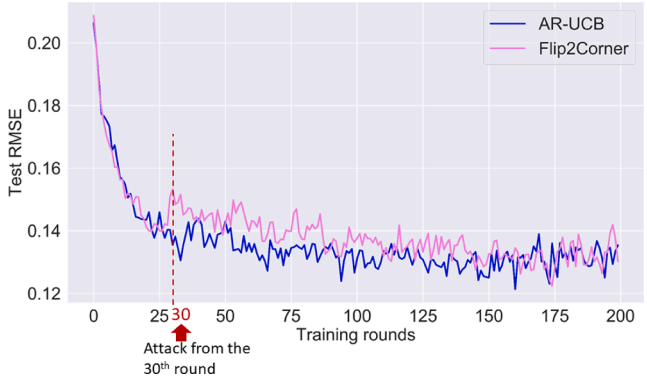


Fig. 12. Test RMSE comparison between AR-UCB and Flip2Corner.

4.6.4.2. Target deflection angle. We compare the attack results under different target deflection angles in Table 2. The attack results are evaluated by the average accumulated reward R_{acc} of five random split data sets, abbreviated as $\overline{R}_{acc}$. Except the attack targets are different, all other settings in these tests are the same, including the federated learning settings, random splitted datasets, AR-UCB strategy, and attack start rounds. Specifically, our candidate attack label region is between (0,1). The main purpose and the significant attack effect is to flip the labels into the target directions. From Table 1, negative targets achieve higher $\overline{R}_{acc}$ than positive target deflections. Thus it is reasonable to set the negative target when the candidate attack label region is in the positive regions. Among all negative targets, target (-0.4) performs better attack results with $\overline{R}_{acc}$ 0.1913. While $\overline{R}_{acc}$ of other negative targets are all far away from the attack target with less $\overline{R}_{acc}$. Therefore, by comparing with multiple target deflection values, target (-0.4) can realize the best attack effect when the target label region is between (0,1). That is also the reason to choose target (-0.4) as the target deflection angle when the candidate label region is set as (0,1) in our experiment.

4.6.4.3. Attack round. We compare the attack effect when the attack starts from different rounds during the federated learning process. Fig. 14 shows the attack results when the attack starts from the 30th, 60th, 100th, and 200th rounds separately. Under the same attack target, when the attack is implemented from the 30th round in the ATT_FLAV, the average accumulated rewards grow to 0.2 in less than 100 round attacks. In comparison, as the attack is exerted in the later rounds, the average accumulated rewards of the attack increase slightly. Attack from the 60th round takes around 150 rounds to achieve its best attack rewards and maintain the attack reward around 0.15 after 200 rounds. The average accumulated reward of the attack climbs to 0.1 from the 100th round to the 200th round and keeps around more than 0.1 after the 300th round of federated learning. When the attack is added in the 200th round of federated learning, the attack is harder to be accomplished. The attack reward raises towards 0.1 slowly during the following 300 rounds' attack. In general, an attack from the 30th round can reach and maintain the best average accumulated reward in the least federated learning rounds. As analyzed in Section 4.6.1, the FLAV model converges after 200 rounds. That is to say, the attack can obtain and maintain more attack rewards when starting from the earlier federated learning rounds, especially before the model convergence, than attacking after model convergence in ATT_FLAV. Thus, when the model is updated based on data with new distributions, it is more likely to be poisoning attacked.

4.6.4.4. Robustness evaluation. The aggregation scheme on the server end can counteract the uploaded poisoning attacks to some extent. This section compares the attack ability of the proposed attack strategy against classical defense schemes FedAvg, Krum and Trimmed Mean in federated learning. Fig. 15 exhibits comparison results of average accumulated rewards between the FedAvg-(AR-UCB), Krum-(AR-UCB) and TrimmedMean-(AR-UCB). FedAvg-(AR-UCB) is the base aggregation used on the server end in our experiments. Krum-(AR-UCB), and TrimmedMean-(AR-UCB) replace the FedAvg with Krum and Trimmed Mean as the defense aggregation scheme on the server ends separately. Note, higher Average Accumulated Rewards mean more attack influence survives. It is obvious that the average accumulated attack reward of

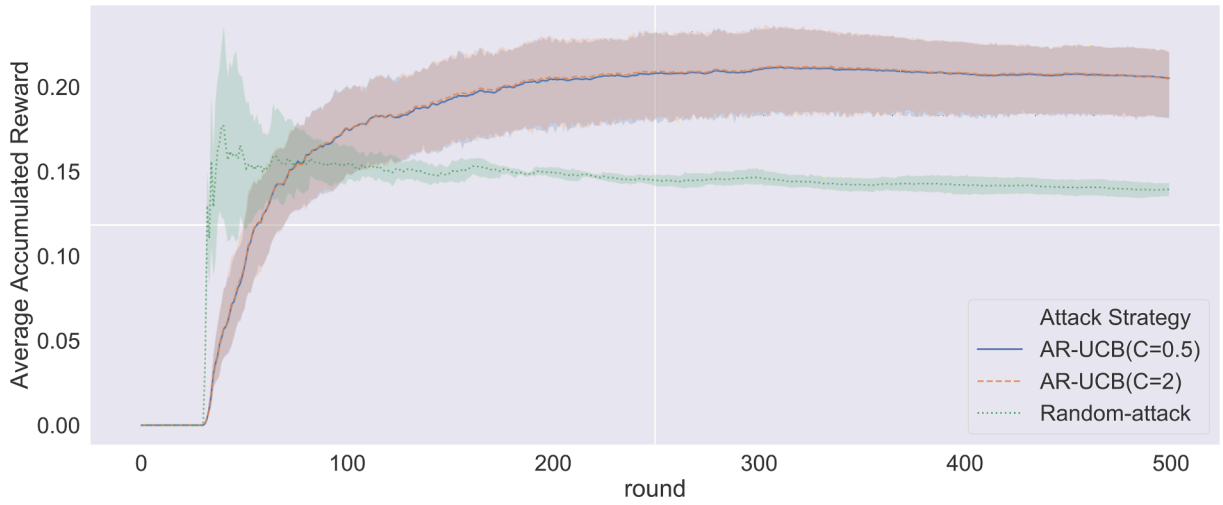


Fig. 13. The average accumulated reward of different C in AR-UCB.

Table 2
Average R_{acc} under different target deflection.

Target	−0.6	−0.5	−0.4	−0.3	−0.2	0.2	0.3	0.4	0.5	0.6
$\overline{R}_{acc}$	0.1189	0.1187	0.1913	0.1170	0.1167	0.1168	0.1161	0.1189	0.1186	0.1178

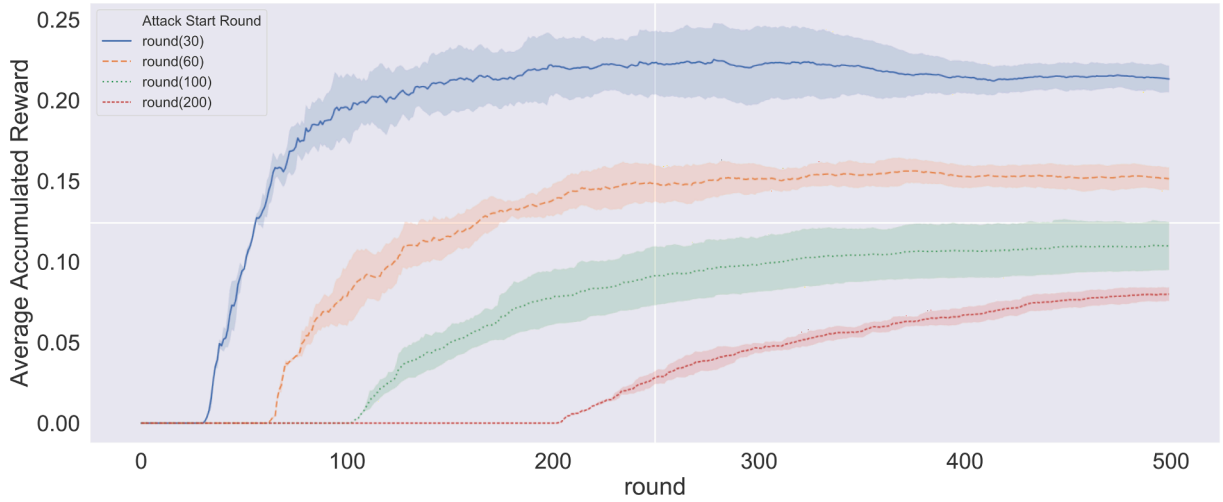


Fig. 14. The average accumulated rewards of attacking from different rounds.

Krum-(AR-UCB) is higher than FedAvg-(AR-UCB), and TrimmedMean-(AR-UCB) from the beginning to the end of the attack. As the attack is carried out, the average accumulated reward of Krum-(AR-UCB) increases more rapidly than FedAvg-(AR-UCB) and TrimmedMean-(AR-UCB).

The average accumulated reward of the Krum-(AR-UCB) keeps increasing from the 30th round to the 500th round, while the FedAvg-(AR-UCB) increases from the 30th round to the 200th round and keeps at 0.2 in the following rounds. Krum-(AR-UCB) achieves 0.4 attack reward which is almost 0.2 more rewards than FedAvg-(AR-UCB) at the 500th round. The average accumulated reward of TrimmedMean-(AR-UCB) also keeps growing from the beginning of the attack but the growing speed is a little slower than Krum-(AR-UCB) and faster than FedAvg-(AR-UCB). At the 500th attack round, the average accumulated reward of TrimmedMean-(AR-UCB) is near but less than 0.4 which is lower than Krum-(AR-UCB) and 0.2 higher than FedAvg-(AR-UCB). Even the lowest point of Krum-(AR-UCB) and TrimmedMean-(AR-UCB) is higher than that in FedAvg-(AR-UCB) after 200 rounds. In other words, the Krum and Trimmed Mean discount less attack effect of the proposed attack strategy than the commonly used FedAvg aggregation scheme. AR-UCB can bypass more defense effects from Krum than Trimmed Mean.

From the view of the aggregation principle of FedAvg, Krum, and Trimmed Mean, FedAvg takes the average weight of all collected client models while Krum and Trimmed Mean select parts of client models for aggregation based on the model distances. Thus, FedAvg definitely discounts the malicious client models. However, since the target label region is selected dynamically in our proposed attack strategy, Krum may preserve the malicious client model for aggregation in some rounds when the distance of the malicious model is closer than the honest one and other client models. Similarly, Trimmed Mean takes the mean of trimmed remaining models as the updated global model. Therefore, when the overall parameters of the malicious model are not the farthest from other models, the malicious model can be maintained and

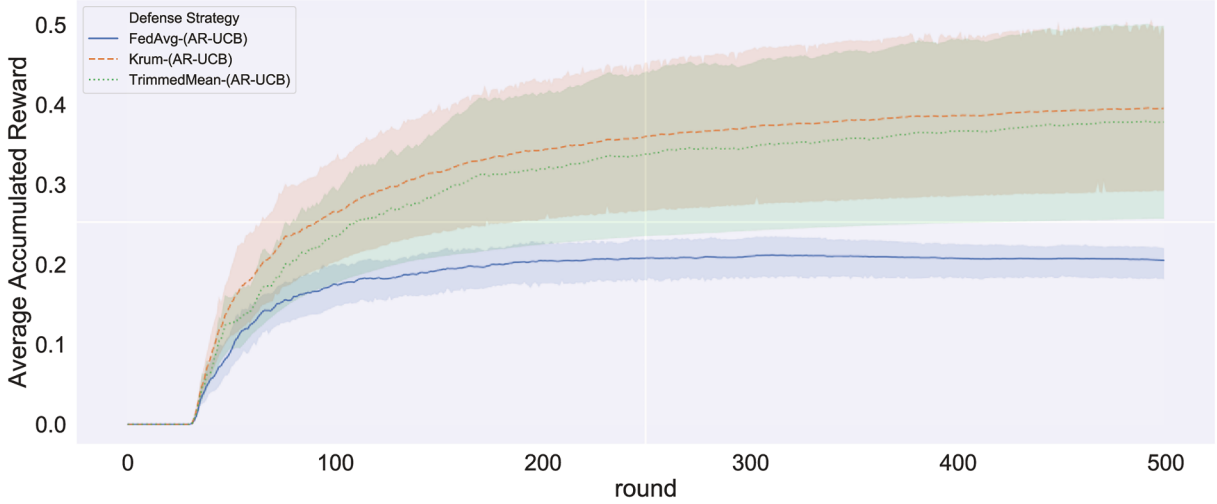


Fig. 15. Average accumulated reward under different defense schemes against the poisoning attack.

aggregated to the global model. Even though the malicious parameters will be partly merged with honest model parameters, the remaining poison can affect the global model updates. Overall, the proposed AR-UCB attack can bypass defense from the three classical defenses, indicating that the proposed attack is robust under classical defense schemes. Hence, it is necessary to develop stronger defense schemas for protecting the training safety of the federated learning framework ahead of time.

5. Conclusion

This paper proposes an attack framework, named ATT_FLAV, for exploring the vulnerability of the autonomous driving intelligent model trained by federated learning. In our work, the autonomous steering angle control model is regarded as the representative use case of the federated learning based non-linear regression learning task. As a result, defense and reinforcement measures can be deployed to the federated learning based autonomous driving framework to enhance its robustness of the framework. Moreover, considering several non-linear regression models used in real-world autonomous driving intelligent decision tasks, a low attack cost data poisoning attack algorithm, **AttackRegion-UCB (AR-UCB)**, used in the ATT_FLAV framework is proposed. The proposed label-flipping attack algorithm provides a novel dynamic black-box target attack for the non-linear regression model updated by federated learning. **AR-UCB** is designed for maximizing the attack reward in continuous updating rounds with limited data and model output information in each round. Based on these contributions, the proposed attack is verified effective with sequential better attack rewards and can defend the classical aggregation schemes.

There are also several limitations existing in our proposed attack method. Firstly, due to the particularity of the non-linear regression model different from linear regression and classification model, AR-UCB still affects a little on non-target label regions. This problem will be further studied to maximize the attack effect in the target region and maintain the model performance on non-target label regions as much as possible. Moreover, the attack still may be defended by other stronger aggregation defense methods proposed in the future. The strong aggregation defense deployed on the federated learning server end usually can detect abnormal client models and obstruct continuous attacks from minority malicious client ends. Thus, the persistence, robustness, and stealth of the low-cost attack still need to be improved as new defense methods come out.

In a word, our method paves the way for studying the model vulnerability of federated learning based non-linear regression model and draws the research attention to the robustness enhancement of the online learning model in real-world applications. Autonomous driving model robustness needs more attention for driving safety. Autonomous driving based on federated learning is vulnerable to the poisoning attack. More researches on the physical world attacks are necessary. Moreover, in the physical world, improving the attack efficiency and the robustness of the attack meets more challenges that need to be resolved.

CRediT authorship contribution statement

Shuo Wang: Conceptualization, Methodology, Software, Validation, Investigation, Writing – original draft, Visualization. **Qianmu Li:** Formal analysis, Supervision, Project administration, Funding acquisition. **Zhiyong Cui:** Conceptualization, Methodology, Data curation, Writing – review & editing, Funding acquisition. **Jun Hou:** Resources, Supervision. **Chanying Huang:** Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data we used is public and we have shared link to the data source in the mansucript

Acknowledgments

This work is supported by the National Key R&D Program of China under Grant [2020YFB1805503]; the Youth Talent Support Program of Beihang University [KG16166701]; the SKL-IoTSC (UM) Open Research Project [SKL-IoTSC(UM)-2021-2023/ORP/GA08/2022]; the National Natural Science Foundation of China [52202378]; the 2022 Jiangsu Province Major Project of Philosophy and Social Science Research in Colleges and Universities “Research on the Construction of Ideological and Political Selective Compulsory Courses in Higher Vocational Colleges” [2022SJZDSZ011]; the Research Project of Nanjing Vocational University of Industry Technology [2020SKYJ03]; the Fundamental Research Fund for the Central Universities [30920041112]; the Fundamental Research Fund for the Central Universities [30920041112].

References

Biggio, B., Nelson, B., & Laskov, P. (2011). Support vector machines under adversarial label noise. *Journal of Machine Learning Research, 20*, 97–112.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., & Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in Neural Information Processing Systems, 2017-Decem*, 119–129.

Chi, L., & Mu, Y. (2017). Learning end-to-end autonomous steering model from spatial and temporal visual cues. *VSCC 2017 - Proceedings of the Workshop on Visual Analysis in Smart and Connected Communities, Co-Located with MM 2017*, 9–16. 10.1145/3132734.3132737.

Chulin Xie, Keli Huang, Pin-Yu Chen, B. L. (2020). Dba : Distributed Backdoor Attacks. *8th International Conference on Learning Representations, {ICLR} 2020*, 1–15. https://openreview.net/pdf?id=rkgyS0VFvr.

Deng, Y., Zheng, X., Zhang, T., Chen, C., Lou, G., & Kim, M. (2020). An Analysis of Adversarial Attacks and Defenses on Autonomous Driving Models. *18th Annual IEEE International Conference on Pervasive Computing and Communications, PerCom 2020*. 10.1109/PerCom45495.2020.9127389.

Elbir, A. M., Soner, B., & Coleri, S. (2020). *Federated Learning in Vehicular Networks*. http://arxiv.org/abs/2006.01412.

Feng, S., Yan, X., Sun, H., Feng, Y., & Liu, H. X. (2021). Intelligent driving intelligence test for autonomous vehicles with naturalistic and adversarial environment. *Nature Communications, 12*(1). https://doi.org/10.1038/s41467-021-21007-8

Fernando, T., Denman, S., Sridharan, S., & Fookes, C. (2017). Going deeper: Autonomous steering with neural memory networks. *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017*, *2018-Janua*, 214–221. 10.1109/ICCVW.2017.34.

Ghafouri, A., Vorobeychik, Y., & Koutsoukos, X. (2018). Adversarial regression for detecting attacks in cyber-physical systems. *IJCAI International Joint Conference on Artificial Intelligence*, *2018-July*, 3769–3775. 10.24963/ijcai.2018/524.

Gidado, U. M., Chiroma, H., Aljojo, N., Abubakar, S., Popoola, S. I., & Al-Garadi, M. A. (2020). A survey on deep learning for steering angle prediction in autonomous vehicles. *IEEE Access, 8*, 163797–163817. https://doi.org/10.1109/ACCESS.2020.3017883

Guan, Z., Ji, K., Bucci, D. J., Hu, T. Y., Palombo, J., Liston, M., & Liang, Y. (2020). Robust stochastic bandit algorithms under probabilistic unbounded adversarial attack. *AAAI 2020 - 34th AAAI Conference on Artificial Intelligence*, *34*(04), 4036–4043. 10.1609/aaai.v34i04.5821.

Gupta, K., Pesquet-Popescu, B., Kaakai, F., Pesquet, J. C., & Malliaros, F. D. (2021). An adversarial attacker for neural networks in regression problems. *CEUR Workshop Proceedings*, *2916*.

Huang, A. (2020). *Dynamic backdoor attacks against federated learning*. http://arxiv.org/abs/2011.07429.

Ilyas, A., Engstrom, L., & Madry, A. (2019). Prior convictions: Black-box adversarial attacks with bandits and priors. *7th International Conference on Learning Representations, ICLR 2019*. https://git.io/fAjOJ.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., & Li, B. (2018). Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning. *Proceedings - IEEE Symposium on Security and Privacy*, *2018-May*, 19–35. 10.1109/SP.2018.00057.

Jagielski, M., Severi, G., Pousette Harger, N., & Oprea, A. (2021). Subpopulation Data Poisoning Attacks. *Proceedings of the ACM Conference on Computer and Communications Security*, 3104–3122. 10.1145/3460120.3485368.

Jiang, W., Li, H., Liu, S., Ren, Y., & He, M. (2019). A Flexible Poisoning Attack Against Machine Learning. *IEEE International Conference on Communications, 2019-May*. 10.1109/ICC.2019.8761422.

Kim, K., Kim, J. S., Jeong, S., Park, J. H., & Kim, H. K. (2021). Cybersecurity for autonomous vehicles: Review of attacks and defense. *Computers and Security, 103*. https://doi.org/10.1016/j.cose.2020.102150

Kong, Z., Guo, J., Li, A., & Liu, C. (2020). PhysGAN: Generating Physical-World-Resilient Adversarial Examples for Autonomous Driving. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 14242–14251. 10.1109/CVPR42600.2020.01426.

Kuleshov, V., & Precup, D. (2014). Algorithms for multi-armed bandit problems. *Journal of Machine Learning Research, 1*, 1–48. 10.48550/arxiv.1402.6028.

Lattimore, T., & Szepesvári, C. (2020). Bandit Algorithms. In *Cambridge University Press*. 10.1017/9781108571401.

Li, X., Kesidis, G., Miller, D. J., & Lucic, V. (2021). *Backdoor Attack and Defense for Deep Regression*. http://arxiv.org/abs/2109.02381.

Liu, X., Si, S., Zhu, X., Li, Y., & Hsieh, C. J. (2019). A unified framework for data poisoning attack to graph-based semi-supervised learning. *Advances in Neural Information Processing Systems, 32*.

Lyu, L., Yu, H., Zhao, J., & Yang, Q. (2020). Threats to Federated Learning. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12500 LNCS*, 3–16. 10.1007/978-3-030-63076-8_1.

M P, A., R, G., & Panda, M. (2021). Steering Angle Prediction for Autonomous Driving using Federated Learning: The Impact of Vehicle-To-Everything Communication. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–7. 10.1109/icccnt51525.2021.9580097.

Ma, Y. (2021). *Adversarial Attacks in Sequential Decision Making and Control*.

Meng, L., Lin, C. T., Jung, T. P., & Wu, D. (2019). White-box target attack for EEG-based BCI regression problems. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-030-36708-4_39

Muller, N., Kowatsch, D., & Bottinger, K. (2020). Data poisoning attacks on regression learning and corresponding defenses. In *Proceedings of IEEE Pacific Rim International Symposium on Dependable Computing*. https://doi.org/10.1109/PRDC50213.2020.00019

Nguyen, A., Do, T., Tran, M., Nguyen, B. X., Duong, C., Phan, T., Tjiputra, E., & Tran, Q. D. (2021). *Deep Federated Learning for Autonomous Driving*. http://arxiv.org/abs/2110.05754.

Paudice, A., Muñoz-González, L., & Lupu, E. C. (2019). Label sanitization against label-flipping poisoning attacks. In *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-030-13453-2_1

Pu, Z., Cui, Z., Tang, J., Wang, S., & Wang, Y. (2022). Multimodal traffic speed monitoring: A real-time system based on passive wi-fi and bluetooth sensing technology. *IEEE Internet of Things Journal, 9*(14), 12413–12424. https://doi.org/10.1109/JIOT.2021.3136031

Pu, Z., Zhu, M., Li, W., Cui, Z., Guo, X., & Wang, Y. (2021). Monitoring public transit ridership flow by passively sensing wi-fi and bluetooth mobile devices. *IEEE Internet of Things Journal, 8*(1), 474–486. https://doi.org/10.1109/JIOT.2020.3007373

Rausch, V., Hansen, A., Solowjow, E., Liu, C., Kreuzer, E., & Hedrick, J. K. (2017). Learning a deep neural net policy for end-to-end control of autonomous vehicles. *Proceedings of the American Control Conference, 4914–4919*. https://doi.org/10.23919/ACC.2017.7963716

Savazzi, S., Nicoli, M., Bennis, M., Kianoush, S., & Barbieri, L. (2021). Opportunities of federated learning in connected, cooperative, and automated industrial systems. *IEEE Communications Magazine, 59*(2), 16–21. https://doi.org/10.1109/MCOM.001.2000200

Sun, G., Cong, Y., Dong, J., Wang, Q., Lyu, L., & Liu, J. (2021). Data poisoning attacks on federated machine learning. *IEEE Internet of Things Journal, 1–1*. https://doi.org/10.1109/jiot.2021.3128646

Sundar, A. P., Li, F., Zou, X., Hu, Q., & Gao, T. (2020). Multi-armed-bandit-based shilling attack on collaborative filtering recommender systems. In *Proceedings - 2020 IEEE 17th International Conference on Mobile Ad Hoc and Smart Systems*. https://doi.org/10.1109/MASS50613.2020.00050

Taheri, R., Javidan, R., Shojafar, M., Pooranian, Z., Miri, A., & Conti, M. (2020). On defending against label-flipping attacks on malware detection systems. *Neural Computing and Applications, 32*(18), 14781–14800. https://doi.org/10.1007/s00521-020-04831-9

Wang, S., Pu, Z., Li, Q., & Wang, Y. (2022). Estimating crowd density with edge intelligence based on lightweight convolutional neural networks. *Expert Systems with Applications, 206*, Article 117823. https://doi.org/10.1016/J.ESWA.2022.117823

Xiao, H., Xiao, H., & Eckert, C. (2012). Adversarial label flips attack on support vector machines. *Frontiers in Artificial Intelligence and Applications, 242*, 870–875. https://doi.org/10.3233/978-1-61499-098-7-870

Xiao, Huang, Biggio, B., Brown, G., Fumera, G., Eckert, C., & Roli, F. (2018). Is feature selection secure against training data poisoning? *32nd International Conference on Machine Learning, ICML 2015*, *2*, 1689–1698. https://arxiv.org/abs/1804.07933v1.

Yin, D., Chen, Y., Ramchandran, K., & Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *35th International Conference on Machine Learning, ICML 2018* (Vol. 13, pp. 8947–8956). PMLR. https://proceedings.mlr.press/v80/yin18a.html.

Zhang, Hongyi, Bosch, J., & Olsson, H. H. (2021). End-to-End Federated Learning for Autonomous Driving Vehicles. *Proceedings of the International Joint Conference on Neural Networks, 2021-July*. 10.1109/IJCNN52387.2021.9533808.

Zhang, X., Ma, Y., Singla, A., & Zhu, X. (2020). Adaptive reward-poisoning attacks against reinforcement learning. *37th International Conference on Machine Learning, ICML 2020*, *PartF16814*, 11161–11170.

Zhang, Hongpo., Cheng, N., Zhang, Y., & Li, Z. (2021). Label-flipping attacks against Naive Bayes on spam filtering systems. *Applied Intelligence, 51*(7), 4503–4514. https://doi.org/10.1007/s10489-020-02086-4

Zhuang, Y., Pu, Z., Hu, J., & Wang, Y. (2022a). Illumination and temperature-aware multispectral networks for edge-computing-enabled pedestrian detection. *IEEE Transactions on Network Science and Engineering, 9*(3), 1282–1295. https://doi.org/10.1109/TNSE.2021.3139335

Zhuang, Y., Pu, Z., Yang, H. F., & Wang, Y. (2022b). Edge-artificial intelligence-powered parking surveillance with quantized neural networks. *IEEE Intelligent Transportation Systems Magazine*. https://doi.org/10.1109/MITS.2022.3182358