applied sciences

MDPI

Article

CANPerFL: Improve In-Vehicle Intrusion Detection Performance by Sharing Knowledge

Thien-Nu Hoang [1], Md Rezanur Islam [2], Kangbin Yim [3] and Daehee Kim [1,*]

[1] Mobility Convergence, Soonchunhyang University, Asan-si 31538, Republic of Korea
[2] Software Convergence, Soonchunhyang University, Asan-si 31538, Republic of Korea
[3] Information Security Engineering, Soonchunhyang University, Asan-si 31538, Republic of Korea
* Correspondence: daeheekim@sch.ac.kr

Abstract: The Controller Area Network (CAN) is a widely used communication protocol in automobiles, but it is vulnerable to various types of attacks. To address this issue, researchers have been exploring the use of intrusion detection systems (IDS) for the CAN bus. Deep learning and machine learning have been proven to be powerful tools for detecting intrusions accurately and quickly. However, deep learning models require large amounts of data to achieve optimal performance, which can be challenging in the case of a CAN bus IDS. To overcome this challenge, we propose a novel machine learning-based IDS called CANPerFL that uses a personalized federated learning scheme to aggregate datasets from different car models. By building a universal model trained on a small amount of data from each manufacturer, we can provide global knowledge that can be transferred to improve the performance of each participant. To demonstrate the efficiency of the proposed model, we collected a real CAN dataset consisting of three different car models: KIA, BMW, and Tesla. The experimental results show that the proposed model increases F1 scores by 4% overall, compared to baselines. Moreover, the proposed system provides significant advantages when the local dataset of each participant is relatively small. According to our experiments, the proposed models can achieve F1 scores of more than 90% with at least 30k training samples on each client. Finally, we show empirically that each participant takes benefits from joining the CANPerFL system.

Keywords: controller area networks; in-vehicle networks; injection attacks; intrusion detection system; personalized federated learning

Citation: Hoang, T.-N.; Islam, M.R.; Yim, K.; Kim, D. CANPerFL: Improve In-Vehicle Intrusion Detection Performance by Sharing Knowledge. *Appl. Sci.* **2023**, *13*, 6369. https://doi.org/10.3390/app13116369

Academic Editor: Christos Bouras

Received: 19 March 2023
Revised: 22 April 2023
Accepted: 20 May 2023
Published: 23 May 2023

1. Introduction

Intelligent and connected vehicles have become a key research area for the automobile industry. These vehicles assist drivers with various advanced features, such as traction control systems (TCS) and automatic collision avoidance systems (ACAS). However, these technologies unintentionally introduce several interfaces for attackers who can access a vehicle and control it for malicious intentions. The idea has been shown practically for a few years. For example, Miller and Valasek [1] demonstrated how an attacker could remotely take control of vehicle's systems, including the engine, brakes, and steering. One critical and vulnerable part of vehicles is in-vehicle networks, e.g., the controller area network (CAN) system. CAN is a high-speed communication protocol used in automobiles to allow different electronic devices to communicate with each other [2]. One potential vulnerability is the lack of authentication, which means that any device connected to the CAN bus can potentially send messages. This can allow an attacker to inject false or malicious messages onto the bus, causing problems with the vehicle's operation. To address these security concerns, an intrusion detection system (IDS) can be installed on an electronic control unit (ECU) of a CAN bus to detect and alert the user of any potential attacks or anomalies in the messages being sent on the bus.

Detection performance is one of the most critical properties of an IDS, with high accuracy and low missed detection rate being essential. Among several approaches such as

rule-based and statistical methods, deep learning-based IDSs are preferred to learn normal behavior and detect anomalies on the CAN bus based on its noticeable achievements. A robust deep learning model for the in-vehicle network IDS, however, requires a large dataset, which takes a long time to collect, particularly with abnormal data. To tackle this problem, previous studies propose unsupervised models [3–5], and semi-supervised models [6]. Unsupervised models learn only normal patterns in the training dataset and detect intrusion by measuring the difference between the learned representation and the sample test. Meanwhile, semi-supervised models utilize the unlabeled data as additional knowledge, which can improve the overall performance of the model when a small labeled dataset is available. Yet, these approaches have some limitations. First, an appropriate threshold needs to be determined in the unsupervised model, which can affect the final result. For example, several studies set a high threshold [3], leading to a high false positive rate. Second, in the semi-supervised models [6], there is an assumption about the attack data amount in unlabeled data, which is difficult to achieve in reality. Moreover, transfer learning is lately applied to design an in-vehicle IDS to address the problem of scarce data. The idea behind transfer learning is that the model's knowledge of a source task can be exploited as a starting point for learning other target tasks, allowing the model to learn more quickly and effectively than if it was starting from scratch. As a result, the target tasks do not need a large dataset. For example, the study [7] successfully transferred knowledge of a pretrained model trained on a large CAN bus dataset to a smaller one in a different manufacturer. In this study, we aim to extend the idea of transfer learning by incorporating different data sources from different car manufacturers.

Existing transfer learning and multiple task learning studies have confirmed that the same domain data may share a global feature representation, despite statistical differences across datasets [8]. In IDS research for CAN bus context, the study [7] utilized only a specific model trained solely on a particular CAN dataset for transferring. This leads to the question whether we could utilize all data from different car manufacturers to improve the performance of IDS for each participant. Nevertheless, the majority of car manufacturers do not share their CAN bus dataset due to confidentiality. As a result, a federated learning scheme is suitable in this circumstance. To overcome this challenge, we propose a personalized federated learning training system called CANPerFL, which extracts a shared feature representation between different car models. In the CANPerFL, all car manufacturers will jointly train a global model which will be then treated as a pretrained model for individual transfer learning of each client. Because the global model learns diverse patterns of various CAN datasets, the latent features extracted from the model will help improve the performance of each client's model by training the frozen global model with a top classification layer on local data. To sum up, the main contributions of our research are as follows.

1. We propose a novel system called CANPerFL which utilizes federated learning to develop a robust in-vehicle IDS, in which different car manufacturers can collaborate in learning to detect attacks without violating data privacy. To the best of our knowledge, our study is the first in applying federated learning across different car manufacturers for learning the IDS model.
2. We collect real CAN datasets consisting of three different car models: Kia, Tesla, and BMW to test the proposed idea with several extensive experiments.
3. The proposed system improves the F1 score of the local models by 4% on average. In addition, we empirically prove that the personalized FL can automatically encourage each client to participate in the global training process, which provides the global learning process a better representation.

The rest of this paper is structured as follows. Section 2 introduces fundamental knowledge about CAN bus as well as security threat models focused on in this study. Then, we provide a review of existing studies related to in-vehicle IDS in Section 3. Next, the details of the proposed system, i.e., personalized federated learning will be described in Section 4. In Section 5, we demonstrate various comprehensive experiments which show

the effectiveness of our idea. Finally, the paper ends with Section 6, which summarizes our key findings and provides promising ideas for the future.

2. Controller Area Network Background

The Controller Area Network (CAN) is a serial data communication technology that was developed by Robert Bosch GmbH in the 1980s to facilitate efficient communication between automotive applications. The data frame format of CAN 2.0B, shown in Figure 1, uses either a single or double wire arrangement with data rates up to 1 Mb/s [2]. This protocol allows each module in a vehicle to communicate with one another, transmitting and receiving driving data. Multiple ECUs are connected in parallel through the CAN bus, including C-CAN, M-CAN, and B-CAN [9]. C-CAN is responsible for generating data such as speed, RPM, and brake data, while B-CAN is a network that communicates messages to control various parts of the vehicle. M-CAN is responsible for navigating media and entertainment information [10].

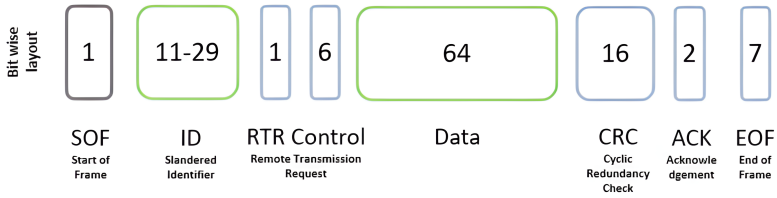


Figure 1. Controller Area Network Specification.

The CAN bus frame consists of four types: data frame, error frame, overload frame, and remote frame [2]. Each of these frames has a standard structure that includes different fields: arbitration identifier, data, acknowledgment, and others. The definition of each field of a CAN frame is shown in Figure 1. The start of frame (SOF) indicates the beginning of a CAN message with a dominant bit, and it informs all nodes to begin CAN message transmission. The arbitration field consists of 11 bits, which can be expanded to 29 bits. The control field, also known as the check field, provides data from the receiver to check if all transmitted packets are working properly. The data field contains actual information for CAN nodes to perform actions and can be 8 bytes. The CRC field, also known as the cyclic redundancy code or security field, is a 15-bit error detection tool that checks for packet validity. The acknowledge (ACK) field, also known as the confirmation field, ensures that the receiver nodes have received the CAN packet correctly. If an error is detected during the transmission process, the transmitter is promptly notified by the receiver to resend their data packets. The EOF field indicates the end of the CAN datasets with a flag of a dormant bit. Data frames are used to transfer CAN packet information (Request Command) from the sender. Therefore, the CAN bus communicates with other nodes by transmitting packets through data frames. Whenever the remote transmission request (RTR) bit is flagged as dominant, it becomes a CAN data frame.

Advanced technology in the automotive industry has led to an increase in the number of vehicles that are connected to external devices and services, making them susceptible to attacks. This includes the use of wireless connections, such as Bluetooth, Wi-Fi, and mobile phone networks, which opens an access point for the assaulters [11]. In addition, electric vehicles' OTA updates also present a potential vulnerability for attackers, where the Tesla S model became under attack through a Wi-Fi hotspot [12]. These entry points for attack include things such as GPS, LiDAR, camera sensors, tire pressure monitoring systems, and the OBD-II port [13–15]. These can all be manipulated to create serious attack scenarios. The internal architecture of vehicles has become more complex as a result of these new technologies, making them more vulnerable to attacks.

3. Related Works

An intrusion detection system (IDS) is a critical countermeasure to detect and protect CAN bus against cyberattacks on the vehicle's systems. There are two main types of IDS on the CAN bus: CAN packet-based and ECU's characteristic-based IDSs [16]. While CAN packet-based approaches exploit various features of CAN frames, ECU's characteristic-based approaches focus on the physical features of ECUs. In this section, we provide a comprehensive literature review of CAN packet-based related works using techniques close to CANPerFL such as machine learning, federated learning, and transfer learning.

3.1. Machine Learning-Based IDS

Many researchers have applied several machine learning and deep learning techniques to detect invasion on the CAN bus scrupulously. For instance, the idea of using hierarchical temporal memory (HTM) was proposed [17] to develop a distributed anomaly detection system for the in-vehicle network. The proposed system predicted the next bit based on the data stream of each CAN ID and calculated the anomalous score based on the incoming data and the prediction to determine if any attack exists.

Song et al. [18] presented an induced Inception ResNet model trained based on a sequential binary CAN ID matrix. The study became state-of-the-art in the field when achieving a relatively small error rate compared to other models. However, the proposed architecture is considered to be complex for deploying in an ECU.

The IDS model was defined as an LSTM autoencoder fed by various signals along with corresponding CAN IDs and trained with reconstruction loss [19]. While the idea is interesting, the results of this study need to be further improved.

Derhab et al. [20] introduced a new way of encoding CAN data by using a histogram of sequential CAN data fields. Then, a one-class support vector machine (OCSVM) model was trained to detect attacks on the CAN bus. However, the model required a large window size to achieve high performance.

Desta et al. [21] developed a lightweight deep-learning CNN model based on recurrence plots built from various continuous CAN IDs. Although the model was able to adapt to time and hardware constraints, it was not good with multiclass classification.

Sequential patterns existing in CAN IDs can be utilized the to build a model, which predicts the next CAN ID given a sequence [22]. For prediction, the authors used a bi-directional generative pretrained transformer (GPT), which is a recent advanced model in natural language processing. There are two bidirectional GPT models implemented as intrusion detection systems (IDS), where the CAN ID sequences are converted into integers, and performance is evaluated using negative log-likelihood (NNL).

To improve the performance of IDS, Zhang et al. [23] combined both rule-based and machine learning-based methods to develop a hybrid two-stage IDS. The proposed system is shown efficient when tested on their own collected datasets.

Electric signals generated during transmission of Controller Area Network (CAN) packets contain slight variations that can identify individual Electronic Control Units (ECUs). Murvay et al. [24] utilized the CAN ID field of CAN packets to create digital fingerprints for all ECUs. By measuring the electric signals associated with the CAN ID field, the authors generated fingerprints for the IDS. The IDS was evaluated in a simulated environment with ten USB-to-CAN devices and five CAN development boards.

Lee et al. [25] found that malicious CAN packets can affect response time distributions of ECUs. They proposed an IDS that periodically sends request messages to all ECUs and compared their response times to known distributions. If a response time deviates from its expected distribution, it may indicate malicious packet injections. However, the IDS is limited to detecting attacks that affect response times.

Markovitz et al. [26] identified four common data types in the Controller Area Network (CAN) data field: constant value, multi-value, counter value, and sensor value. As commercial vehicle CAN packet specifications are confidential, an algorithm was developed to assign each part of the 64-bit CAN data field to one of the four data types. Classifica-

tion and specific rules for each data type were used to identify potential attacks on the automotive CAN system.

Choi et al. [27] proposed VoltageIDS, an IDS for automotive networks that utilizes the physical properties of electrical signals for transmitting CAN messages. The system operates in three phases: feature extraction, feature selection, and intrusion detection. During the feature extraction phase, VoltageIDS extracts 60 features from normal CAN message signals, which are filtered in the feature selection phase to retain the most relevant features. In the intrusion detection phase, a multi-class classifier, such as Support Vector Machine, is constructed using attack-free CAN data to predict whether a message is normal or an intrusion.

The authors of [28] proposed LSTM neural networks for anomaly detection in CAN bus messages in vehicles. Two pattern features, data and time intervals, were used for classification. The multi-dimensional LSTM framework combined these features and included a prediction and detection process. The proposed mobile edge-assisted multi-task LSTM reduced computation time and cost by enabling parallel computation on multiple servers.

Zhang et al. [29] developed an in-vehicle network intrusion detection approach using a Binarized Neural Network (BNN) and Field-Programmable Grid Arrays (FPGAs). BNN uses binary values to accelerate intrusion detection, reducing memory usage and energy consumption. FPGAs improve performance by allowing for concurrent task processing. The IDS was three times faster than traditional IDSs, and 128 times faster after FPGA acceleration.

A combination of CNN, LSTM, and attention mechanism was developed to build a robust model for CAN IDS [30]. In particular, the proposed approach uses a CNN to extract features from raw data, which are then fed into an LSTM to capture temporal dependencies. An attention mechanism is then applied to weight the contributions of different features to the anomaly detection process. The authors showed the effectiveness of their approach through a comprehensive evaluation using a publicly available dataset. The results demonstrate that the proposed approach outperforms several baseline methods in terms of accuracy, precision, recall, and F1 score.

Al-Jarrah et al. [31] implemented a multi-model approach to classify attacks, utilizing both LSTM and ConvLSTM. While the former was utilized to process table data, the latter employed a recursion graph. The integration of the two models resulted in a 2% increase in accuracy, with an overall accuracy of 95.1%. It should be noted that this approach demands higher computational power and incorporates data appearing time, which may not significantly contribute to deep learning models.

3.2. Federated Learning for In-Vehicle Intrusion Detection

Applying machine learning and deep learning techniques to design a robust in-vehicle IDS is an active research topic. Yet, only a few studies have attempted to apply federated learning in the field. To the best of our knowledge, there are two recent studies closely related to our proposed idea.

Hussain et al. [32] designed a ConvLSTM model which will be trained in a federated learning manner on the vehicle level. The model was developed to solve supervised multiclass classification, but the question of how to label data at the vehicle level to train the proposed scheme was not addressed. Meanwhile, they focused on improving the client selection strategy using a deep reinforcement approach.

A federated random forest model with blockchain technology was developed to tackle poisoning attacks [33]. Since the study aims to develop secure blockchain storage for the global model, the issue of data heterogeneity was not mentioned. In addition, the study considered federated learning at the vehicle level, whereas our scheme is at the car manufacturer level where data labels are available.

3.3. Transfer Learning for In-Vehicle Intrusion Detection

Another direction that relates to our work is applying transfer learning for in-vehicle intrusion detection. For example, Kang et al. [34] built an LSTM-based approach, which was trained on a specific car manufacturer and utilized for transferring to another CAN dataset belonging to other car models.

3.4. Research Gaps

Many studies have shown the powerful capabilities of machine learning and deep learning in intrusion detection, especially for CAN bus data. However, the effectiveness of these models heavily relies on the availability of a significant amount of data. Additionally, due to the distribution discrepancy between data from different car manufacturers, as well as data confidentiality, building a universal model for different kinds of car manufacturers is challenging. Although transfer learning can be applied, choosing a specific car manufacturer introduces a bias problem. To address these challenges, we propose a novel deep learning-based IDS called CANPerFL. This approach leverages federated learning to take advantage of global features obtained from different car manufacturers, while preserving the privacy of the data. The final global features can be used to produce robust local models for participants, even with limited data. By doing so, CANPerFL can enhance the performance of intrusion detection in vehicles while addressing the challenges of data distribution discrepancy and confidentiality.

4. Personalized Federated Learning

4.1. Federated Learning

Federated Learning (FL) [35] is an idea of training a model from distributed data across clients while preserving data privacy. The FL setting includes a central server and multiple M clients. Each client $k \in [M]$ owns a dataset $\mathcal{D}_k$ having a size of n_k samples and following an underlying distribution over $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the label space. The final outcome of the learning process is a set of parameters of a mapping model from input space to label space, e.g., $h_{\boldsymbol{\theta}} : \mathcal{X} \rightarrow \mathcal{Y}$. In general, the loss of model parameters $\boldsymbol{\theta}$ on the k-th client is defined as:

$$f_k(\boldsymbol{\theta}) = \frac{1}{n_k} \sum_{i=1}^{n_k} l(h_{\boldsymbol{\theta}}(\mathbf{x}_{k,i}), \mathbf{y}_{k,i}), \tag{1}$$

where $(\mathbf{x}_{k,i}, \mathbf{y}_{k,i})$ is the i-th sample in $\mathcal{D}_k$. To ensure the communication and privacy constraints, the central server exploits the data across clients by learning a global model that minimize the loss f_k for each client. Concretely, the global loss can be defined as the average of the client losses weighted by number of samples:

$$\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{k=1}^{M} n_k f_k(\boldsymbol{\theta}) = \frac{1}{N} \sum_{k=1}^{M} \sum_{i \in \mathcal{D}_k} l(h_{\boldsymbol{\theta}}(\mathbf{x}_{k,i}, \mathbf{y}_{k,i}), \tag{2}$$

where $N = \sum_{k=1}^{M} n_k$ is the total number of samples across clients. As illustrated in Figure 2, the clients send only their models' weights to the server at each step. As a result, the data of each client will be kept privately.

Federated Averaging (FedAvg) [35] is the most popular algorithm to solve (2). On each round t, the server picks a set of clients randomly to perform federated training. Each selected client takes the global parameters $\boldsymbol{\theta}_{t-1}$ from the server, which are used for optimizing (1) by E stochastic gradient descent (SGD) steps. The client k then sends $\boldsymbol{\theta}_t^k$ back to the server which averages all received clients's parameters to obtain the global parameters. The details of FedAvg are summarized in Algorithm 1. Compared to FedSGD corresponding to $E = 1$, FedAvg converges faster and requires fewer communication overhead.

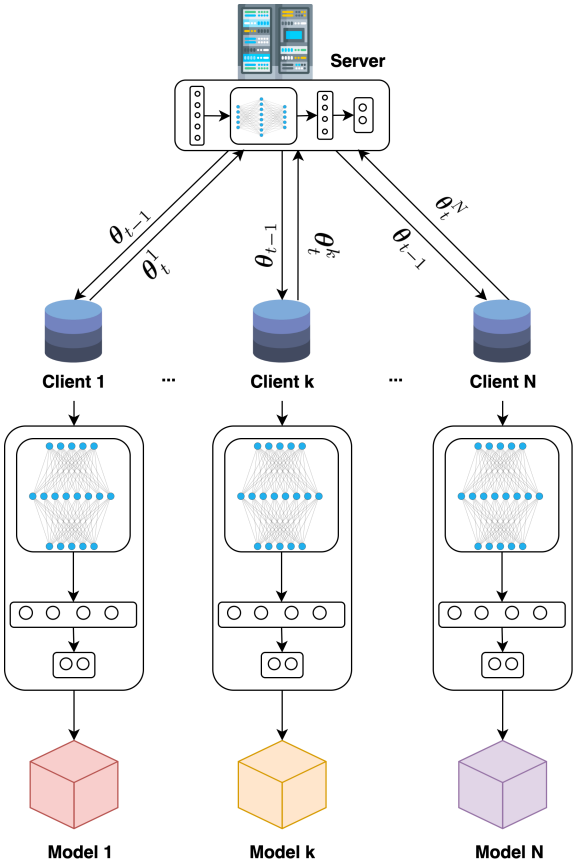


Figure 2. The overview of the personalized federated learning training scheme.

Algorithm 1 FedAvg algorithm.

Input: number of clients M, local batch size B, local epochs E, learning rate α
Output: global parameters $\boldsymbol{\theta}$

Server executes:
Initialize $\boldsymbol{\theta}_0$
for each round $t = 1, 2, 3 \ldots$ **do**
  for each client $k = 1, 2, \ldots, M$ in parallel **do**
    $\boldsymbol{\theta}_t^k = \textbf{ClientUpdate}(k, \boldsymbol{\theta}_{t-1})$
  end for
  $\boldsymbol{\theta}_t = \sum_{k=1}^{M} \frac{n_k}{N} \boldsymbol{\theta}_t^k$
end for

ClientUpdate$(k, \boldsymbol{\theta})$:
$\mathcal{B}$ = split data into batches of size B
for each local epoch = $1, 2, \ldots, E$ **do**
  for batch $b \in \mathcal{B}$ **do**
    $\boldsymbol{\theta} = \boldsymbol{\theta} - \alpha \nabla l(h_{\boldsymbol{\theta}}(b))$
  end for
end for

4.2. FedAvg with Fine-Tuning

Because of heterogeneous data, the global model cannot serve well for all clients. Therefore, a personalized step is proposed, in which the global parameters are fine-tuned on each client before being used. Although the initial purpose of FedAvg is not about representation learning, the study [36] proved that FedAvg can learn an effective representation by exploiting the diversity of the clients. As a result, each client can extract the shared feature representation to train an individual model based on their local data. In particular,

a class of models can be partitioned into two parts: a representation h_{ϕ}^{rep} and a prediction module h_{ψ}^{head}, where ϕ and ψ contain representation and head parameters. The prediction of a sample $\mathbf{x} \in \mathcal{X}$ can be written as follows:

$$h_{\theta}(\mathbf{x}) = (h_{\psi}^{head} \circ h_{\phi}^{rep}(\mathbf{x}) = h_{\psi}^{head}(h_{\phi}^{rep}(\mathbf{x})). \quad (3)$$

Multi-task learning studies have assumed that there exists a common representation $h_{\phi*}^{rep}$ which smoothes the learning process of head module $h_{\psi*,i}^{head}$, such that $h_{\psi}^{head} \circ h_{\phi}^{rep}$ performs well for task i. The study [36] adopted the assumption and stated that all clients in FedAvg can learn this representation.

4.3. Training IDS across Different Manufacturers

In this study, we exploit the personalized FL scheme to build a universal IDS for different automobile manufacturers. We suppose that CAN data from different car models own a common feature representation in spite of distribution discrepancy. Thus, the main goal is to test whether all participants take benefit from the learned global representation with CAN data in the case of car manufacturers' diversity. We utilize the idea of data processing and model architecture from our previous work [7]. Specifically, sequential CAN IDs are stacked to form a CAN frame matrix, which is the input for training a compact Resnet model. Compared to previous studies [7,34,37], which apply transfer learning from a specific car model, the global representation from our proposed system contains all features from various clients. Hence, the results are much better, which will be verified in the following section.

5. Experiments and Results

5.1. Data Collection

Although OBD-II is commonly used by researchers to collect data for diagnostic purposes, not all types of data can be collected through this method because not all ECUs are connected to the OBD-II port. In order to collect data from these ECUs, we use a method called ECU direct approach (EDA) [38], which involves collecting data directly from the internal gateway of the vehicle using line-tapping tools. The PEAK CAN system is used as an interfacing device to access the CAN network. The data collection process is illustrated in Figure 3.

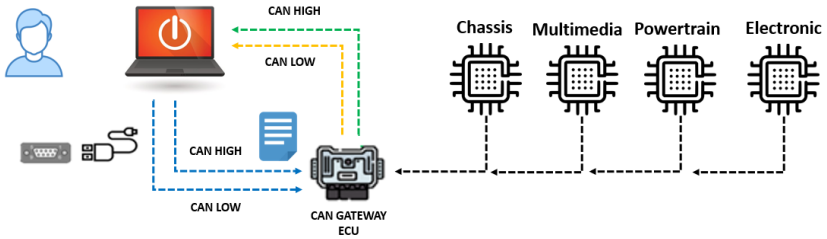


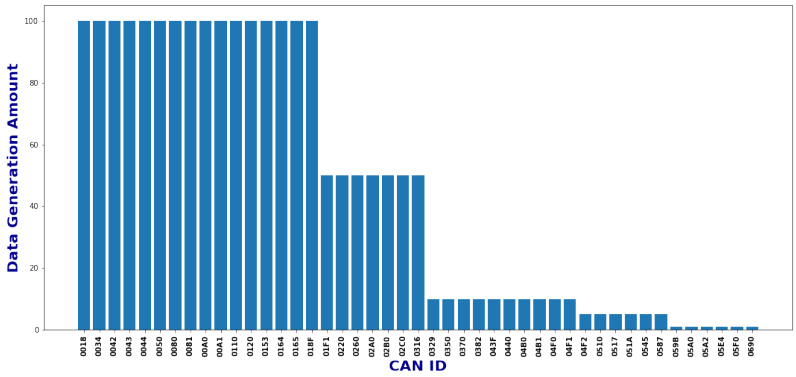
Figure 3. Data Collection Layout.

There are two types of attack data used in this experiment: Fuzzing and Replay. A fuzzing attack is initiated by injecting a high volume of data with a high volume of CAN ID, and as a result, all nodes become busy and oppose the actual command [39]. In a replay attack, the attacker first observes the CAN message and collects data from the targeted vehicle, then injects similar data into the network. This type of attack is difficult to detect compared to other attacks [40]. To conduct the Fuzzing and Replay attacks, we made an effort to gather a variety of data types. The duration of data injection was established to be between 3–5 s. For the Fuzzing attack, 100 and 500 data packets were injected per segment, and for the Replay attack, two randomly selected data injections were executed per segment.

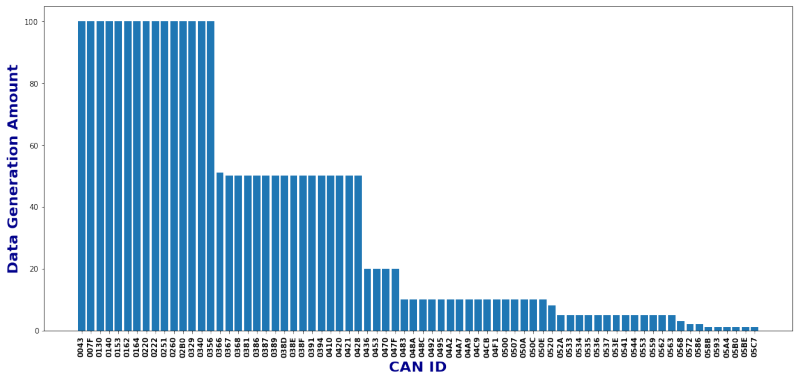
5.2. Data Specification

The in-vehicle network contains unique data compared to other vehicles, even those of the same manufacturer and model [41]. A brief analysis result is shown in [42], where multiple vehicle data were tried to be generalized. This is because every vehicle produces data according to its own CAN Data Base Container (DBC) file, which is confidential and owned by the manufacturer. As a result, it is challenging to translate CAN messages, and traditional IDS systems designed from specific vehicle data will not work for other vehicles. Federated learning can provide a solution to this problem.

In this article, three different vehicles were used in federated experiments, including both mechanical and electronic types of vehicle data. For data structure explanation, we utilized four vehicles, two of which were from the same manufacturer and model. Table 1 shows the total number of CAN IDs and the amount of data generated per second for each vehicle. It is noticeable that every vehicle has a different data generation pattern. BMW has the highest number of CAN IDs but low data generation, while Tesla has a low number of CAN IDs but the highest data generation. From Table 2, it can be observed that despite being the same model and manufacturer, both vehicles have different CAN IDs for their functions. In addition, the data generation process is different from each other according to their IDs (see Figure 4). These comparisons demonstrate the significance of a centralized IDS system utilizing federated learning, highlighting its importance.



(**a**)



(**b**)

Figure 4. Data distribution according to CAN ID. (**a**) Kia Old. (**b**) Kia New.

Table 1. Data variance on CAN from different vehicles.

Name	Total CAN ID	Data Generation (s)
BMW	135	876
Kia (Old)	47	2086
Kia (New)	84	2630
Tesla	69	3128

Table 2. CAN ID variance from same manufacturer and model on the same functions.

	Speed	RPM	Gear
Kia (Old)	440	316	043F
Kia (New)	386	366	367

5.3. Data Processing and Classifier Architecture

Building on our previous work [6], the proposed model in this study uses only CAN IDs as features. In particular, we stacked a series of continuous CAN IDs represented in binary format to form a matrix that captures the temporal and spatial structure of the CAN ID sequence. This CAN IDs matrix is then fed into a convolutional neural network (CNN) for classification. To balance accuracy and computational efficiency, we designed a compact Resnet architecture, which is shown in detail in Figure 5. It is important to note that the primary goal of this study is to test the performance of the proposed framework CANPerFL. Therefore, we did not focus on feature engineering or apply a novel model architecture. Instead, we employed an efficient feature extraction method and a well-known deep learning model to streamline the process.

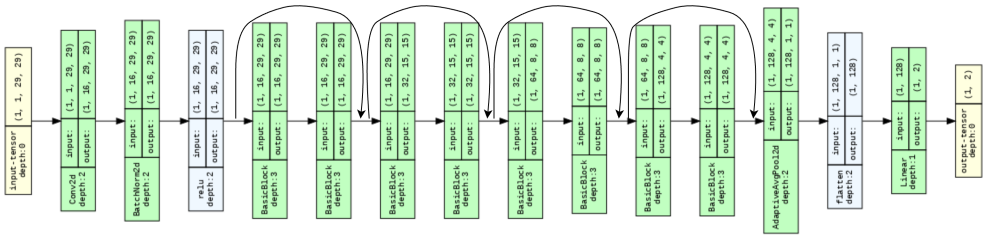


Figure 5. Model architecture used for local IDS.

After being preprocessed, the dataset is then spitted into three non-overlap subsets including training, validation, and test sets. However, we observed that the class distribution in the raw dataset does not match with the real environment, which means there are large attack samples over the normal samples. The ratio between the total number of attack samples over the total number of the dataset is defined as the attack density. Then, a data sampling process (Figure 6) is performed to ensure an attack density in the experimental environment, in which the attack density is uniformly chosen in range $[0.2, 0.3]$. Table 3 shows the details of training data having approximately 50k samples in total after sampling. While the deep learning model is developed by Python 3.9 and Pytorch 1.12, the Flower framework [43] is used for demonstrating the federated learning scheme. In addition, all the experiments run on a server provided with 32 Intel(R) Xeon(R) Silver 4108 CPUs @ 1.80 GHz, a memory of 128 GB, and an Nvidia Titan RTX 24 GB GPU.

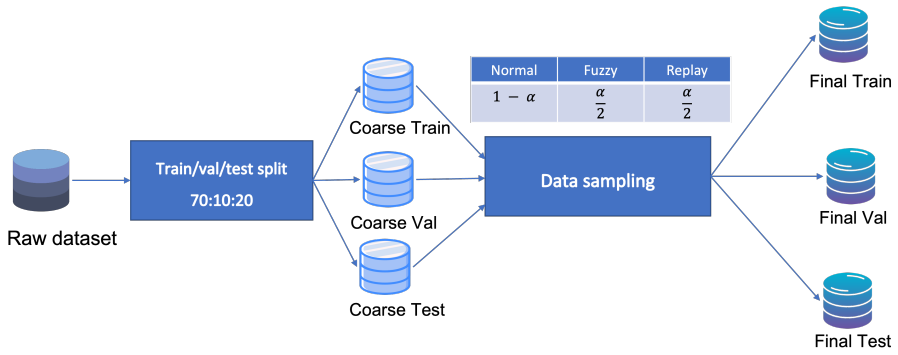


Figure 6. The sampling process.

Table 3. The details of training data with approximately 50k samples in total.

Dataset	#Samples (KIA)	#Samples (Tesla)	#Samples (BMW)
Training	37,132	30,237	36,004
Validation	7331	11,806	11,197
Test	7331	11,806	11,197

5.4. Evaluation Metrics

We used various metrics for evaluation such as error rate (ER), false alarm rate (FAR), precision (Prec), recall (Rec), and F1 score. These metrics can be calculated by the number of true positive (TP), false positive (FP), true negative (TN), and false negative (FN) samples as follows:

$$ER = \frac{FP + FN}{TN + FP + FN + TP} \tag{4}$$

$$FAR = \frac{FP}{TN + FP} \tag{5}$$

$$Prec = \frac{TP}{FP + TP} \tag{6}$$

$$Rec = \frac{TP}{FN + TP} \tag{7}$$

$$F1 = \frac{2 \times Prec \times Rec}{Prec + Rec}. \tag{8}$$

5.5. Detection Performance

5.5.1. Experiment Settings

This subsection shows that the performance of an IDS of a specific car model can be improved by taking advantage of other car models' datasets by the proposed system. In detail, we set up with three settings including:

1. Local models are trained on only the local data of each car model itself. Each model is trained at most 50 epochs, and early stopping is used to prevent overfitting.
2. Federated learning model (Fed) is the model trained on multiple datasets by the proposed federated learning with the FedAvg algorithm. The federated learning runs through 50 rounds in total, whereas each local model runs 15 epochs for client updates.
3. Fine-tuned federated learning model (Fed-FN) is the federated learning model, which is trained through about 10 epochs independently on different CAN datasets of different car models.

These models were trained, tuned, and tested with the number samples of data as shown in Table 3. During the training process, the stochastic gradient descent with momentum is used for optimization. We set the learning rate for local models as 0.05, and for federated learning as 0.005. After getting a model from a setting described above, the model runs through all test sets of the car models to produce local results, which are then averaged to obtain the final results for comparison.

5.5.2. Analysis of Results

As illustrated in Table 4, the F1 scores of Fed-FN models are much higher than those of the local, at 0.9893 on average. In general, Fed-FN increases the recall scores significantly for all examined car models. It should be noticed that the recall metric is more important than the precision in the case of an intrusion detection system, as the consequence of false negative cases are worse than that of false positive cases. What stands out in the result is that transfer learning from the global model increases the F1 score of the BMW car's local model by approximately 1.0, which helps the model satisfy the accuracy constraint for deployment. The primary reason for the suboptimal performance of the local model in BMW cars is their unique electronic architecture, wherein data generation usually exhibits a proportional relationship between the number of ECUs and CAN IDs. In contrast to other car models, BMW has a higher number of CAN IDs and lower data generation frequency shown in Table 1. This indicates a low command frequency and a high command interval time, which could contribute to the less responsive system and the inferior local model performance of BMW compared to other car models. Though modifying federated deep learning hyperparameters can improve BMW's accuracy, it may lead to complexity. Using a global model instead of customizing hyperparameters can improve accuracy without adding complexity. Sharing ideas between local and global models can improve accuracy rates. In addition, the results of the Fed model are shown to demonstrate the essentials of the fine-tuning step. We can see that the Fed model produces the lowest F1 score due to the effect of data heterogeneity. Furthermore, the proposed system decreases the ER and FAR of local models significantly, especially in the case of the BMW (see Figures 7 and 8).

Table 4. Personalized FL compared to other baselines.

Car Brand	Model	Prec	Rec	F1
Kia	Local Model	0.9996	0.9836	0.9915
	Fed	0.5782	0.9723	0.7252
	Fed-FN	**0.9992**	**0.9912**	**0.9952**
Tesla	Local Model	0.9913	0.9655	0.9782
	Fed	0.7014	0.9462	0.8057
	Fed-FN	0.**9915**	**0.9928**	**0.9922**
BMW	Local Model	0.8453	0.9041	0.8737
	Fed	0.5266	0.5897	0.5565
	Fed-FN	**0.9907**	**0.9706**	**0.9806**
Average	Local Model	0.9454	0.9511	0.9478
	Fed	0.6021	0.8361	0.6958
	Fed-FN	**0.9938**	**0.9849**	**0.9893**

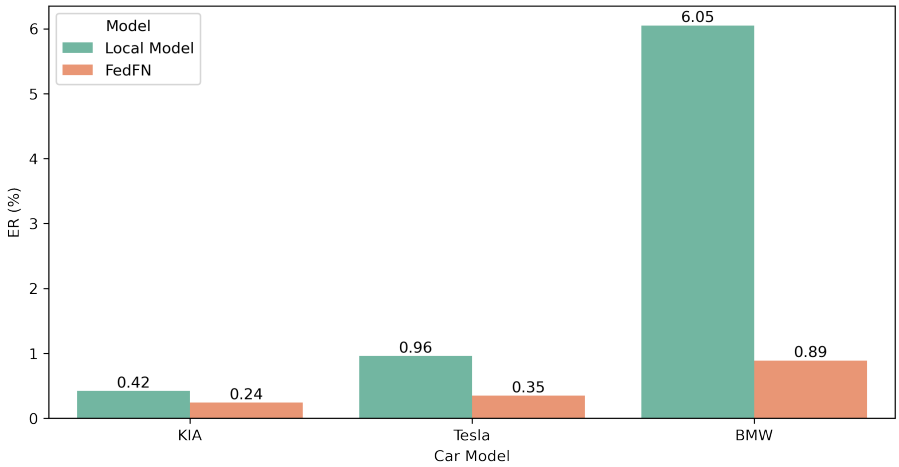


Figure 7. The comparison of ER (%) between local models and Fed-FN by different car models.

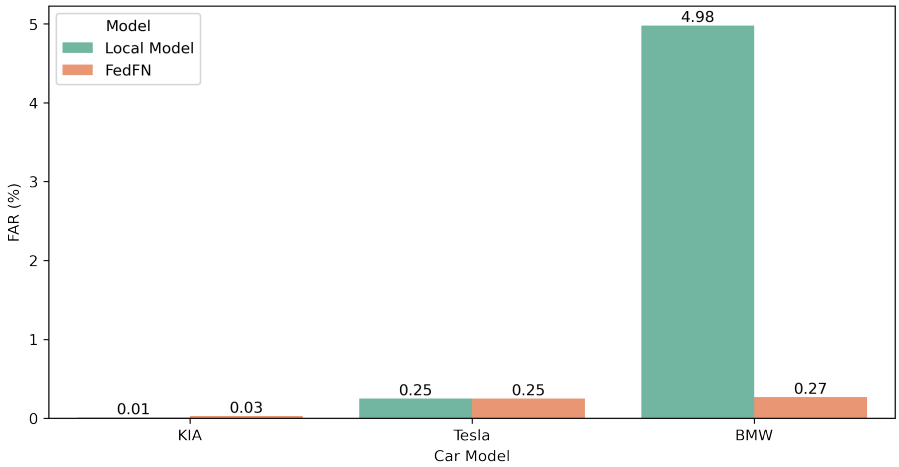


Figure 8. The comparison of FAR (%) between local models and Fed-FN by different car models.

5.6. The Effect of Different Data Sizes

5.6.1. Experiment Settings

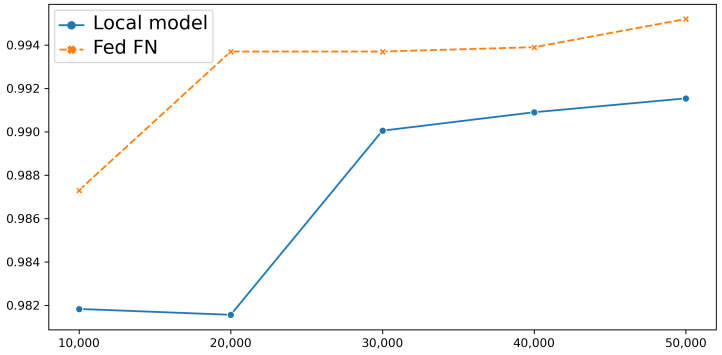
In this subsection, we aim to compare the performance of the local models and the proposed fine-tuned federated models of three car models (KIA, Tesla, BMW) on different training data sizes. Table 5 illustrates the data distribution of different datasets having the total samples from 10k to 50k. The training settings are the same as in the previous section.

Table 5. The details of the dataset for testing the effect of different data sizes.

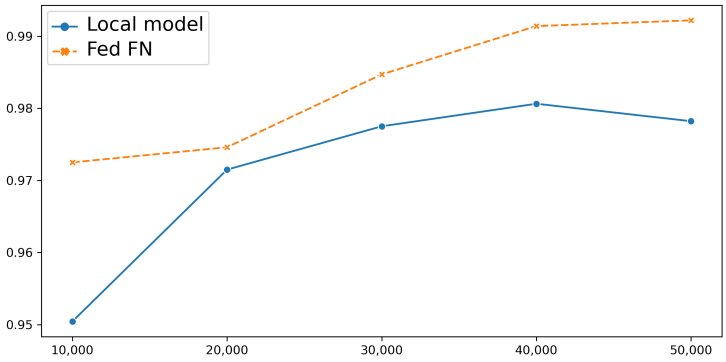
Setting	Label	Kia	BMW	Tesla
10k	Normal Attack	7488 2510	7304 2694	7430 2568
20k	Normal Attack	14,494 5302	14,000 5998	15,892 4106
30k	Normal Attack	21,417 8582	21,859 8140	21,840 8158
40k	Normal Attack	29,300 10,698	30,237 8068	28,208 8806
50k	Normal Attack	37,206 12,792	39,237 10,388	38,292 11,706

5.6.2. Analysis of Results

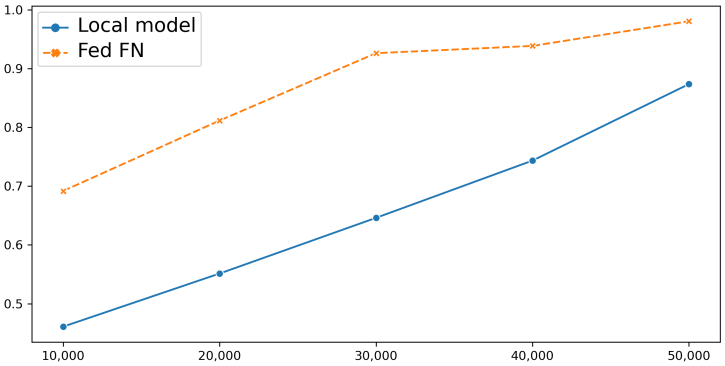
Figure 9 shows that the gap between the F1 scores of these models can be reduced when the data size goes up. This implies the advantage of a personalized FL model when training on a limited dataset. Notably, in the case of 10k training samples of the BMW model, the local model achieves the lowest F1 score, lower than 0.5, which is increased by 20% when the proposed model is utilized. In summary, the IDS models can also achieve an F1 greater than 90% with at least 30k training samples for each car model, as shown in Figure 9.



(**a**)



(**b**)



(**c**)

Figure 9. The F1 scores of different models trained on various number of training sample settings. (**a**) KIA results. (**b**) Tesla results. (**c**) BMW results.

5.7. The Effect of Global Training Participation on the Individual Performance

5.7.1. Experiment Settings

In a personalized FL scheme, a new client can request to obtain the global model trained on other clients' data and utilize it for its own training. This discourages clients

to join the global learning process due to the time and economic costs of global training participation. As a result, the diversity of the global feature representation cannot be fully exploited, as few participants do not want to contribute their models. The purpose of this experiment is to answer whether a participant takes benefits from global training. The BMW model was chosen for this experiment as its data are hard to predict among the examined car models. With 30k training samples, we fine-tuned the BMW model based on two federated models, not including BMW's contributions. In addition, we compare the results with the local model without exploiting the federated model.

5.7.2. Analysis of Results

As shown in Table 6, BMW's model can take benefit from the global model even if it did not actually participate in the global training. Thanks to the meaningful representation provided by the global model, the ER and FAR of the local model decreased four and three times, respectively. Furthermore, these results keep decreasing to approximately 2% of ER and 1% of FAR when BMW contributes its data diversity to the global. Finally, the F1 score of the participation case can reach roughly 95%, which is higher by 5% compared to that of the nonparticipation one. All of the results draw a conclusion that the personalized FL scheme encourages every client to attend the global training, as each client can not only contribute to the community objective, but also improve its own detection performance.

Table 6. The effect of global training participation in case of the BMW model.

	ER	FAR	Prec	Rec	F1
Local model	0.1638	0.1066	0.6461	0.6461	0.6461
Without participation	0.0459	0.0347	0.8882	0.9170	0.9024
With participation	0.0230	0.0128	0.9571	0.943	0.9499

5.8. Computational Analysis

After training the local models with global features, they can be deployed in real-world applications. We tested the computational efficiency of the proposed model by deploying it on an NVIDIA Jetson AGX Xavier, which can be integrated into a vehicle's XPU for detecting intrusions in the CAN bus [30]. This device is equipped with a 512-core Volta GPU with Tensor Cores, an 8-core ARM CPU, and 32 GB of RAM. Our tests revealed that the proposed model takes approximately 5.96 milliseconds to detect a frame containing 29 messages, and can process up to approximately 4800 messages per second. Given that there are around 2000 CAN messages on the bus [18], our model is capable of handling the workload in a real ECU.

6. Conclusions

Our study presents a novel framework for training a global IDS in a distributed manner that addresses the difficulty of training a deep learning-based IDS model when data are limited. In addition, the proposed approach uses a federated learning scheme, where participants can jointly train a global model without sharing their confidential datasets, thereby ensuring data privacy. To our best knowledge, this is the first study proving that an IDS model of a specific car can be improved by learning from other CAN datasets, demonstrating the transfer learning from global latent features which are extracted from a model trained in a federated manner. The proposed model is trained and tested on a real dataset consisting of three different car models: KIA, Tesla, and BMW with different settings. The experimental results show that the proposed model increases the average of the F1 score by 4%, compared to local models. According to the results from experiments on various training data sizes, the proposed model requires at least 30k training samples to achieve the F1 score of greater than 90% for all car models. Federated learning will be a promising direction, especially for vehicle securities. Our study, along with most other research in this field, has utilized deep learning models as a black box, making it difficult

to interpret the results. For example, we may not understand why the model is producing poor results on BMW cars. However, we recognize that Explainable AI (XAI) can be a powerful tool for addressing this challenge. Therefore, as part of our future work, we plan to integrate XAI with our federated learning scheme to provide more insight into the decision-making process of the model and enable us to better understand the factors that contribute to its performance.

Author Contributions: Conceptualization, T.-N.H.; data curation, M.R.I.; methodology, software, investigation, writing—original draft preparation, T.-N.H. and M.R.I.; resources, writing—review and editing, supervision, K.Y. and D.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (No. 2021R1A4A2001810), by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. 2022-0-01197, Convergence security core talent training business (SoonChunHyang University)), by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-004), and this work was supported by the Soonchunhyang Research Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Miller, C.; Valasek, C. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* **2015**, *2015*, 1–91.
2. *BOSCH CAN Specification Version 2.0*; BOSCH: Gerlingen, Germany, 1991.
3. Ashraf, J.; Bakhshi, A.D.; Moustafa, N.; Khurshid, H.; Javed, A.; Beheshti, A. Novel Deep Learning-Enabled LSTM Autoencoder Architecture for Discovering Anomalous Events From Intelligent Transportation Systems. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4507–4518. [CrossRef]
4. Barletta, V.S.; Caivano, D.; Nannavecchia, A.; Scalera, M. Intrusion Detection for in-Vehicle Communication Networks: An Unsupervised Kohonen SOM Approach. *Future Internet* **2020**, *12*, 119. [CrossRef]
5. Binbusayyis, A.; Vaiyapuri, T. Unsupervised deep learning approach for network intrusion detection combining convolutional autoencoder and one-class SVM. *Appl. Intell.* **2021**, *51*, 7094–7108. [CrossRef]
6. Hoang, T.N.; Kim, D. Detecting in-vehicle intrusion via semi-supervised learning-based convolutional adversarial autoencoders. *Veh. Commun.* **2022**, *38*, 100520. [CrossRef]
7. Hoang, T.N.; Kim, D. Supervised Contrastive ResNet and Transfer Learning for the In-vehicle Intrusion Detection System. *arXiv* **2022**, arXiv:2207.10814. https://doi.org/10.48550/arXiv.2207.10814.
8. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1798–1828. [CrossRef]
9. An, Y.; Park, J.; Oh, I.; Kim, M.; Yim, K. Design and Implementation of a Novel Testbed for Automotive Security Analysis. In Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Lodz, Poland, 1–3 July 2020. [CrossRef]
10. A Study on the Implementation and Analysis Method of the Connected Car Accident Scenario Model (KISA-WP-2018-002).
11. Cheah, M.; Bryans, J.; Fowler, D.S.; Shaikh, S.A. Threat intelligence for bluetooth-enabled systems with automotive applications: an empirical study. In Proceedings of the 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Denver, CO, USA, 26–29 June 2017; pp. 36–43. [CrossRef]
12. Nie, S.; Liu, L.; Du, Y. Free-fall: Hacking tesla from wireless to can bus. *Brief. Black Hat USA* **2017**, *25*, 1–16.
13. Oruganti, P.S.; Appel, M.; Ahmed, Q. Hardware-in-loop based automotive embedded systems cybersecurity evaluation testbed. In Proceedings of the Proceedings of the ACM Workshop on Automotive Cybersecurity, Richardson, TX, USA, 27 March 2019; pp. 41–44. [CrossRef]
14. Petit, J.; Stottelaar, B.; Feiri, M.; Kargl, F. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe* **2015**, *11*, 995.
15. Rouf, I.; Miller, R.; Mustafa, H.; Taylor, T.; Oh, S.; Xu, W.; Gruteser, M.; Trappe, W.; Seskar, I. Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study. In Proceedings of the 19th USENIX Security Symposium (USENIX Security 10), Washington, DC, USA, 11–13 August 2010.

16. Jo, H.J.; Choi, W. A Survey of Attacks on Controller Area Networks and Corresponding Countermeasures. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 6123–6141. [CrossRef]
17. Wang, C.; Zhao, Z.; Gong, L.; Zhu, L.; Liu, Z.; Cheng, X. A Distributed Anomaly Detection System for In-Vehicle Network Using HTM. *IEEE Access* **2018**, *6*, 9091–9098. [CrossRef]
18. Song, H.M.; Woo, J.; Kim, H.K. In-vehicle network intrusion detection using deep convolutional neural network. *Veh. Commun.* **2020**, *21*, 100198. [CrossRef]
19. Hanselmann, M.; Strauss, T.; Dormann, K.; Ulmer, H. CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data. *IEEE Access* **2020**, *8*, 58194–58205. [CrossRef]
20. Derhab, A.; Belaoued, M.; Mohiuddin, I.; Kurniawan, F.; Khan, M.K. Histogram-Based Intrusion Detection and Filtering Framework for Secure and Safe In-Vehicle Networks. *IEEE Trans. Intell. Transp. Syst.* **2021**, *23*, 2366–2379. [CrossRef]
21. Desta, A.K.; Ohira, S.; Arai, I.; Fujikawa, K. Rec-CNN: In-vehicle networks intrusion detection using convolutional neural networks trained on recurrence plots. *Veh. Commun.* **2022**, *35*, 100470. [CrossRef]
22. Nam, M.; Park, S.; Kim, D.S. Intrusion Detection Method Using Bi-Directional GPT for in-Vehicle Controller Area Networks. *IEEE Access* **2021**, *9*, 124931–124944. [CrossRef]
23. Zhang, L.; Ma, D. A Hybrid Approach toward Efficient and Accurate Intrusion Detection for In-Vehicle Networks. *IEEE Access* **2022**, *10*, 10852–10866. [CrossRef]
24. Murvay, P.S.; Groza, B. Source Identification Using Signal Characteristics in Controller Area Networks. *IEEE Signal Process. Lett.* **2014**, *21*, 395–399. [CrossRef]
25. Lee, H.; Jeong, S.H.; Kim, H.K. OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame. In Proceedings of the 2017 15th Annual Conference on Privacy, Security and Trust (PST), Calgary, AB, Canada, 28–30 August 2017; pp. 57–5709. [CrossRef]
26. Markovitz, M.; Wool, A. Field classification, modeling and anomaly detection in unknown CAN bus networks. *Veh. Commun.* **2017**, *9*, 43–52. .: 10.1016/j.vehcom.2017.02.005. [CrossRef]
27. Choi, W.; Joo, K.; Jo, H.J.; Park, M.C.; Lee, D.H. VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2114–2129. [CrossRef]
28. Zhu, K.; Chen, Z.; Peng, Y.; Zhang, L. Mobile Edge Assisted Literal Multi-Dimensional Anomaly Detection of In-Vehicle Network Using LSTM. *IEEE Trans. Veh. Technol.* **2019**, *68*, 4275–4284. [CrossRef]
29. Zhang, L.; Yan, X.; Ma, D. A Binarized Neural Network Approach to Accelerate in-Vehicle Network Intrusion Detection. *IEEE Access* **2022**, *10*, 123505–123520. [CrossRef]
30. Sun, H.; Chen, M.; Weng, J.; Liu, Z.; Geng, G. Anomaly Detection for In-Vehicle Network Using CNN-LSTM with Attention Mechanism. *IEEE Trans. Veh. Technol.* **2021**, *70*, 10880–10893. [CrossRef]
31. Al-Jarrah, O.Y.; El Haloui, K.; Dianati, M.; Maple, C. A Novel Intrusion Detection Method for Intra-Vehicle Networks Using Recurrence Plots and Neural Networks. *IEEE Open J. Veh. Technol.* **2023**, *4*, 271–280. [CrossRef]
32. Hussain, S.; Ali Imran, M.; Yang, J.; Hu, J.; Yu, T. Federated AI-Enabled In-Vehicle Network Intrusion Detection for Internet of Vehicles. *Electronics* **2022**, *11*, 3658. [CrossRef]
33. Aliyu, I.; Feliciano, M.C.; Van Engelenburg, S.; Kim, D.O.; Lim, C.G. A Blockchain-Based Federated Forest for SDN-Enabled In-Vehicle Network Intrusion Detection System. *IEEE Access* **2021**, *9*, 102593–102608. [CrossRef]
34. Kang, L.; Shen, H. A Transfer Learning based Abnormal CAN Bus Message Detection System. In Proceedings of the 2021 IEEE 18th International Conference on Mobile Ad Hoc and Smart Systems, MASS 2021, Denver, CO, USA, 4–7 October 2021; pp. 545–553. [CrossRef]
35. Brendan McMahan, H.; Moore, E.; Ramage, D.; Hampson, S.; Agüera y Arcas, B. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, Fort Lauderdale, FL, USA, 20–22 April 2016. [CrossRef]
36. Collins, L.; Hassani, H.; Mokhtari, A.; Shakkottai, S. FedAvg with Fine Tuning: Local Updates Lead to Representation Learning. *arXiv* **2022**, arXiv:2205.13692
37. Li, X.; Hu, Z.; Xu, M.; Wang, Y.; Ma, J. Transfer learning based intrusion detection scheme for Internet of vehicles. *Inf. Sci.* **2021**, *547*, 119–135. [CrossRef]
38. Koh, Y.; Kim, S.; Kim, Y.; Oh, I.; Yim, K. Efficient CAN dataset collection method for accurate security threat analysis on vehicle internal network. In Proceedings of the International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Palermo, Italy, 4–6 July 2022; pp. 97–107. [CrossRef]
39. Lee, H.; Choi, K.; Chung, K.; Kim, J.; Yim, K. Fuzzing can packets into automobiles. In Proceedings of the 2015 IEEE 29th International Conference on Advanced Information Networking and Applications, Gwangju, Republic of Korea, 24–27 March 2015; pp. 817–821. [CrossRef]
40. Hoppe, T.; Kiltz, S.; Lang, A.; Dittmann, J. Exemplary Automotive Attack Scenarios: Trojan horses for Electronic Throttle Control System (ETC) and replay attacks on the power window system. *VDI BERICHTE* **2007**, *2016*, 165.
41. Liu, J.; Zhang, S.; Sun, W.; Shi, Y. In-vehicle network attacks and countermeasures: Challenges and future directions. *IEEE Netw.* **2017**, *31*, 50–58. [CrossRef]

42. Islam, M.R.; Oh, I.; Yim, K. CANTool An In-Vehicle Network Data Analyzer. In Proceedings of the 2022 International Conference on Information Technology Systems and Innovation (ICITSI), Bandung, Indonesia, 8–9 November 2022; pp. 252–257. [CrossRef]
43. Beutel, D.J.; Topal, T.; Mathur, A.; Qiu, X.; Fernandez-Marques, J.; Gao, Y.; Sani, L.; Li, K.H.; Parcollet, T.; de Gusmão, P.P.B.; et al. Flower: A Friendly Federated Learning Research Framework. *arXiv* **2020**. arXiv:2007.14390.