EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

Model Protection Scheme Against Distillation Attack in Internet of Vehicles

Weiping Peng[1], Jiabao Liu[1,*], Yuan Ping[2] and Di Ma[1]

[1]School of Computer Science and Technology, Henan Polytechnic University, No. 2001, Century Road, Jiaozuo 454003, China
[2]School of Information Engineering, Xuchang University, No. 88, Bayi Road, Xuchang 461000, China

Abstract

Aiming at the problems of model security and user data disclosure caused by the deep learning model in the Internet of Vehicles scenario, which can be stolen by malicious roadside units or base stations and other attackers through knowledge distillation and other techniques, this paper proposes a scheme to strengthen prevent against distillation. The scheme exploits the idea of model reinforcement such as model self-learning and attention mechanism to maximize the difference between the pre-trained model and the normal model without sacrificing performance. It also combines local differential privacy technology to reduce the effectiveness of model inversion attacks. Our experimental results on several datasets show that this method is effective for both standard and data-free knowledge distillation, and provides better model protection than passive defense.

Keywords: Internet of vehicles, Privacy protection, Distillation immunity, Model reinforcement, Differential privacy

Received on 07 May 2023, accepted on 21 May 2023, published on 27 June 2023

doi: 10.4108/eetel.v8i3.3318

1. Introduction

The Internet of Vehicles (IoV) is a technology that uses artificial intelligence and 5G communications to achieve intelligent traffic management and vehicle control through multidimensional interactions between vehicles and other vehicles, vehicles and people, and vehicles and the road environment. The goal of IoV is to provide a safe, comfortable and efficient driving experience and transport services. In the IoV system, vehicles are equipped with devices that have data collection, processing and storage capabilities. These devices generate a large amount of network data, such as vehicle speed, orientation, road information and traffic conditions. This data supports the development of various technologies and applications, including traffic flow prediction, vehicle trajectory prediction, pedestrian collision detection, high-precision in-vehicle navigation, and in-vehicle entertainment. Deep learning provides a new solution for efficiently fusing and processing this data and information [1].

In the IoV scenario, it takes a lot of effort and resources for companies to train advanced deep learning models for vehicles. These trained models and proprietary training data have high intellectual property rights, making it legally and ethically prohibited to share them publicly. However, during the process of information exchange between vehicles and external nodes, attackers can use knowledge distillation [2] techniques to mimic the input and output behavior of the black box, and thereby steal vehicle deep learning models. In addition, data-free knowledge distillation combined with adversarial network attack generation, membership inference, model inversion, and other reverse engineering methods can enable the recovery of private training data from black box models[3-6], seriously undermining the privacy of IoV users.

Scholars around the world have conducted a lot of research on this topic and achieved a number of results. For example, Reference [7] combined federated learning and local differential privacy to propose the LDP-FedSGD algorithm to coordinate cloud servers and vehicles to collaboratively train models, which significantly reduces the risk of data leakage while considering practicality.

Reference [8] proposed a hybrid blockchain architecture consisting of a permissioned blockchain and a local directed acyclic graph to reduce the transmission load and address the privacy concerns of providers. The reliability of the shared data can also be ensured by integrating the learned model into the blockchain and performing a two-step verification. Liu et al. [9] proposed a hybrid proxy authentication scheme by introducing the concept of proxy vehicles and integrating hybrid authentication based on identity and PKI, which ensures the data security of IoV users while improving the effectiveness of roadside units in terms of authentication messages. The above solutions have provided some security protection for different application in IoV; however, all of them only consider the traditional data leakage problem and ignore the possibility of theft of vehicle deep learning models. Recent work relies on watermark-based [10] or passport-based [11] authentication methods to protect models. However, they can only detect model attribution and are ineffective in avoiding model cloning. The above defense methods are all reactive and have not explored knowledge distillation-based model attacks in the IoV environment, which is a problem worth investigating.

To address the aforementioned limitations, this paper proposes a defense scheme, called Strengthen Prevent Distillation (SPD), for the three-layer architecture of the cloud-side-end of the IoV [12]. The scheme constructs a deep learning model of the vehicle as a specially trained network that performs similarly to the corresponding normal model, but renders inversion of the model by an attacker through methods such as knowledge distillation ineffective. Our main contributions are summarized as follows:

- First, we have summarized the defense methods for model and data theft in the context of the IoV and analyzed their pros and cons.
- Next, we creatively embedded the ideas of attention mechanism and local differential privacy into the method for defending against knowledge distillation attacks. Simulation experiments have verified the effectiveness and rationality of the SPD algorithm.
- Finally, we conducted extensive comparative experiments to verify the superiority of our method over other traditional methods, demonstrated its terrific performance in the absence of data distillation, and identified the effectiveness of our method through qualitative analysis.

The remainder of this paper is organized as follows. In section 2, we introduced the basic principles of knowledge distillation and the foundations of model reinforcement and differential privacy, and reviewed some of the previous contributions. In Section 3, we presented the overall architecture of the IoV and provided a detailed discussion of our proposed method. The simulation results were presented in Section 4 to demonstrate the effectiveness of our proposed mechanism. Finally, Section 5 summarised the work done in this paper and outlined future research directions.

2. Related Work

2.1. Image Classification

Image classification has extensive applications in various fields, such as computer vision, natural language processing, intelligent transportation, and medical image analysis. By selecting appropriate feature extraction methods and classification algorithms, high-precision image classification tasks can be achieved. With the rising number of vehicles on urban roads, Intelligent Transportation Systems (ITS) play a vital role in enhancing traffic flow and efficiency while minimizing accidents. The vast amount of data generated by various digital devices connected to the transportation network facilitates the creation of datasets, which can be analyzed using advanced deep learning techniques. This approach helps in predicting traffic performance, automating traffic signal management, detecting lanes, and recognizing objects in close proximity to vehicles, thereby improving the safety and efficacy of ITS [13]. Wang et al. introduced Particle Swarm Optimization to construct a PSO-guided Self-Tuning Convolution Neural Network (PSTCNN), enabling the model to automatically adjust hyperparameters and allowing deep learning models to more quickly and accurately diagnose COVID-19, effectively alleviating the problem of global healthcare resource scarcity [14]. The effectiveness of artificial intelligence technology in diagnosing COVID-19 and the superiority of Adaptive Jaya algorithm over Jaya algorithm in medical image classification tasks were demonstrated in Reference [15].

2.2. Knowledge Distillation

Knowledge distillation is a widely used method for model compression and optimisation in deep learning. It is based on the concept of a "teacher-student model" for training and is highly regarded for its simplicity and effectiveness. Knowledge Distillation facilitates the training of student models by extracting "knowledge" from one or more pre-trained teacher models using the soft-label probabilistic output of the teacher models. This soft-label output is a mapping from input vectors to output vectors that captures specific knowledge from instantiated objects, with incorrect classification predictions providing insight into how the teacher model generalizes. The student model can improve its performance by mimicking the probabilistic output of the teacher model, and can incorporate the knowledge that the teacher model has already acquired. The process of knowledge distillation is illustrated in Fig. 1.

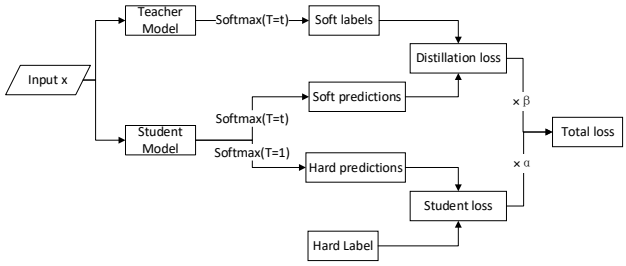

Figure 1. Flow chart of knowledge distillation. Teacher networks can transfer their model capabilities to student networks through knowledge distillation.

As shown in Eq. (1), neural networks typically generate class probabilities by using a "softmax" output layer that compares the output z_i of each class with other logits, converting the logit z_i calculated for each class into a probability q_i in a standardized way. In addition, where T represents temperature in knowledge distillation, using a larger value than 1 for it produces a softer class probability distribution that allows better transfer of knowledge to the model to be distilled.

$$q_i = \frac{\exp(z_i / T)}{\sum_j \exp(z_j / T)} \tag{1}$$

Given a pre-trained teacher model $f_{\theta_T}(\cdot)$ and a student model $f_{\theta_S}(\cdot)$, where θ_T and θ_S denote the model parameters. Knowledge distillation aims to make the output probability of $f_{\theta_S}(\cdot)$ as close to $f_{\theta_T}(\cdot)$ as possible. Let (x_i, y_i) denote the training sample in datasets χ and $p_{f_\theta}(x_i)$ denote the logit response of x_i to $f_\theta(\cdot)$. The student model $f_{\theta_S}(\cdot)$ can be learned by means of as in Eq. (2):

$$\begin{aligned} Tea &= \alpha \tau_S^2 KL\left(\sigma_{\tau_S}\left(p_{f_{\theta_T}}(x_i)\right), \sigma_{\tau_S}\left(p_{f_{\theta_S}}(x_i)\right)\right) \\ Stu &= (1-\alpha)\chi\varepsilon\left(\sigma\left(p_{f_{\theta_S}}(x_i)\right), y_i\right) \\ L &= \min\left(Tea + Stu\right) \end{aligned} \tag{2}$$

Where $KL(\cdot,\cdot)$ and $\chi\varepsilon(\cdot,\cdot)$ represent the *K-L* divergence and cross-entropy loss functions, respectively. The introduced "softmax temperature" function $\sigma_{\tau_S}(\cdot)$ produces a soft probability output when a larger temperature τ_S is selected, decays to a normal softmax function $\sigma(\cdot)$ equal to 1, and another hyperparameter α to balance the cost minimization of knowledge distillation.

2.3. Prevent Knowledge Distillation

Combining knowledge distillation with methods such as generative adversarial networks can lead to the theft of deep learning models and user privacy. Ma et al. [16] proposed a special deep learning model with slightly worse performance compared to its normal counterpart, both with the ability of classification and regression in deep learning, from which no malicious third-party network can extract useful parameters using knowledge distillation. The algorithm is implemented by maintaining its correct category assignment and disrupting its incorrect category assignment as much as possible to prevent attackers from stealing model information and raw data through distillation. The process of constructing the prevent distillation model is shown in Eq. (3):

$$\begin{aligned} nor &= \chi\varepsilon(\sigma((p_{f_{\theta_T}}(x_i)), y_i) \\ dis &= \omega\tau_A^2 KL(\sigma_{\tau_A}(p_{f_{\theta_T}}(x_i)), \sigma_{\tau_A}(p_{f_{\theta_A}}(x_i))) \\ L &= \min\left(nor - dis\right) \end{aligned} \tag{3}$$

The first part of Eq. (3) aims to maintain the accuracy of the model by minimizing the cross-entropy loss, while the second part maximizes the K-L divergence between the pre-trained model and a regular network to hide the "useful knowledge" and achieve "prevent-distillation". In this equation, τ_A represents the temperature of self-sabotage, and ω balances the weight of the loss function accounted for by both normal training and adversarial learning.

2.4. Model Enhancement

Zhang et al. [17] proposed the concept of self-distillation by closing the gap between the deep and shallow modules of the model without the help of an external model, which improves the overall accuracy of the model. Chen et al. [18] proposed a knowledge review approach to improving the performance of the student model by packaging the knowledge of the shallow modules of the teacher model and imparting it to the student model. Hou et al. [19] proposed a self-distillation-based lane line detection algorithm that utilizes the concept of an intermediate layer attention map, where each layer receives attention-guided training from the last layer to improve the performance of the lane line detection model by passing features from the deeper layers of the model to the shallower layers in advance for learning. Vaswani et al. [20] creatively proposed a simple network architecture based on an attention mechanism that reduces the training time while optimizing the model. Hu et al. [21] automatically obtain the importance of each channel by explicitly modelling the interdependencies between the feature channels, then boost the useful features and suppress the features that are less useful for the current task according to their importance,

and finally, improve the overall performance of the network. These methods are validated on public datasets and provide good ideas for the self-optimization of the model.

2.5. Differential Privacy

Differential privacy is a cryptographic technique that aims to maximize the accuracy of data queries and reduce the chance of identifying records from statistical database queries and is widely used in deep learning models to protect data privacy [22]. Local differential privacy (LDP) is one of these models, which does not have any trusted third party and needs to add perturbations to its data before sharing it with other data parties. Arachchige et al. [23] proposed the LATENT algorithm, which redesigned the training process and added a randomization layer at this stage before the data leaves the device and reaches the server, significantly improving the utility of differential privacy in the deep learning process. Using the concept of LDP, Wei et al. [24] proposed a user-level differential privacy algorithm that adds artificial noise to the shared model before uploading it to the server and derived a theoretical convergence upper bound for the framework.

The mechanism of differential privacy with parameters (ε,δ) provides a strong criterion for privacy preservation in distributed data processing systems. For example, v and v' are real data sets for two users, given a perturbation algorithm S with output y'. S satisfies localized differential privacy if the probability of obtaining any y' on both v and v' under the action of S satisfies the inequality shown in Eq. (4).

$$\Pr[S(v) \in y'] \le e^{\varepsilon} \times \Pr[S(v') \in y'] + \delta \qquad (4)$$

Where ε is the privacy budget, which indicates the distinguishable boundary between two adjacent datasets, it takes a value greater than 0, and a smaller value indicates a higher level of data protection. $\delta(\delta \in (0,1])$ is the privacy leakage probability.

Knowledge distillation-based attack and defense in the IoV environment covers many aspects. This section reviews the ways of knowledge distillation, the basic practices of prevent distillation, and briefly introduces the classical practices of model reinforcement and how to add local differential privacy noise to deep learning models. These contents laid the foundation for the subsequent methods proposed.

3. Methodology

3.1. System Architecture of IoV

The IoV system architecture is shown in Fig. 2 and consists of 3 parts: vehicle, roadside unit(RSU), and base station(BS). It is assumed that a single base station can cover all the areas shown in Fig. 2 and provide remote communication services to initialize the whole IoV application system and generate system-related parameters. Three roadside units are deployed near each road section, connecting upwards to the base station and downward to the vehicles on the road via wired or wireless channel communication links to provide authentication and real-time data services to the vehicles. In terms of computing and communication capabilities, the base station is more powerful and the roadside unit is weaker [25]. The vehicle is equipped with an intelligent vehicle system that communicates with roadside units and base stations in real-time and can select the appropriate roadside unit for authentication and information interaction according to its area and handle complex and changing road information, to ensure that the vehicle can be safely exercised on the road. If a vehicle is not within the coverage area of any roadside unit, it interacts directly with the base station for information.

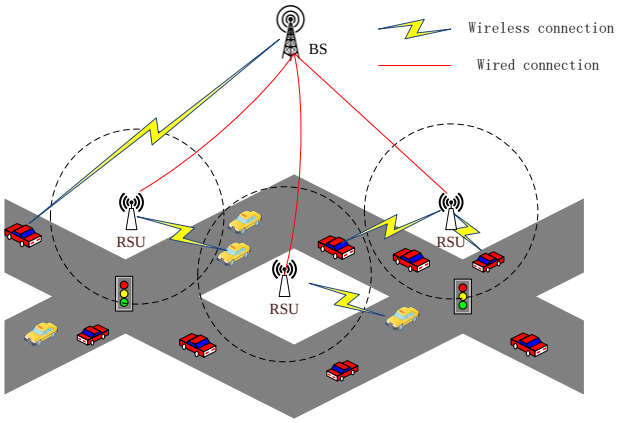

Figure 2. The IoV system architecture. The vehicles on the road communicate information with base stations or roadside unit nodes through wired and wireless connections.

Table 1. The IoV system parameters

Parameter	Meaning
v_j^i	The i-th vehicle within the j-th roadside unit
r_n^j	The j-th roadside unit within the range of the n-th base station
d_j^i	Number of samples contained in the orresponding vehicle data set
b_n	The n-th base station
$w_j^i(t)$	Model parameters of vehicle v_j^i in the t-th iteration
$L_j^i(w)$	Loss function during vehicle training
$L(w(t))$	Global aggregation loss function

$\lvert R_J \rvert$	Number of roadside units in the area to which the base station belongs
$\lvert V_J^I \rvert$	Number of vehicles in the area to which the base station belongs

The system model proposed in this paper consists of a master chain computation process consisting of the base station and the roadside units within its coverage area, a slave chain computation process consisting of the roadside units and the vehicles within its coverage area, and a local computation process of the vehicle deep learning model. In this paper, we assume that some of the nodes can be exploited by attackers to eavesdrop on the output information of the vehicle's deep learning model, and that there is no slack state in the nodes themselves and no possibility of malicious uploading of incorrect parameters. The relevant parameters are listed in Table 1.

The vehicle performs a deep learning-based model training process locally and, after a while, sends the training results up to the adjacent roadside units. In the local training phase, each vehicle is trained with a deep learning model based on the local dataset, and the loss function of vehicle v_j^i on the training dataset d_j^i is shown in Eq. (5).

$$L_j^i(w) = \frac{1}{\left|d_j^i\right|} \sum_{u \in \left|d_j^i\right|} l_u(w, x_u, y_u) \tag{5}$$

Where $l_u(w, x_u, y_u)$ is the value of the loss function on the data samples (x_u, y_u), and w is the parameter of the trained model. In different algorithms, the loss function is calculated in different ways. In this paper, the most common gradient descent method is used to construct the loss function and thus update the values of the weight parameters, as shown in Eq. (6).

$$w_j^i(t) = w_j^i(t-1) - \eta \nabla L_j^i(w_j^i(t-1)) \tag{6}$$

Where $w_j^i(t)$ is the model weight parameter for the t-th iteration, η is the learning rate, and $\eta \nabla L_j^i(w_j^i(t-1))$ is the gradient of the loss function for parameter $w_j^i(t-1)$. After each round of training, these updated weight parameters are uploaded to nearby roadside units via a wireless network.

During the iterative process from the slave chain, the roadside unit receives the model prediction results from all the vehicles involved in the training. It can aggregate these data to minimize the loss function and improve the accuracy of the vehicle deep learning model. The weighted aggregation approach used in this paper is as shown in Eq. (7).

$$w_j(t) = \frac{1}{\sum_{i=1}^{I} \left|d_j^i\right|} \sum_{i=1}^{I} \left|d_j^i\right| w_j^i(t) \tag{7}$$

In the iterative process of the master chain, similar to the learning process of the roadside unit, the base station stores the predictions of the roadside unit locally and simultaneously aggregates all the received parameters globally, where the loss function is defined as Eq. (8).

$$L(w(t)) = \frac{1}{\left|R_J\right|} \frac{1}{\left|V_J^I\right|} \sum_{j \in J} \sum_{i \in I} \sum_{u \in \left|d_j^i\right|} \frac{l_j^i(w_j^i, x_i^u, y_i^u)}{\left|d_j^i\right|} \tag{8}$$

Where $\left|R_J\right|$ and $\left|V_J^I\right|$ represent the number of roadside units and the number of vehicles in the area to which the base station belongs. The training process minimizes the overall loss function along the opposite direction of the gradient $L(w)$.

3.2. Strengthen Prevent Distillation Process

The vehicle deep learning model is denoted by *S*, and the information collected by the vehicle is denoted by *X*, $Y_S = S(X)$ indicates the output of the model logit. Model S can be divided into different parts $(S_1, S_2, \cdots, S_n, S_c)$, where S_c is the classifier. The execution process of the model is shown in Eq. (9), where " $\circ$ " indicates the nesting of functions:

$$Y_S = S_c \circ S_n \circ \cdots \circ S_2 \circ S_1(X) \tag{9}$$

The intermediate layer features of the deep learning model are $(F_S^1, F_S^2, \cdots, F_S^n)$, then the i-th feature is calculated as shown in Eq. (10).

$$F_S^i = S_i \circ \cdots \circ S_2 \circ S_1(X) \tag{10}$$

During the process of training a distillation prevention model using the method described in Eq. (3) on local vehicles in IoV, although the above algorithm can have a protective effect on the model, it inevitably results in a weakening of the model's performance. To remedy these shortcomings, this paper proposes a Strengthen prevent distillation model algorithm that uses the deep part of the model to guide the shallow part based on the algorithm shown in Eq. (3), and its own iteration requires relearning the output of each layer of the model as a way to improve the effectiveness of the model itself and further enhance the degree of self-destructive distillation. Its loss function is calculated as shown in Eq. (11) and Eq. (12).

$$L_R = \sum_{i=1}^{n} \left(\sum_{j=1}^{n-i} D\left(M\left(F_s^i\right), M\left(F_s^{i+j}\right)\right)\right) \tag{11}$$

$$L = L_{CE} + L_R \tag{12}$$

Where M denotes the transformation of the attention and feature maps and D denotes the distance function of the two

parts of the model. In this paper, we use the Attention-based Fusion (ABF) mechanism to adjust the deep high-dimensional features and the shallow low-dimensional features to the same size, and connect the features of different dimensions to generate an attention graph. This attention map is multiplied with the two previous feature maps and finally stitched into the final output. The ABF architecture is shown in Fig. 3.

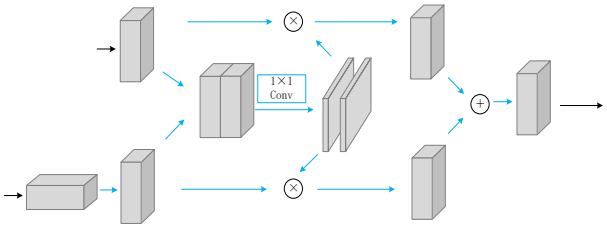

Figure 3. ABF Architecture. The features of different levels in each module are aggregated together through attention maps.

In order to reduce the complexity of model training, the final architecture is progressively improved in this paper. Taking the classic residual network as an example, the specific implementation is that the output of each layer needs to be combined with the output of the later layers to produce a cross-entropy loss function that uses backpropagation to strengthen the model accuracy of the previous layer by layer and to expand the error distribution of the incorrect class until the final output can achieve a prediction accuracy comparable to or better than the initial model. The specific approach is shown in Fig. 4.

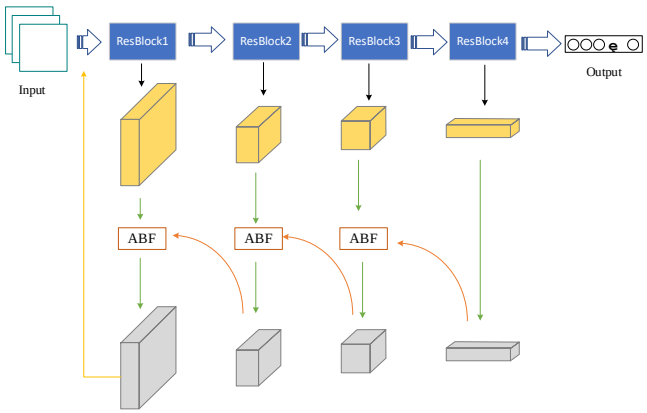

Figure 4. Progressive model architecture. We progressively train the shallow layers of the model to learn from its deep layers, with repeated learning as the final architecture.

3.3. Adding Adaptive Differential Privacy Process

In this paper, a Vehicle Adaptive Differential Privacy (VADP) algorithm is proposed to further prevent malicious attribute inference during the information interaction between the master and slave chains of the connected vehicle system architecture. The algorithm is incorporated into the previous model to further enhance its effectiveness in protecting private data and preventing information leakage during the upload process. This is done by adaptively cropping the gradient according to the change in learning rate during the vehicle's deep learning training process, and by adding a small amount of Gaussian noise to the model's parameters. When all vehicle deep learning models have completed local training, the parameters with noise are uploaded to roadside units or base stations for aggregation. Due to local perturbations, it is difficult for an attacker to infer the private characteristics of a given vehicle and then reconstruct the user's private data by reverse engineering. The algorithm consists of the following components.

step1: Input vehicle initialization model w_0, clipping threshold C and LDP parameters $(\varepsilon_i, \delta_i)$;

step2: Update the local gradient during training, $g_{i,m}^t(d_j^i, m) = \nabla w^t F_i(d_j^i, w^t)$;

step3: Cropping gradient,

$$g_{i,m}^t(d_j^i, m) = g_{i,m}^t(d_j^i, m) / \max(1, \frac{\left\| g_{i,m}^t(d_j^i, m) \right\|_2}{C});$$

step4: Local parameter update,

$$w_i^{t+1} = w_i^t - \frac{1}{|d_j^i|} \sum_{m=1}^{|d_j^i|} \eta g_{i,m}^t(d_j^i, m);$$

step5: Calculating σ_i from LDP parameter $(\varepsilon_i, \delta_i)$ and adding Gaussian noise to the model, $w_i^t = w_i^t + N(0, \sigma_i^2 I)$;

step6: Output the noise-added model and interact the information with the roadside unit.

The process $g_{i,m}^t$ denotes the gradient of the t-th training of the i-th vehicle on the m-th mean dataset, w_i^t denotes the model parameters after noise addition.

4. Experiment

4.1. Experimental Setup

The proposed SPD scheme in this paper first performs self-destructive training according to Eq. (2) to create a distillation-proof model in IoV and performs an strengthen prevent distillation process, and adds adaptive Gaussian noise to optimize itself. To evaluate the effectiveness of the proposed model, we use Eq. (1) to conduct knowledge distillation on a given malicious third-party model and

evaluate the performance of the model. We draw corresponding conclusions from the comparison.

In order to verify the effectiveness of the proposed mechanism in this paper, We used the CIFAR-10, CIFAR-100, SVHN, and Tiny-imagenet datasets. The CIFAR and Tiny-imagenet dataset is used to validate the general applicability of the SPD approach, while the SVHN dataset is used to evaluate the effectiveness of the SPD approach specifically in IoV. CIFAR-10 and CIFAR-100 are often used as classical datasets to test the effectiveness of image classification models, and they both consist of 60,000 32× 32 colour images, of which 50,000 are used as training sets and 10,000 are used as test sets; the difference between them is that CIFAR-10 is used for 10 classification problems, while CIFAR-100 is used for 100 classification problems, and CIFAR-100 is much more difficult to train than the former. The SVHN dataset is extracted from Google Street View images of door numbers and is suitable for in-vehicle sensors reading image data around vehicles in IoV. SVHN contains over 600,000 digital images, including 73,257 images in the training set and 26,032 images in the test set; an additional 531,131 images are also available for training if the model requires a larger amount of data. Tiny-imagenet is derived from the classic dataset ImageNet. It consists of 200 classes, with each class having 500 training images, 50 validation images, and 50 test images, all of which are 32×32 color images. ResNet-18 and ResNet-50 are used as vehicle deep learning models, and ResNet-18, ShufflenetV2, MobilenetV2 and 5-layer normal CNN are used as attacker models as a way to fully evaluate the scheme.

All experiments were conducted on GPU devices under the pytorch 1.11.0 environment. Each network was trained for 100 epochs on two different datasets using the SGD optimizer to optimize the neural network. The initial learning rate was set to 0.1, and it decreased by a factor of 1/10 at 30, 60, and 90 epochs. Other training hyperparameters include weight_decay=5e-4 , momentum=0.9, and a batch size of 128.

In this section, the following comparison scheme is designed for simulation and verification of the algorithm proposed in this paper.

- The prediction accuracies of the vehicle deep learning and attacker models are obtained experimentally and used as a baseline. Comparing the SPD model constructed by the method proposed in this paper with the common vehicle model, it can be seen that the present method hardly affects the prediction accuracy of the model.
- By comparing the distillation of the model constructed by the method proposed in this paper with the distillation of a standard vehicle model, it can be seen that the present method significantly reduces the utility of knowledge distillation, making it meaningless to obtain a vehicle model by means of knowledge distillation.
- By comparing the accuracy in a data distillation-free environment, it is concluded that the SPD scheme can protect the data privacy of users.
- The superiority of the proposed algorithm in this paper is derived by comparing it with the standard resistance distillation algorithm [16] and the adaptive false alarm algorithm [26].

4.2. Experimental Results

The experimental results on CIFAR-10, CIFAR-100, SVHN and Tiny-imagenet are shown in Table 2, Table 3, Table 4 and Table 5 respectively, where the normal model is denoted by NM (Normal) and the enhanced resistance to distillation model is denoted by SPD (Strengthen Prevent Distillation). For ease of presentation, we define the vehicle deep learning model as the teacher network and the attacker model as the student network. To further eliminate chance, 10 simulation experiments are run for each of the above algorithms and the results of each iteration are averaged as the final result.

First, we observed that all SPD models performed similarly to the corresponding normal models. Second, the attacker model steals the normal vehicle model through knowledge distillation, which can improve the accuracy by up to 9.53%. However, distillation of the model proposed in this paper reduces the accuracy by 1.92% to 66.44%, indicating that distillation-prevent vehicle deep learning models can successfully provide a false sense of generalization for malicious roadside units or base station models. In addition, comparing the data in the table shows that weaker attacker networks (e.g. MobilenetV2) may be more vulnerable to errors than stronger networks (e.g. ResNet-18). The published vehicle deep learning models are experimentally "distillation-prevent", so knowledge distillation-based model steganography is no longer be applicable.

Table 2. Experimental results on CIFAR-10

Vehicle models	Model Accuracy	Accuracy of student models after distillation		
		CNN	MobilenetV2	ShufflenetV2
Baseline	-	89.41	81.71	88.32
ResNet18(NM)	94.78	90.98(+1.57)	91.07(+9.36)	92.37(+4.05)
ResNet18(SPD)	94.29(-0.49)	87.19(-2.22)	65.84(-15.87)	82.94(-5.38)
Resnet50(NM)	94.03	90.66(+1.25)	91.24(+9.53)	92.30(+3.98)
Resnet50(SPD)	93.81(-0.22)	87.49(-1.92)	66.05(-15.66)	83.02(-5.30)

Table 3. Experimental results on CIFAR-100

Vehicle models	Model Accuracy	Accuracy of student models after distillation		
		MobilenetV2	ShufflenetV2	Resnet18
Baseline	-	68.46	70.71	77.45
ResNet18(NM)	77.45	73.84(+5.38)	74.30(+3.59)	77.90(+0.45)
ResNet18(SPD)	77.38(-0.07)	2.02(-66.44)	63.69(-7.02)	73.03(-4.42)
Resnet50(NM)	78.01	73.32(+4.86)	74.05(+3.34)	77.95(+0.50)
Resnet50(SPD)	77.97(-0.04)	2.97(-65.49)	62.28(-8.43)	72.67(-4.78)

Table 4. Experimental results on SVHN

Vehicle models	Model Accuracy	Accuracy of student models after distillation		
		MobilenetV2	ShufflenetV2	Resnet18
Baseline	-	89.25	89.69	96.44
ResNet18(NM)	96.44	90.06(+0.81)	92.90(+3.21)	96.89(+0.45)
ResNet18(SPD)	95.98(-0.46)	26.43(-62.82)	85.75(-3.94)	95.32(-1.12)
Resnet50(NM)	95.28	91.12(+1.87)	93.26(+3.57)	96.67(+0.23)
Resnet50(SPD)	94.85(-0.43)	28.31(-60.94)	84.89(-4.80)	95.11(-1.33)

Table 5. Experimental results on Tiny-imagenet

Vehicle models	Model Accuracy	Accuracy of student models after distillation		
		MobilenetV2	ShufflenetV2	Resnet18
Baseline	-	56.17	57.26	63.24
ResNet18(NM)	63.24	59.48(+3.31)	60.31(+3.05)	63.83(+0.59)
ResNet18(SPD)	63.13(-0.11)	49.32(-6.85)	53.42(-3.83)	60.51(-2.73)
Resnet50(NM)	63.98	61.58(+5.41)	61.41(+4.15)	64.27(+1.03)
Resnet50(SPD)	63.86(-0.12)	48.87(-7.30)	50.75(-6.51)	59.98(-3.26)

In order to more clearly see the effectiveness of the algorithm proposed in this paper in preventing knowledge distillation, Fig. 5 visualizes the iterative accuracy of the ordinary model, this paper's model, the attacker's model, the distillation ordinary model, and the distillation this paper's model (assuming the dataset is CIFAR-100, ResNet18 is the vehicle model, and MobilenetV2 is the attacker model). The experimental results show that the model proposed in this paper can reach convergence faster under the condition that the accuracy is not inferior to that of the normal model, and the model accuracy of the attacker will be severely reduced in the face of distillation.

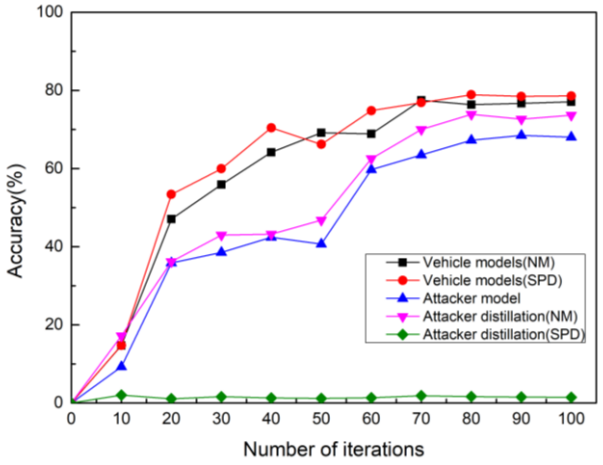

Figure 5. Several iterations of the model. As distillation attacks continue, the performance of the model against a normal model attacker will improve, but its performance will severely degrade against an SPD model.

Table 6. data-free knowledge distillation results

Dataset	CIFAR-10		CIFAR-100	
Vehicle models	Accuracy	DAFL Accuracy	Accuracy	DAFL Accuracy
ResNet18(NM)	94.78	91.56	77.45	71.04
ResNet18(SPD)	94.29(-0.49)	85.58(-5.98)	77.38(-0.07)	65.28(-5.76)

To verify that the model proposed in this paper is still valid in a data distillation-free (DAFL) environment, we used the classical ResNet18 as the underlying network and conducted experiments using the method proposed by Chen et al. [3], and obtained the results shown in Table 6. Comparing the data in the table, it can be seen that the attacker's gain will be greatly reduced compared to distillation ordinary model by DAFL's method to steal the user privacy of distillation resistant vehicle deep learning model.

To verify the superiority of the method proposed in this paper, the accuracy of the models constructed by the strengthen prevent distillation (SPD) scheme, the ordinary prevent distillation (PD) scheme, the adaptive false alarm (AFA) scheme and the normal model (NM) are compared, and the changes in model accuracy caused by the distillation of the models constructed by the above algorithms are compared (still assuming that the dataset is CIFAR-100, ResNet18 is the vehicle model , MobilenetV2 is the attacker model), it can be seen that the SPD scheme has the least impact on the accuracy of the model itself and produces the best model protection in the face of knowledge distillation. The experimental results are shown in Fig. 6.

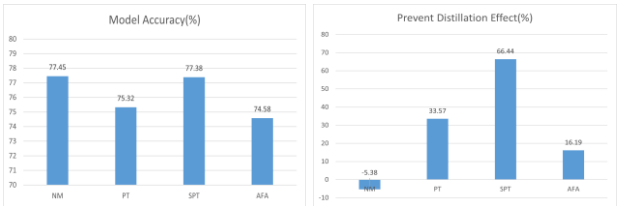

Figure 6. Performance and prevent to distillation effects of several models

4.3. Qualitative Analysis

The scheme is effective in the IoV environment because it maximizes and reinforces the output of the correct categories and confounds the ranking of the incorrect categories. A visualization of the output probability of the ResNet-18 model on the CIFAR-10 dataset is shown on Fig. 7 to qualitatively analyze the reasons why the scheme is effective.

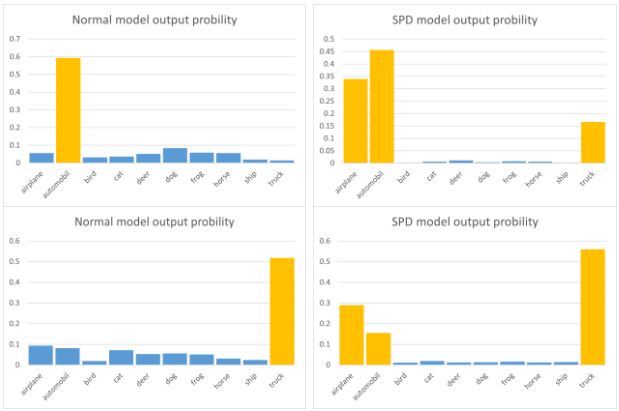

Figure 7. Visualization of softmax output (The car is on the top and the truck is on the bottom)

As shown in Fig. 7, it visualizes the logit response of the normal ResNet-18 and its counterpart after processing with the strengthen prevent distillation function, using the output of a truck and a car as examples. It can be seen that the normal model always outputs one peak, but the output response of the strengthen prevent distillation vehicle deep learning model consists of multiple peaks. Multi-peak logic misleads the learning process of knowledge distillation and degrades the performance of the attacker model, giving the attacker model a false sense of generalization, then the malicious roadside unit or base station also learns the wrong knowledge from the vehicle model, leading to a decrease in its own accuracy, which in turn protects the security of the vehicle model as well as the privacy of the user.

4.4. Ablation Experiment

As shown in Fig. 8, the proposed method is capable of reducing model performance for malicious attackers, irrespective of the selected value of parameter ω, which varies from 0 to 0.01 on CIFAR-100 dataset (assuming that the dataset is CIFAR-100, ResNet18 is the vehicle model , ShufflenetV2 is the attacker model). Additionally, by adjusting the value of w, it is possible to achieve a balance between performance loss and resistance to distillation attacks. Specifically, a higher value of w can result in a more resilient model against distillation attacks, but at the expense of greater accuracy loss.

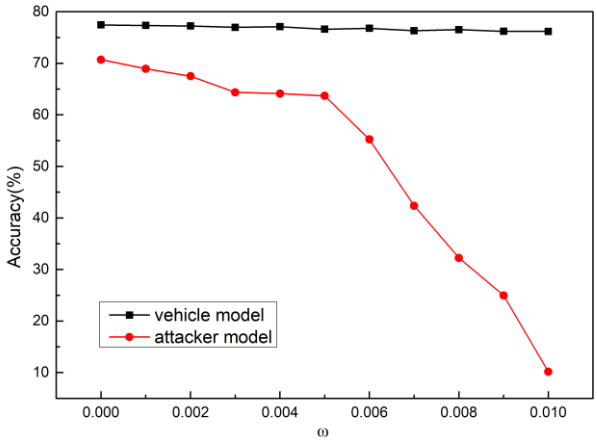

Figure 8. Several iterations of the model. As the balance parameter ω increases, the performance of the attacker model will severely degrade, but at the cost of the defender model's performance being negatively impacted as well.

5. Conclusion

In practice, the owner of the vehicle deep learning model can achieve the effect that the model cannot be stolen by resisting distillation training, self-enhancement training, and adding local differential privacy noise without sacrificing its own performance. The related performance improvements are due to the fact that resistance to distillation training has reconstructed the internal structure of the model, self-enhancement training has expanded the degree of reconstruction, and adding differential privacy noise has further improved the privacy protection efficiency of the scheme. Even if attackers have the same training data, they do not have the ability to use knowledge distillation to clone published models, as their model performance would be severely degraded instead of being boosted as usual where performance degradation is unacceptable in some security-critical environments, such as autonomous driving, so that cloned models or illegal data theft through knowledge distillation can be avoided.

5.1. Study Limitations

Extensive experiments conducted on multiple datasets quantitatively show that the vehicle deep learning model with strengthened prevent distillation is effective in either the standard knowledge distillation or data-free knowledge distillation settings. This scheme is more complex and takes longer in the training process, but it is acceptable in the model training phase, and the size of the model itself is not affected.

5.2. Future Scope of Research

In the future, other methods will be explored to improve the current resistance to distillation and to speed up the training time of the model so that the proposed concept can be generally applied in practice. At the same time, we will also consider adding a model watermark to protect the ownership of the vehicle model.

Acknowledgements.

This work is supported by the National Natural Science Foundation of China under Grant no. 62162009 and 62101478, the Key Technologies R&D Program of He'nan Province under Grant No. 212102210084 and 222102210048, the Foundation of He'nan Educational Committee under Grant No. 18A520047, the Scientific Research Innovation Team of Xuchang University under Grant No. 2022CXTD003, and Innovation Scientists and Technicians Troop Construction Projects of Henan Province.

References

[1] Mekala M S, Dhiman G, Patan R, et al. Deep learning-influenced joint vehicle-to-infrastructure and vehicle-to-vehicle communication approach for internet of vehicles[J]. Expert Systems, 2022, 39(5): e12815.

[2] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. arXiv preprint arXiv:1503.02531, 2015, 2(7).

[3] Chen H, Wang Y, Xu C, et al. Data-free learning of student networks[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3514-3522.

[4] Lopes R G, Fenu S, Starner T. Data-free knowledge distillation for deep neural networks[J]. arXiv preprint arXiv:1710.07535, 2017.

[5] Yin H, Molchanov P, Alvarez J M, et al. Dreaming to distill: Data-free knowledge transfer via deepinversion[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 8715-8724.

[6] Wu Z, Wang Z, Wang Z, et al. Towards privacy-preserving visual recognition via adversarial training: A pilot study[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 606-624.

[7] Zhao Y, Zhao J, Yang M, et al. Local differential privacy-based federated learning for internet of things[J]. IEEE Internet of Things Journal, 2020, 8(11): 8836-8853.

[8] Lu Y, Huang X, Zhang K, et al. Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 4298-4311.

[9] Liu H, Wang H, Gu H. HPBS: A hybrid proxy based authentication scheme in VANETs[J]. IEEE Access, 2020, 8: 161655-161667.

[10] Zhang J, Chen D, Liao J, et al. Model watermarking for image processing networks[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 12805-12812.

[11] Fan L, Ng K W, Chan C S. Rethinking deep neural network ownership verification: Embedding passports to defeat ambiguity attacks[J]. Advances in neural information processing systems, 2019, 32.

[12] Ribeiro D A, Melgarejo D C, Saadi M, et al. A novel deep deterministic policy gradient model applied to intelligent transportation system security problems in 5G and 6G network scenarios[J]. Physical Communication, 2023, 56: 101938.

[13] Guerrero-Ibañez J, Contreras-Castillo J, Zeadally S. Deep learning support for intelligent transportation systems[J]. Transactions on Emerging Telecommunications Technologies, 2021, 32(3): e4169.

[14] Wang W, Pei Y, Wang S H, et al. PSTCNN: Explainable COVID-19 diagnosis using PSO-guided self-tuning CNN[J]. Biocell, 2023, 47(2): 373-384.

[15] Wang W, Zhang X, Wang S H, et al. Covid-19 diagnosis by WE-SAJ[J]. Systems Science & Control Engineering, 2022, 10(1): 325-335.

[16] Ma H, Chen T, Hu T K, et al. Undistillable: Making a nasty teacher that cannot teach students[J]. arXiv preprint arXiv:2105.07381, 2021.

[17] Zhang L, Song J, Gao A, et al. Be your own teacher: Improve the performance of convolutional neural networks via self distillation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 3713-3722.

[18] Chen P, Liu S, Zhao H, et al. Distilling knowledge via knowledge review[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 5008-5017.

[19] Hou Y, Ma Z, Liu C, et al. Learning lightweight lane detection cnns by self attention distillation[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2019: 1013-1021.

[20] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[21] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.

[22] Abadi M, Chu A, Goodfellow I, et al. Deep learning with differential privacy[C]//Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. 2016: 308-318.

[23] Arachchige P C M, Bertok P, Khalil I, et al. Local differential privacy for deep learning[J]. IEEE Internet of Things Journal, 2019, 7(7): 5827-5842.

[24] Wei K, Li J, Ding M, et al. User-level privacy-preserving federated learning: Analysis and performance optimization[J]. IEEE Transactions on Mobile Computing, 2021.

[25] Zheng Y, Zou L, Zhang W, et al. Contract-based Cooperative Computation and Communication Resources Sharing in Mobile Edge Computing[J]. Journal of Grid Computing, 2023, 21(1): 14.

[26] Kariyappa S, Qureshi M K. Defending against model stealing attacks with adaptive misinformation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 770-778.