IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier

A Distributed Aggregation Approach for Vehicular Federated Learning

LUCAS PACHECO[1,2], TORSTEN BRAUN[1], DENIS ROSÁRIO[2], ANTONIO DI MAIO[1], AND EDUARDO CERQUEIRA[2]

[1]University of Bern, Switzerland
[2]Federal University of Pará, Brazil

Corresponding author: Lucas Pacheco (e-mail: lucas.pacheco@unibe.ch).

ABSTRACT Federated Learning (FL) has rapidly become a crucial paradigm for training Machine Learning (ML) models when datasets are spread across several devices without compromising the privacy of the data owners. In vehicular networks, FL can be used to train driving models and object detection and classification over sensitive datasets to continuously improve user experience and driving safety. However, the majority of FL implementations cannot efficiently filter malicious vehicular users and low-quality contributions. This article proposes Distributed OT-based Federated Learning (DOTFL), an aggregation mechanism based on the clustering of the received trained Neural Networks Neural Network (NN) at the vehicular devices and on outlier detection. The proposed mechanism can detect malicious contributions by comparing them to previously received contributions and following a clustering approach. Furthermore, the convergence time of the FL process is improved by distributing trained NN weights directly through vehicle-to-vehicle links. Experimental analysis shows an improvement of up to 22% in terms of accuracy compared to state-of-the-art FL approaches. This is achieved by using clustering models and removing outliers, enabling a significantly lower presence of malicious contributions in aggregated models.

INDEX TERMS Federated Vehicular Networks, Robust Model Aggregation, Privacy Preservation, Vehicular Ad hoc Networks

I. INTRODUCTION

Big data and Deep Learning (DL) are transforming our daily lives in the rapidly evolving digital technology landscape. By leveraging ML models, these technologies provide intelligent services across diverse domains, including Intelligent Transportation Systems (ITS), entertainment, and personalized services [1]. For example, Vehicular Ad-Hoc Network (VANET) is expected to benefit significantly from integrating intelligence into various vehicular systems, such as driving assistance, recommendations, and in-vehicle entertainment. The potential impacts on road safety, traffic efficiency, and driving experience are substantial [2].

Continuous learning plays an essential role in achieving a high level of intelligence in autonomous vehicle networks [3].

In the context of autonomous vehicle networks, one crucial enabler is Federated Learning (FL). Federated Learning over Vehicular Network (FVN) opens up possibilities for enhancing vehicular intelligence while addressing concerns about data privacy [4]–[6]. FVN can enable vehicles to collectively learn from diverse scenarios and environmental situations, such as urban traffic patterns, highway driving, and adverse weather conditions. By sharing knowledge gained from traversing these different situations, the connected vehicles can comprehensively understand the road environment and make better decisions. Moreover, FVN can facilitate the development of advanced driver assistance systems (ADAS), where vehicles can learn and adapt to the driving behaviors of individual drivers, providing personalized assistance and improving overall road safety. Additionally, FVN can be utilized in traffic management systems, allowing traffic lights and road infrastructure to optimize their operation by learning from real-time data on traffic flow and congestion patterns, ultimately leading to smoother and more efficient traffic control.

The vehicular network characteristics, such as very dynamic topology and mobility can impose heavy constraints in the FVN training process, such as the need for a long enough link duration with attachment points (i.e., infrastructure and other vehicles) limits user participation in the FL process [7], [8]. The growing complexity and number of parameters of the Machine Learning (ML) models to be transmitted over the

network causes the number of participating vehicular users to decrease [9]. In addition, traditional implementations of FVN often consider a single aggregation server to receive, aggregate, and distribute models, which restricts the effective utilization of network resources.

In this context, FVNs can leverage Vehicle-To-Vehicle (V2V) links for distributed aggregation. By allowing models to be received via V2V communication, the efficiency of the FVN process can be improved, and a distributed approach can be enabled, leading to improved model convergence times. Instead of relying solely on a centralized server, the distributed aggregation approach requires executing certain FVN functions within vehicles, such as model aggregation and verification of received model contributions [10]. Moreover, this approach facilitates asynchronous sharing of ML models, allowing trained NNs to be aggregated directly at the vehicle without central server intervention for synchronization. Vehicles can opportunistically share models within a communication window, eliminating the need for coordination from a central server at specific times. To achieve this, FVN must leverage the trained models' mobility prediction, channel characteristics, and statistical features.

Furthermore, ML models' accuracy and convergence time are highly impacted by the presence of non-Independent and Identically Distributed (IID) datasets and even poisoning attacks from malicious vehicular users who may share incorrect parameters [11]. Such attacks tackle the aggregation phase of the FVN, in which a server merges the trained weights of the received contributions, potentially including low-quality weights, which decreases the prediction accuracy of the aggregated model. While filtering low-quality contributions is essential, it remains a computationally challenging task [12]. One way to optimize the accuracy of FVN models is by clustering vehicular user models based on the statistical features of their datasets. However, integrating new knowledge and data into trained models demands more communication and training rounds, which further complicates the process [13]. To the best of our knowledge, the issue of designing a distributed aggregation in FVN scenarios with malicious vehicular users attempting model poisoning attacks remains a challenge.

This article introduces a distributed aggregation approach for FVN combined with an Optimal Transport (OT) clustering algorithm to verify the quality of received models at the vehicles to improve communication efficiency while ensuring robustness against poisoning attacks, called Distributed OT-based Federated Learning (DOTFL). In its operation, DOTFL considers V2V links to opportunistically disseminate newly trained ML models in a decentralized way with improved communication efficiency. In this sense, vehicular users can train their models and distribute them opportunistically to neighboring users without intervention from the central FVN server at the edge layer. In addition, DOTFL form clusters of models according to the similarity calculation to not aggregate malicious models and outliers, ensuring robustness against poisoning attacks. Experimental results demonstrate significant improvements in both the convergence time and accuracy of the models. For instance, DOTFL provides an improvement of up to 22% in terms of model accuracy and rejects the majority of malicious users from the FL aggregation when compared to traditional centralized aggregation and D2D aggregation with model clustering.

The main contributions of this article are three-fold: First, it introduces the design and implementation of a distributed aggregation approach for FVN; Second, this article proposes an NN- based clustering technique to determine the distance between trained NN models, which enhances the accuracy of aggregated models and identifies outliers; Finally, it presents an exponential smoothing-based aggregation technique to support the integration of new knowledge in the network.

The remainder of this article is organized as follows: Section II presents the state-of-the-art in FVN. Section III details the architecture and operation of DOTFL. Section IV describes the simulation setup, discusses the experimental results, and provides insight into the observed data behavior. Finally, Section VI concludes the article and discusses potential future directions.

II. RELATED WORK

This section reviews the state-of-the-art in FL and its implementation on a VANET and mobile network.

A. COMMUNICATION EFFICIENCY AND MODEL AGGREGATION IN FEDERATED LEARNING

Selecting which users participate in the FL process is one of the significant challenges in FL, as not all vehicular users in the network have the computing and resources to participate in the process. Chen *et al.* [16] tackles the issue through probabilistic device selection for aggregation to minimize the latency introduced by the transfer of the ML parameters during the aggregation process. While the proposed approach can minimize the convergence time for the model parameters, it also excludes vehicular users with unstable connections from the FL process. However, the approach can still select users with low-quality datasets or malicious vehicular users.

FL under V2V communications has been modeled and studied to some extent by Xing *et al.* [17], which offers an alternative to traditional centralized topology FL. However, the majority of works that consider V2V FL cannot be directly applied to dynamic network conditions, such as in the case of VANETs, with its shorter communication windows and high rates of packet loss. The introduction of V2V aggregation in the FL process is also studied by Hosseinalipour *et al.* [18], who introduced a semi-decentralized process, in which users exchange model parameters locally until a consensus is formed. However, they do not consider the case in which users send malicious updates or the high mobility of VANETs.

TABLE 1: Summary of the state-of-the-art Federated Learning techniques for vehicular networking

Work	Year	V2V/D2D Aggregation	User Clustering	Mobility-Aware
Pervej *et al.* [14]	2023	✓	x	✓
Taik *et al.* [15]	2022	x	✓	x
Chen *et al.* [16]	2021	x	x	x
Xing *et al.* [17]	2020	✓	x	x
Hosseinalipour *et al.* [18]	2020	✓	x	x
Kong *et al.* [19]	2021	x	x	✓
Liu *et al.* [20]	2021	x	x	x
Kornblith *et al.* [21]	2019	x	✓	x
Alvarez-Melis *et al.* [22]	2020	x	✓	x
Li *et al.* [23]	2021	x	✓	x
Zhang *et al.* [24]	2021	x	✓	x
DOTFL	2023	✓	✓	✓

B. FEDERATED LEARNING OVER VANETS

Kong *et al.* [19] proposed a vehicular fog-based aggregation scheme for supporting robust user privacy against inference attacks and flexible participant joining and leaving. While the proposed scheme achieves low complexity for participants leaving and joining the network, the robustness of the models over non-IID datasets has not been studied. Furthermore, the geographical distribution of datasets across the scenarios can be challenging for disseminating knowledge across different vehicles. Liu *et al.* [20] propose a blockchain-based intrusion detection model for vehicular networks. While FL presents a good alternative for building robust intrusion detection ML models, it must also provide robustness against poisoning attacks with low complexity computations, which can be performed within the short communication windows of the scenarios. The blockchain stores and distributes models in a trustworthy manner; it can incur a higher computational cost and delay in aggregating and validating model contributions, particularly in dynamic environments, such as VANETs.

Pervej *et al.* [14] present a vehicular edge FL solution, aiming to leverage onboard central processing units and local datasets of highly mobile connected vehicles for training a global model. This work aligns with the about selecting appropriate vehicular users for the FL process based on their computational resources and connectivity [16]. However, while Chen *et. al* propose a probabilistic device selection to minimize latency, the paper does not explicitly address how the high mobility and potential unstable connections of vehicular users are handled during the training process [16]. Furthermore, there is no clear mechanism discussed for filtering out malicious or low-quality datasets, which remains a crucial challenge in FVN as outlined in the provided related works section.

Taik *et al.* [15] address the non-IID data challenge by proposing a new architecture for vehicular FL, aimed at improving learning accuracy under mobility constraints through clustering. This work is by the works of Liu *et al.* [25] and Wang *et al.* [26] on clustering in vehicular FL to improve model performance and communication efficiency. However, while the clustering approach is an important aspect, the paper does not describe in depth how the clustered vehicular FL architecture would cope with malicious updates or high mobility in VANETs, which are significant concerns in the scenario being studied. Moreover, the paper does not make clear how the clustering process performs in real-time or dynamic network conditions, which could potentially affect the accuracy and robustness of the aggregated models in vehicular FL.

C. FEDERATED LEARNING OVER NON-IID DATA

Non-IID data over model contributions can be addressed by measuring the similarity between trained NN representations, as shown in Kornblith *et al.* [21]. However, given the variability of NN architectures for different prediction tasks, finding a general metric for NN similarity remains a challenge [27]. Considering the similarity of raw datasets, Alvarez-Melis *et al.* [22] propose the usage of Optimal Transport as a robust similarity metric for finding IID datasets by considering the similarity of the probability distributions of features and labels present in the dataset of user devices. Li *et al.* [23] tackled the challenge of aggregating non-IID models in FL through contrastive model aggregation. However, estimating model similarity in a federated manner poses challenges for effectively identifying and clustering models at a low computational cost. Zhang *et al.* [24] introduced a reputation-based incentive for user participation in FL, as users with higher reputation scores can obtain higher monetary incentives for their contributions. However, the quality of models trained over FL depends on the quality of the datasets at the participating devices. It can be affected by malfunctioning sensors, incorrect data labeling, or the presence of non-IID datasets within participating devices.

D. CLUSTERING IN VEHICULAR FEDERATED LEARNING

Clustering algorithms play a vital role in FL by grouping participants with similar data distributions or learning objectives, thus improving model performance. In recent years, numerous studies have explored various clustering algorithms for FL, including k-means, Hierarchical Clustering (HC), Density-based Spatial Clustering Of Applications With Noise (DBSCAN), and spectral clustering. Liu *et al.* [25] proposed

a K-means clustering-based algorithm to group vehicles with similar learning objectives and data distributions, resulting in reduced communication overhead and improved model performance. Wang *et al.* [26] applied *DBSCAN* to identify spatially dense regions where vehicles share similar data, enhancing communication efficiency and model performance. However, K-means clustering and *DBSCAN* have limitations, such as the need for pre-determined cluster numbers and the inability to handle non-convex clusters.

Hierarchical clustering is a promising approach for FVN as it builds a tree-like structure of nested clusters, allowing vehicles to participate in FL at different levels of granularity. Li *et al.* [28] proposed a hierarchical clustering-based algorithm that adapts to dynamic network topologies, enabling vehicles to join or leave clusters without affecting the overall structure, facilitating seamless communication and learning. Samarakoon *et al.* [29] presented a multiresolution learning approach based on hierarchical clustering, where vehicles can learn models at different levels of detail, allowing for flexible trade-offs between communication efficiency and model performance. Yan *et al.* [30] developed a scalable hierarchical clustering algorithm that enables local aggregation within subclusters before global aggregation, reducing communication overhead and latency. These studies demonstrate the potential of hierarchical clustering in enhancing the efficiency and performance of Vehicular FL.

E. CONTINUOUS LEARNING

Several approaches have been proposed in the literature to address the continual learning problem, including regularization-based and rehearsal-based methods. While DOTFL shares some similarities with existing continual learning approaches, such as the use of clustering algorithms to group data sources and the need to avoid catastrophic forgetting, it also has several unique features tailored to the FL process [31], [32]. Recent studies have also explored the use of FL for continual learning, with some proposing hybrid approaches that combine centralized and FL [33]. However, these approaches often require centralized storage of the model parameters, which can raise privacy concerns.

F. SUMMARY

Table 1 summarizes the main characteristics of the analyzed FL works in the context of their support for distributed aggregation using V2V, the presence of an efficient malicious user filtering scheme to protect the models from poisoning attacks, as well as the use of contextual and mobility information of nodes to assist in the distribution and aggregation of ML models in the network. While many existing works tackle current issues in FL with custom model aggregation and reputation management, performing these tasks in a scalable manner remains an open issue. To the best of our knowledge, no other work in the state-of-the-art has proposed a comprehensive algorithm to enable distributed FVN while providing sufficient filtering of malicious vehicular users in the context of FVNs. We can see that among the listed works, DOTFL is the only one that combines these characteristics into a single algorithm for training FVN models in a decentralized and robust against-attacks manner.

III. Distributed OT-based Federated Learning

TABLE 2: Table of Symbols

Symbol	Description
	Scenario Parameters
N	Number of vehicles
C	Number of base stations and edge servers
D_i	Dataset of the i-th user
J	Number of computed clusters in H_i
SD	Symmetrical difference $\triangle$
	Federated Learning Parameters
H_i	Set of contributions for the i-th user
k	Dataset sample without label
ζ	Fraction of malicious users in the network
	CNN Model Hyperparameters
A	Neural network architecture
S_M	Size of neural network in bits
W	Neural network weights
L	Number of dense layers in A
κ	Size of the kernels in the CNN
θ	number of neurons in a dense layer
δ	Dropout rate
$l(W, D)$	Loss functions over the weights W and dataset D
$\mu(a)$	Aggregation smoothing function
	Distance-metric Symbols
M_i	Model/user distance matrix generated by NSIM on the vehicle i
$\phi(\psi, \nu)$	EMD distance between distributions ψ and ν
$Z(\psi, \nu)$	Set of all possible couplings between distributions ψ and ν
γ	Coupling between two distributions
$d(x, y)$	Distance between points x and y
K_k	Kalman Gain
w_t	Noise factor in the Kalman gain computation
	Networking Parameters
Θ	Real transfer size between vehicles
b	Estimated transfer size between vehicles
η	Inefficiency of wireless encoding
τ	Interval between training rounds
a	Smoothing factor of μ
	KF Symbols
x_t	Neighboring vehicle's position at time t
A	Factor relating two consecutive measurements
w_t	Noise factor
e_k	Error estimation between previous state x_k and predicted state $\hat{x}_k$
$\hat{x}_k^-$	Estimated next state
K_k	Kalman gain
P	Error covariance matrix
H	Matrix relating state to measured variable
z_k	Measurement
R	Measurement noise covariance matrix
	Channel Symbols
$d(t)$	Relative distance between u and k as a function of time
W	Channel data rate
$\Gamma(d(t))$	Spectral efficiency of the channel
$\text{SNR}(d(t))$	Signal-to-Noise Ratio in the communication link between u and k
U_c	Signaling overhead
$\Theta(d(t))$	Instantaneous throughput

This section introduces a distributed aggregation approach for FVN combined with an OT clustering algorithm to verify the quality of received NN models at the vehicles to improve communication efficiency while ensuring robustness against poisoning attacks.

A. SYSTEM MODEL

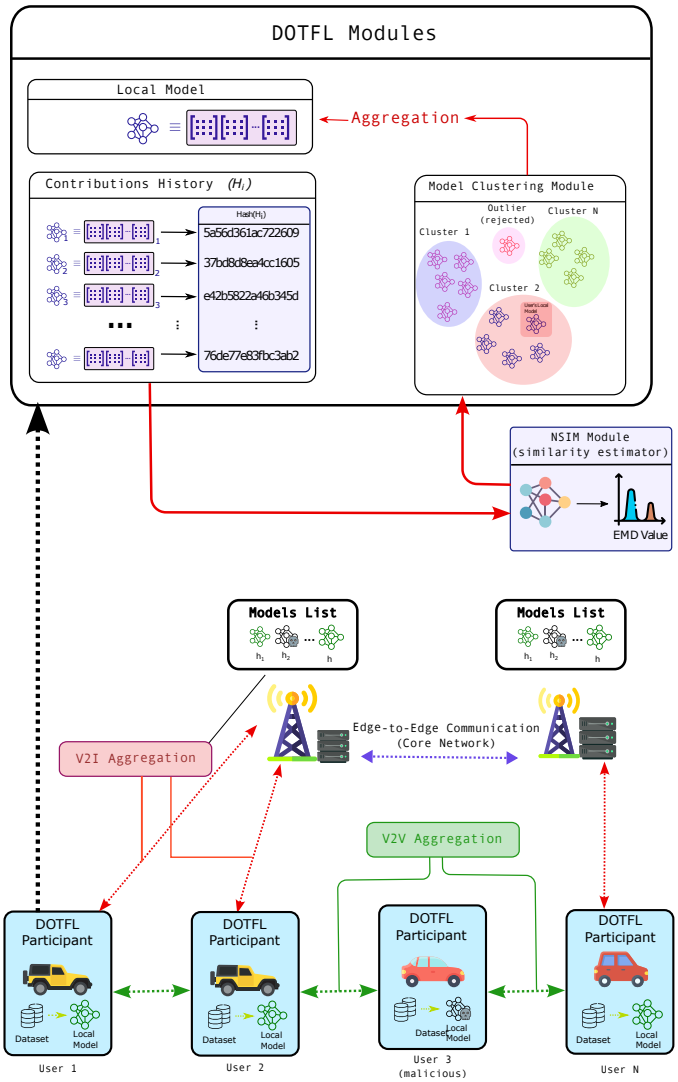


FIGURE 1: Vehicular Federated Learning Scenario.

Figure 1 shows a FVN scenario where a set of vehicles possesses local datasets from which they can train NN models according to the FL optimization objective. In this context, vehicles can communicate with the fixed network infrastructure (*i.e.*Vehicle-To-Infrastructure (V2I)) and directly with other vehicles (*i.e.*V2V) to propagate their trained NNs. Vehicles opportunistically send trained models via V2V or V2I when being inside the coverage areas of other vehicles or base stations. Thus, the models can be aggregated in vehicles or edge servers based on the quality and duration of their communication links. Figure 1 shows the DOTFL modules, considering a local dataset of previously received models for dissemination, a clustering module based on the model similarity estimation, and an aggregation module that generates the local aggregated model for predictions. The DOTFL modules are present in all nodes of the network (vehicles and edge servers). A mobility prediction module is responsible for estimating the contact time between vehicles to assist in the transfer of models. The presence of malicious vehicular users disseminating incorrect weights is possible in the scenario (center vehicle). In addition, malicious vehicular users could disseminate incorrect weights, which may decrease the accuracy of aggregated models and increase the time for convergence in FL.

We consider a vehicular networking scenario (e.g, VANETs, connected vehicles, autonomous vehicles, etc) with N mobile vehicles $u_i \in \{u_1, \ldots, u_N\}$, where each vehicle has local dataset $D_i \in \{D_1, \ldots, D_N\}$. Each dataset D_i contains a set of *features* $x_{k,i}$, with $k \in \{1, \ldots, \|D_i\|\}$, each associated with a *label* $y_{k,i}$. The scenario also contains C base stations $\{c_1, c_2, ..., c_C\}$, located at arbitrary positions, that can communicate with a set $E = \{e_1, e_2, ..., e_C\}$ of C edge servers through the core network and with vehicles through their communication interfaces (e.g., Dedicated Short Range Communications (DSRC) and 4G/5G). In this scenario, vehicles can communicate through direct V2V links, but we assume they cannot directly access each other's datasets to guarantee privacy. Each vehicle u_i in the system locally trains a model architecture A to obtain the NN model weights W_i that minimize a loss function l on its local dataset D_i, as shown in Equation (1). The *local loss* $l(W_i, D_i)$ is defined as the average loss, as the prediction error, across all predictions for the dataset D_i using the weights W_i.

$$l(W_i, D_i) = \frac{1}{\|D_i\|} \sum_{k=1}^{\|D_i\|} f(W_i, x_{k,i}, y_{k,i}) \tag{1}$$

We assume that edge servers take care of system initialization and provide every vehicle with the NN architecture A, consisting of the NN hyperparameters and loss function via base stations. Edge servers also provide computation support for training by disseminating partially trained models to accelerate convergence. The goal of the FL process is to compute the set $W^* = \{W_1, \ldots, W_N\}$ of weights that minimize the *global average loss function* [17] l_s, formulated in Equation (2), over a series of model aggregations, where the global loss $l_s(W^*)$ is defined as the average of the local loss across users with their local weights and local datasets.

$$l_s(W^*) = \frac{1}{N} \sum_{i=1}^{N} l(W_i, D_i) \tag{2}$$

Furthermore, we consider that a portion of the users may be malicious regarding their contributions to the model. The weights W_i from non-malicious vehicles in $\mathcal{N}$ are distributed according to a distribution P, i.e., $W_i \sim P$, for $u_i \in \mathcal{N}$. However, the weights W_j from malicious vehicles in $\mathcal{M}$ are distributed according to a different distribution Q, i.e., $W_j \sim Q$, for $u_j \in \mathcal{M}$, where $P \neq Q$. Thus, the participation of such users may compromise the convergence of the ML models as the malicious weights may compromise prediction accuracy for non-malicious users' datasets.

B. ALGORITHM DESCRIPTION

Let us define M_i as the distance matrix generated by a trained Neural Network Similarity Estimator (NSIM) [27] on vehicle i, which contains the distance values between models.

Let us define the *contribution history* H_i as a bounded-capacity FIFO queue, in which each element is a set of NN weights, trained and stored on node i, called *model contribution*. We now model a local instance of the FL model for the vehicular user u_i as a 4-tuple (A, D_i, M_i, H_i), where D_i is the i-th node's local dataset, and A is the global NN architecture, received from the network.

We consider that the most recent model contributions from other vehicular users or the FL server are stored in the user's device and fed into the NSIM module to compute the model distance. NSIM computes the distance matrix of the vehicle user's contribution history based on the NN weights contained within the contributions. After the distance matrix M_i has been computed, it is fed into a Hierarchical Clustering algorithm [34], pre-configured by the network at system startup. Each contribution H_i is assigned a label i_l, l being the internal cluster number of a given contribution assigned at the user device. Given that the user's trained contributions have a label i', the aggregation is performed over contributions with label i_l, such that $i_l = i'$. In this context, each vehicle computes its FL model by aggregating contributions in their contribution history, which are IID with the users' dataset.

The model weights define the user's local updates after a round of training over their local datasets. Based on the model's prediction accuracy, a loss function, described in Equation (1), must be computed and minimized for the FL process to converge with a minimum accuracy value across users. The weights computed by the vehicular user are then committed to the contributions history H_i with a hash computed from the trained weights, which can uniquely identify the computed model at the corresponding iteration.

C. COMMUNICATION AND CONTACT ESTIMATION

Let us define $S_M \in \mathbb{N}$ as the size of the model weights in bits and $\|H_i\| \in \mathbb{N}$ as the number of models exchanged from the user's contribution history. The total amount of data b_i to transfer the whole contribution history H_i from the vehicular user i to another vehicular user is given by Equation (3), where η represents the inefficiency of the encoding (*i.e.*, the difference between the actual transmission size and the minimum size given by the entropy) [35].

$$b_i = \eta S_M \|H_i\| \tag{3}$$

In this context, the uplink and downlink transfers are not necessarily symmetric. Users may have more unique contributions history than their counterparts in the communication round. The compression scheme's efficiency for transferring weights will also impact the number of bits transmitted.

To estimate the mobility of a neighboring vehicle, we consider data collected by the vehicle's sensors, which includes direction and velocity data. This information is communicated through beacons, enabling a vehicle to monitor its environment. We use a Kalman Filter (KF) on board the vehicle to estimate the future positions of its neighboring vehicles. This information is important to decide if a vehicle will (or not) exchange models with a specific neighbor. The KF is an integral part of the contact estimation process. This article assumes KF as a mobility prediction mechanism, but other approaches could be used, such as the work by Emami *et al.* [36].

The position of a neighboring vehicle at any given moment (t) is modeled as a point x_t, which represents the state of the system - the position of the vehicle. The term A is a scaling factor that helps convert the previous state into the current state, indicating how the system changes from one moment to the next. Noise in the system, accounting for uncertainties in our model, is denoted by w_t. Estimation error, represented by e_k, is the difference between the previous actual state x_k and the predicted state $\hat{x}_k$. This error is used to predict the next state, denoted as $\hat{x}_k^-$.

As per the KF, the current state x_t is a linear combination of the previous state x_{t-1} and a noise-adjusted correction term w_{t-1}, as shown in Equation 4. The predicted state at any moment t is thus formed by combining historical measurements. The difference or discrepancy between historical data points and their corresponding predictions can be calculated using Equation 5. This helps to quantify the estimation error at each step. The KF also calculates a value known as the Kalman gain, denoted as K_k. This is done by considering the error covariance matrix P, which describes the uncertainty of our state estimate, and matrix H, which is the observation model that relates the state of the system to the measurements we have. The calculation of Kalman gain is outlined in Equation 7. This gain value essentially provides a weightage determining how much importance should be given to the new measurement versus the previous estimate.

$$x_t = Ax_{t-1} + w_{t-1} \tag{4}$$

$$e_k = x_k - \hat{x}_k \tag{5}$$

$$\hat{x}_k = \hat{x}_k^- + K\left(z_k - H\hat{x}_k^-\right) \tag{6}$$

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{7}$$

Based on the predicted positions of neighboring vehicles, a vehicular user estimates the data transfer capacity within the communication window when the vehicles are within communication range. In the system, we model the communication capacity between two nodes u and k (e.g., a pair of vehicular users with a V2V link or a vehicular user and a base station). We assume available channel fading statistics for the scenario. We consider the mobility of k predicted by u and the relative distance between u and k as a function of time $d(t)$. Assuming a channel data rate W, we calculate the spectral efficiency of the channel as Γ and

the Signal-to-Noise Ratio (SNR)$(d(t))$ in the communication link between u and k. We calculate the spectral efficiency of the transmission within the communication window using Equation (8) [37].

We define the spectral efficiency of the communication channel between nodes u and k as $\Gamma(d(t))$, representing the maximum achievable data rate in bits per second per Hertz (bps/Hz), considering available channel fading statistics. To calculate the spectral efficiency, we integrate the probability that the logarithm of the SNR is more significant than a threshold z, integrating from 0 to infinity. The instantaneous throughput $\Theta(d(t))$ represents the data transmitted per unit time. It depends on the relative distance $d(t)$ between nodes u and k, which varies over time due to the mobility of k predicted by u. Assuming a channel data rate W (bps), we express the throughput as $W \cdot \Gamma(d(t)) \cdot (1 - U_c)$. Here, U_c denotes the signaling overhead, accounting for additional data exchanged during communication for control purposes. To estimate the total data exchanged over the link within a given communication window, we integrate the instantaneous throughput $\Theta(d(t))$ for time t over the interval from t_0 to t_1. This calculation yields the total data throughput in bits for the specified duration, as shown in Equation (9).

$$\Gamma(d(t)) = \int_0^{\infty} \mathbb{P}(log_2(1 + SNR(d(t)) > z))dz \tag{8}$$

$$\int_{t_0}^{t_1} \Theta(d(t))dt = \int_{t_0}^{t_1} W \cdot \Gamma(d(t)) \cdot (1 - U_c)dt \tag{9}$$

The estimated communication capability during the contact window is then compared to the bits b_i necessary for the transfer user i's contribution history of H_i over the wireless channel. When the necessary number of bits for the transfer is superior to the communication capability, the sender i excludes some contributions to make the transaction size smaller (*truncated transfer*). After the vehicle has selected the contributions to send to its neighbor, models are bundled together, compressed, and quantized for transmission. Note that compression within a single cluster may achieve high compression rates, as the model weights tend to share similar features, increasing the redundancy of the cluster.

D. NSIM AND MODEL CLUSTERING

DOTFL calculates the pairwise similarity between the model contributions trained by individual users. In this context, models trained by a given vehicular user encode the statistical features of the user's underlying dataset for training. We compare the probability distributions of users' datasets and, based on the pairwise values $d_{ij} \in \mathbb{R}_+$ denoting the distance between the probability distributions of two trained models i and j, a distance matrix $M = (d_{ij}) \in \mathbb{R}_+^{N \times N}$ is computed for N vehicular users.

Consider two trained ML models with their respective NN weights. Let us define NSIM, a special NN that accepts as input the weights W_i and W_j of two distinct NNs and outputs an estimation of their separation within the entire possible space of NN weights.

Existing research illustrates that NNs trained with varying random initializations on similar datasets result in equal weights. Techniques such as kernel-based metrics can be used to identify such similarities [21]. In this study, we adopt the OT theory to calculate the Earth Mover's Distance (EMD) distance between two datasets, considered as the optimal transformation from one feature distribution in a dataset to another [22].

The EMD represents a numerical measure of the distance between the probability distributions of the datasets' features owned by two vehicular users. The strength of the EMD lies in its ability to calculate the similarity between distributions in a comprehensive manner, which can be generalized for different data types with minimal adjustments. We compute the EMD between two models based on the trained features in their NN weights without knowledge of the vehicle's dataset.

Let us define a distance metric $d(x, y)$ between any two points $(x, y) \in K^2$. Furthermore, let us denote $Z(\psi, \nu)$ as the set of couplings between the distributions of weights in trained models ψ and ν, defined over the domain K. A *coupling* $\gamma(x, y) \in Z(\psi, \nu)$ describes the mass transferred from point $x \in K$ to point $y \in K$.

The p-Wasserstein distance, denoted as $\phi_p(\psi, \nu)$, between the two distributions ψ and ν represents the displacement for mapping distribution ψ onto distribution ν with minimum cost, as depicted in Equation (10). This metric provides a solid basis for understanding the degree of similarity between two different ML models.

$$\phi_p(\psi, \nu) := \left(\inf_{\gamma \in Z(\psi,\nu)} \int_{K^2} d(x, y)^p \, \mathrm{d}\gamma(x, y) \right)^{1/p} \tag{10}$$

For our case, which considers probability distributions, the EMD distance $\phi(\psi, \nu)$ is equivalent to the 1-Wasserstein distance, which can be expressed as in Equation (11).

$$\phi(\psi, \nu) := \inf_{\gamma \in Z(\psi,\nu)} \int_{K^2} d(x, y) \mathrm{d}\gamma(x, y) \tag{11}$$

If $\psi : K \to [0, 1]$ and $\nu : K \to [0, 1]$ are two single-dimensional discrete probability mass functions over finite support $\{1, \ldots, \omega\} = K \subset \mathbb{N}$, the coupling γ is a bivariate joint probability mass function that can be represented as a two-dimensional matrix $\gamma \in \Gamma = [0, 1]^{\omega \times \omega}$. In this case, the EMD distance $\phi(\psi, \nu)$ between ψ and ν is the minimum of the utility function in the constrained minimization problem 12.

$$\begin{aligned}
\phi(\psi, \nu) := \min_{\gamma \in \Gamma} \quad & \langle \gamma, d \rangle_F && && \text{(12a)} \\
\text{s.t.} \quad & \gamma \mathbf{1} = \psi, && && \text{(12b)} \\
& \gamma^\intercal \mathbf{1} = \nu, && && \text{(12c)} \\
& \gamma_{ij} \geq 0, && \forall i, j \in K && \text{(12d)}
\end{aligned}$$

Here $\langle \cdot, \cdot \rangle_F$ is the Frobenius inner product between two matrices so that $\langle \gamma, d \rangle_F = \sum_{(i,j) \in K^2} \gamma_{ij} d_{ij}$, and $d = (d_{ij}), (i,j) \in K^2$ is the matrix of distances between i and j. A common choice for d is the squared Euclidean distance, where $d_{ij} = \sqrt{(i-j)^2}$, but any distance notion can be applied. The constraints in Equation (12b) and Equation (12c) impose that the marginalizations of the coupling γ are equal to ψ and ν, respectively. Constraint (12d) guarantees that all entries of the coupling are positive, as they represent probabilities.

Since no raw data from vehicular users is available for the computation at the edge servers, we assume that a central server has a reference distribution consisting of data samples and labels. A series n' of datasets are built based on the data samples at the server $\{Ds_1, Ds_2, ..., Ds_{n'}\}$ and are used to train n' machine learning models, such that the weights of the trained models are collected to build a training dataset for Neural Network Similarity Estimator (NSIM). NSIM is then trained over the corresponding data points consisting of the NN weights trained for the i-th layer being considered and the computed EMD value and used for predicting the EMD value given only the trained NN.

Figure 2 shows how EMD can be calculated even between different data types, as it compares the probability densities of label distributions in the vehicular user datasets. The distance matrices show the pairwise EMD values between twenty sample users in the network storage. Note that the distance matrix must follow certain constraints, such as being symmetric and having a zero-valued main diagonal, as a given user's distance to themselves must be zero. Such constraints are also applied to the predicted matrix generated by NSIM based on the trained ML models computed for the twenty users. This provides redundancy in the calculation and enables more robust distance estimation by NSIM. In other words, users with similar data samples in their datasets should be attributed a high similarity score by NSIM. The predicted similarities are fed into a hierarchical clustering algorithm chosen for its ability to discover meaningful structures and relationships within vehicular user datasets. This approach enables us to detect outliers and group similar models, facilitating the identification of potential malicious users and enhancing the aggregation process. In this context, vehicular users with malicious models are expected to have a significant distance value from all other vehicular users in the network. They are not included in the clusters used for aggregation.

E. MODEL AGGREGATION AND PARTICIPATION INCENTIVE

DOTFL considers an asynchronously aggregation of FVN models. Thus, after a given vehicular user u_i has received and trained the ML model over their local dataset, the model is included in H_i.

Upon contact, a pair of vehicular users u_i and u_k advertise a list of the contributions in H_i and H_k as a list of hashes computed for each model in the form $\texttt{hash}(h), \forall h \in H$. The hashes are calculated to be advertised for other vehicular users without transferring the complete trained models. We consider the symmetric difference between the advertised hash lists, denoted by $\triangle$, as the contributions present in H_i and not present in H_k as $SD_{i,k} \triangleq H_i \triangle H_k$, and vice versa.

Upon receiving the trained models from u_k, h_{ki}, NSIM computes the pairwise similarity between the received and pre-existing models in H_n. The contributions history is then updated to include the newly received models, becoming $\{H_n \cup SD_{n,k}\}$. The Hierarchical Clustering module of DOTFL assigns a cluster label to all contributions in H_n. Contributions not clustered with others are considered outliers and discarded before aggregation. Considering that J clusters have been formed, we consider the contributions trained by u_n and the models with the same cluster label for an IID set of contributions.

We define two models of aggregation: 1) Aggregating only over the same cluster labels as the user's model; 2) Aggregating over all valid (i.e., non-outliers) models in H with decreasing weights for non-IID contributions. Each user builds an aggregated model used for prediction. The aggregated model is computed as the weighted sum of the user's IID cluster contributions in which more recently trained models are given a higher weight than older ones. However, we consider users to sort the contributions based on cluster membership and the timestamp of the model creation if the information is available.

Models are aggregated according to an exponential smoothing factor $\mu = \mu_1, \mu_2, ..., \mu_J$, as defined in Equation (13). The sum of the weight vector μ is scaled by a factor $s \in [0, 1]$, which dictates the weight of the contributions history compared to the previous state of the model, similar to the learning rate in traditional FL. Furthermore, the network pre-configures the smoothing factor $a \in [0, 1]$ to define how fast the weights of older contributions decrease as new models are introduced.

Equation (14) shows how model contributions are aggregated via the FedAvg algorithm, where the model weights $W_{i,t}$ are updated with a factor of the average between all received contributions at the t-th round of communication. However, in DOTFL, we consider the aggregation to happen locally at user devices and only happen over a subset of all received model contributions. This is necessary as vehicular users are only expected to trust some received model contributions from other users.

$$\mu_i = \frac{a(1-a)^i}{s \cdot \sum_{j=0}^{J} a(1-a)^j} \tag{13}$$

$$W_{i,t+1} = (1-\mu_i) W_{i,t} + \mu_i \frac{\sum_{j=0}^{N} W_{j,t}}{N} \tag{14}$$

The model contributions in H_i are sorted by their cluster distance to the user's cluster and aggregated with the previous state of the model as shown in Equation (15). The current state of the local model W_{t+1} consists of a linear combination

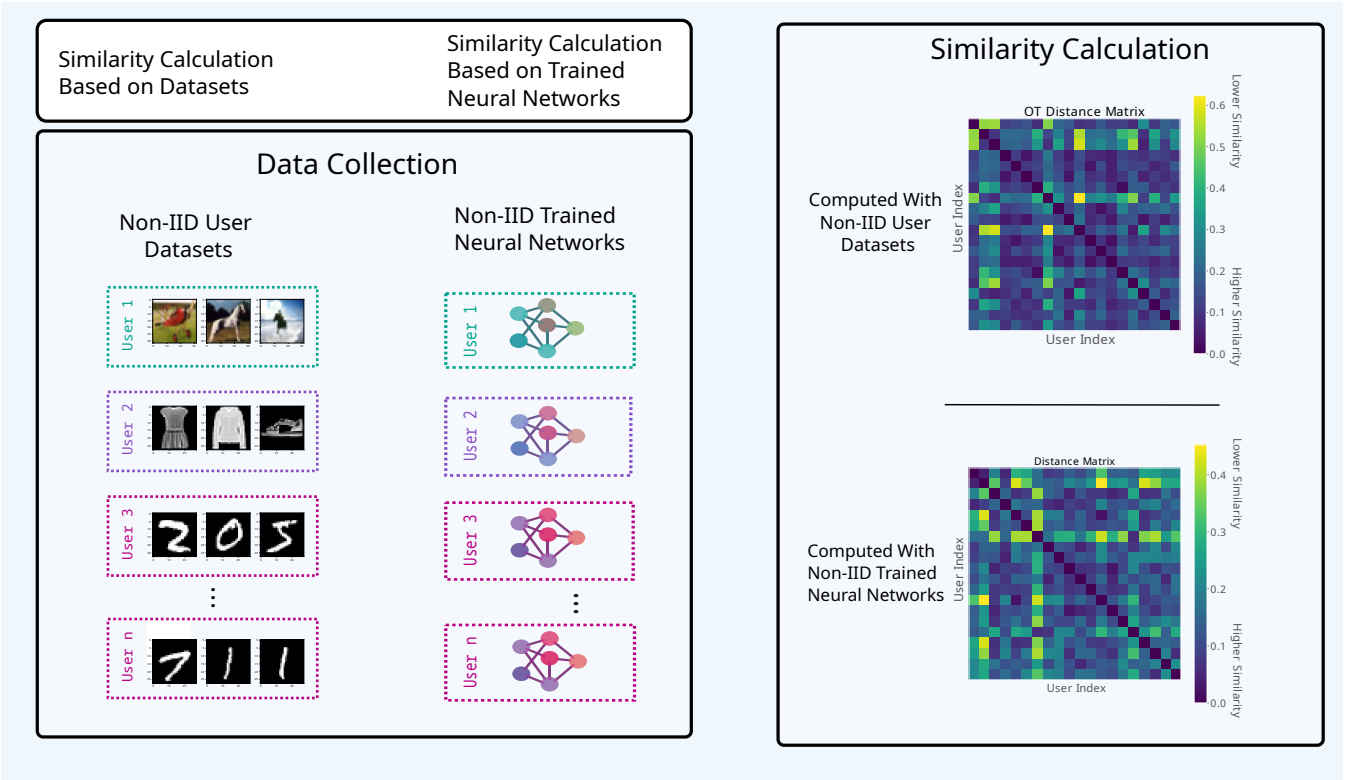


FIGURE 2: Similarity Estimation based on the raw datasets (left) and predicted by NSIM based on trained NNs (right).

of the received contributions and the previous state of the aggregated model.

$$W_{t+1} = W_t + \sum_{i=0}^{J} \mu_i W_{i,t} \tag{15}$$

Algorithm 1 describes the operation of DOTFL as an instance in a vehicular device for learning and distributing FL models. We consider that the edge layer of the network is responsible for distributing the instances and optimizing objectives to all participating vehicles in lines 4 and 5, as participating vehicular users trust the edge layer. Furthermore, at the system setup, the edge layer must also distribute the weights of the NSIM model for the specific learning task, as shown in line 6. The received models are trained over the vehicular user's dataset, shown in line 8. As vehicles move through the scenario, they come within range of other vehicles running a DOTFL instance. Within the communication window between vehicles, the first vehicle that initiates the transmission is responsible for advertising the models contained in its contribution history, as described in lines 9 - 13. Afterward, each vehicle must estimate the amount of data that can be exchanged, considering both mobility and channel characteristics, as shown in lines 14 - 15. In lines 16 - 18, we consider an NN weights compression scheme to bundle the models sent during the transmission based on the DEFLATE algorithm, which combines LZ77 lossless compression and Huffman coding. Both participating vehicles must contribute with the unique model contributions in their storage, as the contributions are disseminated through the network. After transferring all models, each vehicle can proceed to cluster and aggregate the received contributions and discard outlier contributions, as shown in lines 19 - 20.

IV. METHODOLOGY AND EXPERIMENTAL DESIGN

A. SIMULATION SCENARIO

The performance of DOTFL is compared to state-of-the-art techniques through simulated urban scenarios with vehicles performing FL tasks, base stations, and the respective communication links for V2I and V2V model aggregation. For each simulation, $C = 10$ base stations are arbitrarily placed in the scenario, such that all points in the environment are covered by at least one of the base stations. Furthermore, N vehicles are placed in each simulation, with $N \in \{10, 30, 50, 100\}$, each with a maximum speed restriction of $50\,\mathrm{km/h}$.

The mobility of vehicles follows a realistic mobility trace, namely, the Köln Vehicular Mobility Dataset [38], consisting of mobility traces for a large number of vehicles based on real-world mobility measurements from the city of Köln, Germany.

In all scenarios, vehicles can directly communicate with each other and with the base stations to distribute and collect the trained ML models. As expected in 5G scenarios, the training of NNs by vehicles is controlled by the edge computing servers situated at the base stations in terms of

Algorithm 1: Distributed OT-based Federated Learning

Data: Dataset format, optimization objective
Result: Trained FL models
Define optimization objective;
The edge layer computes sample datasets;
NSIM model builds on sample datasets;
for $u \in U$ **do**
  Receive model architecture A from edge server $e \in E$;
  Receive NSIM weights edge server $e \in E$;
  while *Local model not converged* **do**
    Perform local training;
    if *Neighbor FL instance in range* **then**
      k $\leftarrow$ neighbor instance;
      Initiate communication;
      Advertise list of contributions $\texttt{hash}(h) \forall h \in H_u$;
      Compute symmetrical difference $SD \leftarrow H_u \triangle H_k$;
      Predict the next positions for k;
      Calculate total data exchange possible Θ;
      Compress contributions in $SD_{u,k}$;
      Send compressed contributions to k;
      Receive contributions from k, $SD_{k,u}$;
      Cluster received contributions;
      Aggregate over chosen cluster;

architecture and hyper-parameters. Figure 3 shows the architecture of the NN used in the experiments.

B. NEURAL NETWORK ARCHITECTURE

All participants in the FL process must agree on the same neural network architecture to be used. Thus, in our experiments, we chose the MobileNet model [39] as a base network architecture for the majority of experiments, except Figure 8.

The base model, namely MobileNet was pre-trained on the ImageNet dataset [40], and was chosen here due to its fast prediction latency and lower memory footprint, making it an extremely efficient architecture for image classification [41]. In the context of vehicles, such low prediction latency is desirable to make quick and accurate driving decisions.

The model is tailored to fit well the CIFAR-10 [42], the CIFAR100 [43], and MNIST [44] image classification datasets. Notably, an extra hidden layer composed of 256 neurons is inserted after the last layer of the original MobileNet model. This dense layer uses the Rectified Linear Unit (ReLU) activation function to introduce non-linearity, helping the model learn more complex patterns in the CIFAR-10 dataset. To mitigate overfitting, a dropout layer is incorporated after this hidden layer, with a dropout rate of 0.5, randomly freezing the weights of specific neurons during training to reduce overfitting. Following the hidden layer, an output layer consisting of 10 neurons is added. Each neuron corresponds to one of the ten classes in the CIFAR-10 dataset. A softmax activation function is used in this layer, converting the model's outputs into a probability distribution over the ten classes.

In addition to the MobileNet architecture, we also perform experiments considering other NN architectures to verify DOTFL's performance in different scenarios. Thus, experiments were implemented also on the ResNet50 and a plain Convolutional Neural Network (CNN) architecture.

- The ResNet50 is a more complex NN model, consisting of a deep architecture that incorporates residual blocks to alleviate the vanishing gradient problem, thus facilitating the training and generalization of ML tasks [45].
- On the other hand, a simple CNN was designed to serve as a baseline for comparing the performance of more sophisticated models, such as MobileNet and ResNet50. This CNN comprises three convolutional layers with Rectified Linear Unit (ReLU) activations, interspersed with max-pooling layers to reduce spatial dimensions and extract the most significant features.

C. NON-IID DATA DISTRIBUTION

We aim to attest to the performance of the FL process in the presence of non-IID datasets across users. Given the datasets used in the experiment (which are image classification datasets), we devise a method for distributing non-IID local datasets by over-representing certain classes for each user at random.

This is accomplished by selecting one of the dataset classes at random for each user and defining the number of samples the vehicle's local dataset will contain for each of the classes according to a Gaussian distribution centered on the overrepresented class. Thus, we assign a larger proportion of samples from a single class to each user, while still including samples from other classes to maintain diversity. This means that some classes were significantly over-represented in the datasets of certain users compared to others, which reflects real-world scenarios where data can be unevenly distributed across nodes in distributed learning systems. This is especially true in the case of vehicular networks, where users' driving patterns and areas they drive across may significantly change from user to user.

D. SIMULATED ALGORITHMS

For each simulated scenario, the performance of DOTFL is compared against to other state-of-the-art FL algorithms, namely: *(i)* FedAvg, *(ii)* Device-to-Device (D2D) Aggregation [17], and *(iii)* SCAFFOLD [46].

- The FedAvg aggregation mechanism [4] is a centralized approach where vehicular users communicate with the base stations to receive aggregated versions of the ML model. Users subsequently perform additional local training rounds and transmit their models back to the central server through the base stations. Finally, the

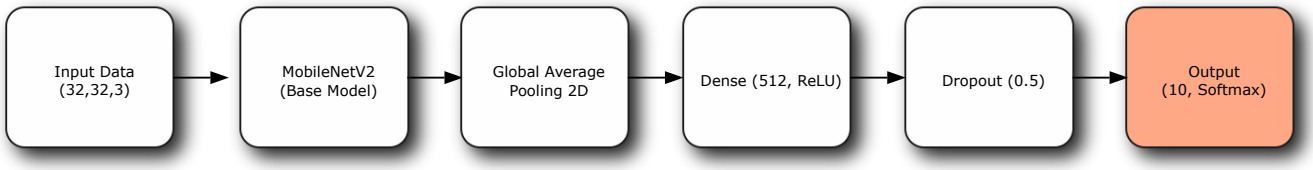


FIGURE 3: Neural Network Architecture

central server aggregates the received models in the corresponding round by averaging the model weights and sends the aggregated model back to mobile users.

- On the other hand, the D2D aggregation approach [17] is a decentralized approach where vehicles receive the initial model hyperparameters configuration from the edge servers via the base stations. However, they distribute their models to other participating vehicles by using direct D2D communication for aggregation. Upon receiving trained models, each vehicle performs a local aggregation round over the model received.
- SCAFFOLD Federated Learning method was also implemented to evaluate its performance within the experimental setups of DOTFL. SCAFFOLD (Stochastic Controlled Averaging Federated Learning) addresses the issue of statistical heterogeneity among client data distributions, which can significantly impede the convergence rate and overall performance of federated learning models [46]. This method introduces a control variate approach to correct the client updates' direction, in terms of their gradients, based on the variance observed across different clients, aiming to reduce the drift caused by non-IID data distributions commonly found in vehicular networks. Thus, it serves as a baseline for the effectiveness of DOTFL's model clustering in the presence of model poisoning attacks and non-IID datasets.

V. PERFORMANCE EVALUATION

The experiments measure the performance of all compared mechanisms through three metrics: (i) the models' *convergence*, (ii) *accuracy scores*, and (iii) the *ratio of malicious vehicular users whose models are rejected*, which takes the number of malicious users introduced in each simulation and scores the ratio of their model contributions which were not used in the aggregation procedure. This can be influenced by failures during the transfer of these models due to lack of sufficient communication time and quality, or, in the case of DOTFL, due to such contributions being rejected at the server. All components of DOTFL were implemented in the Keras [47] and Tensorflow 2.15 [48] frameworks. The implementation source code is made available for download [1]. Table 3 summarizes the simulation parameters.

[1]https://github.com/lsiddd/federated_sid (complete source codes will be made public upon approval.)

Additionally, before deployment to vehicle users, the models underwent pre-training on the specific datasets under evaluation in the experiment. This approach simulates models that vehicle manufacturers could ship, already pre-trained for designated tasks.

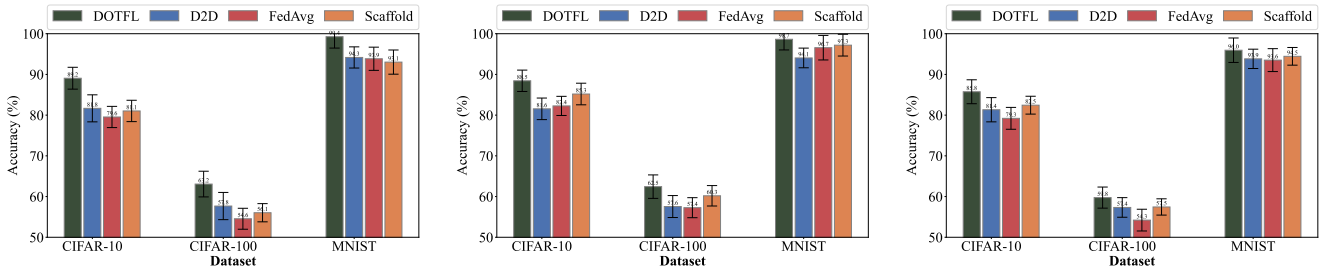
TABLE 3: Simulation Parameters

Parameter	Value
Scenario size	2000x2000 meters
Number of vehicular users N	10, 15, 30, 50, 100
Ratio of malicious vehicular users ζ	0.1, 0.5, 0.7, 0.9
Max. velocity of vehicles	$50\,\mathrm{km/h}$
Number of base stations C	10
Macrocell transmission power	$46\,\mathrm{dBm}$
Small-cell transmission power	$23\,\mathrm{dBm}$
Small-cell height	$10\,\mathrm{m}$
Macrocell height	$45\,\mathrm{m}$
Propagation loss model	Close In
Downlink frequency	$2120\,\mathrm{MHz}$
Uplink frequency	$1930\,\mathrm{MHz}$
CNN size S_M	343 922 parameters
CNN hyperparameters	$\kappa = 5$, $L = 2$, $\theta = (72, 72)$, $\delta_1 = 0.1$, $\delta_2 = 0.1$

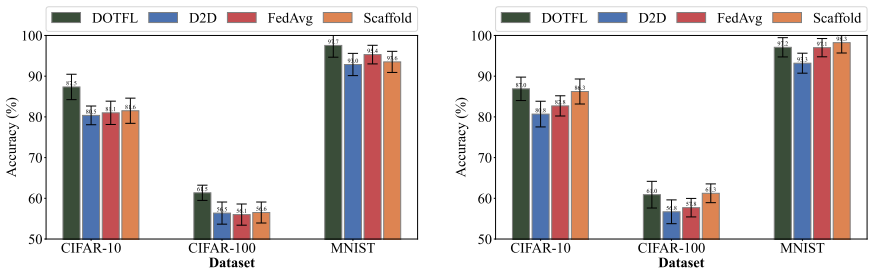
Figure 5 presents the mean accuracy achieved by users across a range of scenarios, characterized by differing numbers of participants (10, 15, 30, 50, and 100), employing four distinct federated learning algorithms: DOTFL, D2D Aggregation, FedAvg, and SCAFFOLD.

Across all evaluated algorithms, DOTFL consistently exhibits a higher average accuracy across the evaluated scenarios. Specifically, DOTFL achieves approximately 5% to 6% higher average accuracy compared to D2D Aggregation, around 2% to 3% higher compared to FedAvg, and closely matches or exceeds the accuracy of SCAFFOLD, depending on the number of vehicles involved. SCAFFOLD's performance stems from its capacity to account for variations in user contributions through a corrective factor, mitigating the impact of model heterogeneity. However, we can still attest to DOTFL's improvement, mainly due to its higher aggregation frequency achieved by D2D aggregation, as well as its ability to filter out malicious users in the clustering process. The findings underscore the importance of implementing model clustering within scenarios characterized by high heterogeneity, particularly when malicious users are present. Furthermore, we observe the behavior of these algorithms concerning the simulated datasets and the number of users included in each simulation in Figure 4.

The convergence plots displayed in Figure 6 provide in-



(a) Accuracy for 10 Users (b) Accuracy for 15 Users (c) Accuracy for 30 Users



(d) Accuracy for 50 Users (e) Accuracy for 100 Users

FIGURE 4: Accuracy comparison of DrivePFL, D2D, and FedAvg across varying numbers of vehicles.

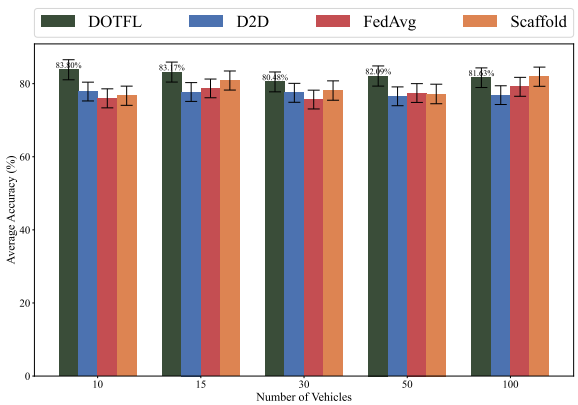
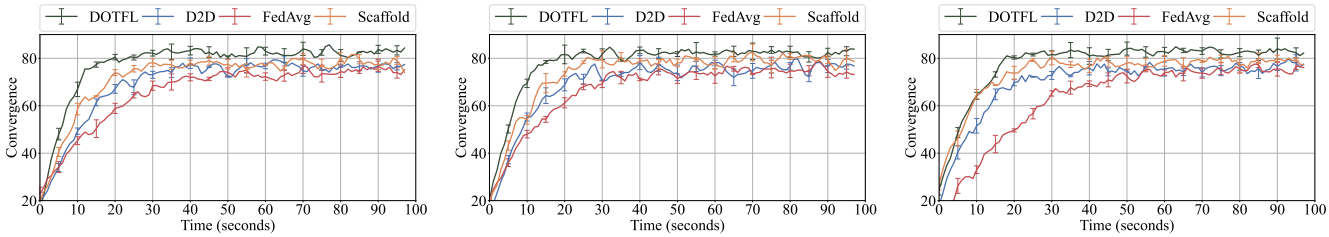


FIGURE 5: Average Accuracy of Users in Scenarios with 10, 15, 30, 50, and 100 users, with 30% of malicious users.

sights into model convergence in different scenarios containing varying numbers of users (10, 30, 50, and 100) and three FL algorithms: D2D Aggregation, DOTFL, FedAvg, and SCAFFOLD. We can observe that DOTFL is the algorithm with the highest convergence speed and accuracy. This can be attributed to its utilization of D2D FL aggregation and the integration of clustering techniques to mitigate model poisoning attacks, as well as increasing aggregation frequency. By leveraging D2D FL aggregation, DOTFL benefits from the collaborative learning capabilities of nearby devices and more efficient distribution of models compared to FedAvg. However, as we can note in the D2D Aggregation case, the presence of malicious users in a D2D setting can significantly compromise the convergence and performance of the model, as malicious users can deliver low-quality weights. The behavior of SCAFFOLD also exhibits better aggregation speed across the scenarios than both FedAvg and D2D. However, its accuracy is also impacted by the presence of malicious users in the network, maintaining converge speeds consistently below DOTFL. This is mitigated in DOTFL using the NSIM similarity estimator and model clustering, as malicious users can be more effectively detected and rejected.

Figure 7 shows simulation scenarios with varying ratios of malicious users to assess the resilience of the evaluated FL algorithms, namely, DOTFL, D2D Aggregation, FedAvg, and SCAFFOLD, against model poisoning attacks. We can observe in the results that DOTFL achieves a superior ability to reject malicious contributions. The ratio of malicious contributions rejected by DOTFL is significantly higher than that observed in FedAvg, D2D Aggregation, and SCAFFOLD, with rejection rates of 24.7% on CIFAR-10, 32.0% on CIFAR-100, and 35.1% on MNIST, surpassing the performance of other algorithms across all evaluated datasets.

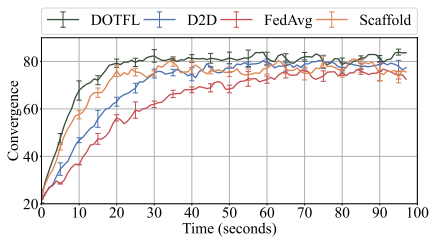
In the context of mitigating malicious contributions, D2D Aggregation and FedAvg demonstrate reduced robustness, featuring significantly lower rejection rates compared to those achieved by DOTFL. While SCAFFOLD enhances the handling of non-IID data and better aligns client updates with the global model, its strategies are not explicitly tailored for identifying and mitigating model poisoning attacks as effectively as DOTFL's model clustering and NSIM similarity estimator techniques. These results highlight the critical need for robust defense mechanisms against model poisoning attacks in FL algorithms, especially for applications vul-



(a) Model Convergence for Scenarios with 10 Users

(b) Model Convergence for Scenarios with 30 Users

(c) Model Convergence for Scenarios with 50 Users



(d) Model Convergence for Scenarios with 100 Users

FIGURE 6: Model convergence for scenarios with 10, 15, 30, 50, 100 users, with 30% malicious users.

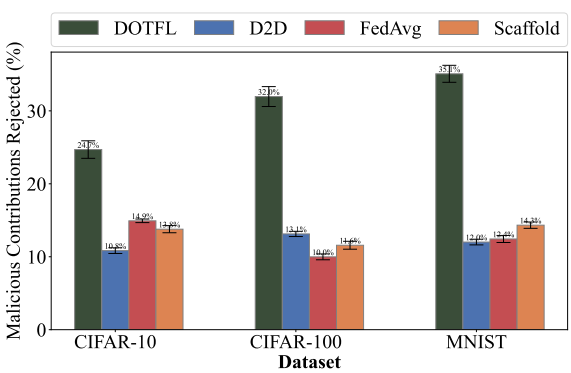


FIGURE 7: Ratio of contributions from malicious users rejected from aggregation.

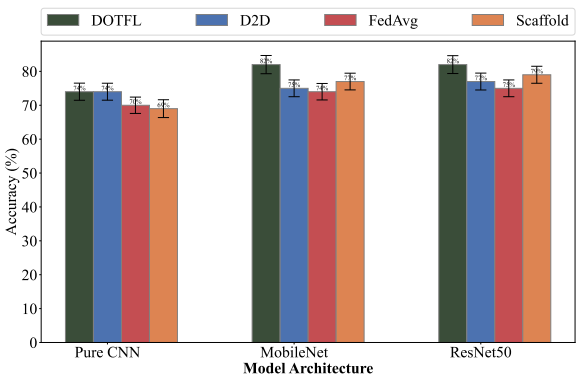


FIGURE 8: Average Accuracy Values for different neural network architectures used as base layers.

nerable to significant rates of malicious interference. The comparative robustness of DOTFL against such threats, as evidenced in our experiments, showcases the potential for collaborative learning in adversarial contexts.

In our simulations, we evaluated the impact of different neural network architectures on the FL process, focusing on the first layers of the ML model. The tested architectures included a traditional CNN model, MobileNet, and ResNet50, each distinct in their design principles and suitability for various use cases. These differences crucially influenced their effectiveness within FL environments.

The architecture's complexity and efficiency are necessary considerations, especially in the case of more critical applications, such as those in vehicular networks. DOTFL demonstrated superior accuracy results across all architectures. It achieved a baseline accuracy of 74% with the simpler Pure CNN, and around 82% when employing either MobileNet or ResNet50. MobileNet and ResNet50 have notably better capability in extracting features from the data samples.

The performance improvements observed in D2D, FedAvg, and SCAFFOLD were more modest. The adoption of more complex architectures yielded moderate benefits for these algorithms. Although FedAvg and D2D exhibited some performance increase when transitioning from a Pure CNN to MobileNet and ResNet-50 architectures, these gains were not statistically significant.

SCAFFOLD's performance demonstrably improved when employing MobileNet and ResNet-50 architectures. This suggests a potential dependence of its corrective mechanism on more complex architectures for effectual mitigation of

client drift and heterogeneity in FL. Our findings demonstrate a substantial influence of neural network architecture on the performance of FL algorithms. Models with enhanced capacity for feature extraction and the ability to learn and abstract more intricate features within the datasets exhibited superior performance.

The performance on the evaluated datasets exhibited minimal variation between ResNet-50 and MobileNet architectures. The results suggest both models possess sufficient learning capacity for the FL algorithms employed. Consequently, the prior prioritization of MobileNet in vehicular contexts might be particularly advantageous due to its potential to reduce latency in prediction, a crucial factor influencing driver experience and safety.

VI. CONCLUSIONS

Federated Learning over Vehicular Network is a rapidly growing ML paradigm for training high accuracy ML models over highly distributed datasets without compromising the privacy of participating vehicular users in the VANET. However, the dynamic nature of VANET and the lack of a centralized manner to establish the trustworthiness of all participants can leave FVNs vulnerable to low-quality contributions, as well as model poisoning attacks.

This article proposes DOTFL (Distributed OT-based Federated Learning), an FVN algorithm based on D2D FL which uses a novel model similarity metric, as well as clustering of similar models to adapt the system to the presence of non-IID models in the scenario, as well as the presence of malicious users trying to poison the model.

Experimental results show the effectiveness of DOTFL in terms of convergence time, accuracy, and the ratio of correctly rejected malicious vehicular users across different scenarios. DOTFL outperforms the existing methods in prediction accuracy by up to 22%. Furthermore, DOTFL rejects the participation of most malicious vehicular users in the network by clustering received models and removing outliers from the clustering process. This shows that DOTFL's usage of a D2D FL in tandem with the NSIM model similarity estimator and the clustering of ML models can significantly improve the performance and resilience of FVN in a variety of scenarios.

REFERENCES

[1] L. Lyu, H. Yu, J. Zhao, and Q. Yang, "Threats to Federated Learning," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 12500 LNCS, pp. 3–16, 2020.

[2] D. Ye, R. Yu, M. Pan, and Z. Han, "Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach," IEEE Access, vol. 8, pp. 23920–23935, 2020.

[3] Y. Liu, C. Yan, and L. Li, "A k-means clustering-based federated learning algorithm for vehicular networks," IEEE Internet of Things Journal, vol. 7, no. 2, pp. 1011–1021, 2020.

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

[5] T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and robust federated learning through personalization," in International Conference on Machine Learning, pp. 6357–6368, PMLR, 2021.

[6] Z. Du, C. Wu, T. Yoshinaga, K. L. A. Yau, Y. Ji, and J. Li, "Federated Learning for Vehicular Internet of Things: Recent Advances and Open Issues," IEEE Computer Graphics and Applications, vol. 1, no. May, pp. 45–61, 2020.

[7] C. S. Evangeline, V. B. Kumaravelu, and A. Joshi, "Safety and driver assistance in vanets: an experimental approach for v2v," in 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 397–402, IEEE, 2019.

[8] C. Suganthi Evangeline and S. Appu, "An efficient data transmission in vanet using clustering method," International Journal of Electronics and Telecommunications, 2017.

[9] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," Knowledge-Based Systems, vol. 216, p. 106775, 2021.

[10] J. Posner, L. Tseng, M. Aloqaily, and Y. Jararweh, "Federated learning in vehicular networks: opportunities and solutions," IEEE Network, vol. 35, no. 2, pp. 152–159, 2021.

[11] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated Learning with Non-IID Data," arXiv preprint arXiv:1806.00582, 2018.

[12] M. Fang, X. Cao, J. Jia, and N. Z. Gong, "Local model poisoning attacks to byzantine-robust federated learning," Proceedings of the 29th USENIX Security Symposium, pp. 1623–1640, 2020.

[13] A. M. Elbir and S. Coleri, "Federated Learning for Vehicular Networks," arXiv, pp. 1–6, 2020.

[14] M. F. Pervej, R. Jin, and H. Dai, "Resource constrained vehicular edge federated learning with highly mobile connected vehicles," IEEE Journal on Selected Areas in Communications, 2023.

[15] A. Taik, Z. Mlika, and S. Cherkaoui, "Clustered vehicular federated learning: Process and optimization," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 12, pp. 25371–25383, 2022.

[16] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," Proceedings of the National Academy of Sciences, pp. 1–8, 2021.

[17] H. Xing, O. Simeone, and S. Bi, "Decentralized federated learning via sgd over wireless d2d networks," in 2020 IEEE 21st International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1–5, IEEE, 2020.

[18] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, "Multi-Stage Hybrid Federated Learning over Large-Scale D2D-Enabled Fog Networks," arXiv preprint arXiv:2007.09511, no. i, pp. 1–37, 2020.

[19] Q. Kong, F. Yin, R. Lu, B. Li, X. Wang, S. Cui, and P. Zhang, "Privacy-Preserving Aggregation for Federated Learning-Based Navigation in Vehicular Fog," IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8453–8463, 2021.

[20] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing," IEEE Transactions on Vehicular Technology, vol. 70, no. 6, pp. 6073–6084, 2021.

[21] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," 36th International Conference on Machine Learning, ICML 2019, vol. 2019-June, pp. 6156–6175, 2019.

[22] D. Alvarez-Melis and N. Fusi, "Geometric dataset distances via optimal transport," Advances in Neural Information Processing Systems, vol. 2020-Decem, no. NeurIPS, 2020.

[23] Q. Li, B. He, and D. Song, "Model-contrastive federated learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10713–10722, 2021.

[24] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in Proceedings of the Web Conference 2021, pp. 947–956, 2021.

[25] Y. Liu, X. Yuan, Z. Xiong, J. Kang, X. Wang, and D. Niyato, "Federated learning for 6G communications: Challenges, methods, and future directions," arXiv, pp. 105–118, 2020.

[26] S. Wang, Y. Jiang, Y. Chen, and Y. Sun, "Spatial clustering-based federated learning for vehicular networks," IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6637–6648, 2021.

[27] L. Pacheco, D. Rosário, E. Cerqueira, and T. Braun, "Federated user clustering for non-iid federated learning," Electronic Communications of the EASST, p. Volume 80: Conference on Networked Systems 2021 (NetSys 2021), Sep 2021.

[28] S. Li, X. Huang, Y. Zhang, and Y. Du, "A hierarchical clustering-based approach for vehicular edge computing in iov," IEEE Internet of Things Journal, vol. 5, no. 6, pp. 4927–4935, 2018.

[29] S. Samarakoon, H. Yang, and M. Bennis, "Federated learning for ultra-dense networks: A hierarchical clustering approach," in Proceedings of the 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), pp. 1–9, 2019.
[30] C. Yan, L. Li, and Y. Huang, "A scalable hierarchical clustering-based federated learning algorithm for vehicular networks," IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5274–5284, 2021.
[31] J. Zhao, X. Liu, T. Zhou, J. Zhang, and X. Yin, "Distributed optimal transport federated learning for privacy-preserving data aggregation," IEEE Journal on Selected Areas in Communications, vol. 39, no. 8, pp. 2283–2295, 2021.
[32] Q. Guo, Y. Wu, F. Jiang, Y. Zhu, and L. Wang, "Federated learning with non-iid data: An optimal transport approach," arXiv preprint arXiv:2011.07458, 2020.
[33] X. Li, H. Huang, X. Xue, Y. Wu, and T. Huang, "Hybrid federated and centralized learning with hierarchical clustering for resource-constrained internet of vehicles," IEEE Transactions on Industrial Informatics, vol. 17, no. 4, pp. 2594–2604, 2021.
[34] F. Murtagh and P. Contreras, "Algorithms for hierarchical clustering: an overview," Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 2, no. 1, pp. 86–97, 2012.
[35] F. Sattler, S. Wiedemann, K. R. Muller, and W. Samek, "Robust and Communication-Efficient Federated Learning from Non-i.i.d. Data," IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 9, pp. 3400–3413, 2020.
[36] N. Emami, L. Pacheco, A. Di Maio, and T. Braun, "Rc-tl: Reinforcement convolutional transfer learning for large-scale trajectory prediction," in NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium, pp. 1–9, IEEE, 2022.
[37] Z. Zhao, L. Pacheco, H. Santos, M. Liu, A. Di Maio, D. Rosário, E. Cerqueira, T. Braun, and X. Cao, "Predictive UAV base station deployment and service offloading with distributed edge learning," IEEE Transactions on Network and Service Management, vol. 18, no. 4, pp. 3955–3972, 2021.
[38] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," IEEE Transactions on Mobile Computing, vol. 13, no. 5, pp. 1061–1075, 2013.
[39] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," in arXiv preprint arXiv:1704.04861, 2017.
[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255, 2009.
[41] D. Sinha and M. El-Sharkawy, "Thin mobilenet: An enhanced mobilenet architecture," in 2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON), pp. 0280–0285, IEEE, 2019.
[42] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.
[43] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep. 0, University of Toronto, Toronto, Ontario, 2009.
[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.
[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.
[46] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for federated learning," in Proceedings of the 37th International Conference on Machine Learning (H. D. III and A. Singh, eds.), vol. 119 of Proceedings of Machine Learning Research, pp. 5132–5143, PMLR, 13–18 Jul 2020.
[47] F. Chollet et al., "Keras." https://keras.io/, 2015.
[48] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016.

LUCAS PACHECO completed his undergraduate studies in Computer Engineering at the Federal University of Pará (UFPA). He further pursued a master's degree in Electrical Engineering from UFPA. Currently, he is a doctoral student at the University of Bern, under joint supervision with the Federal University of Pará. Lucas's research interests lie mainly in vehicular networks, having previously explored mobility. He is currently investigating the effects of vehicle mobility on FL, focusing on clustering and filtering low-quality and malicious participants.

PROF. DR. TORSTEN BRAUN is the distinguished Head of the Communication and Distributed Systems research group at the University of Bern. He studied Computer Science at the University of Karlsruhe (T.H.), obtaining his Ph.D. with accolades for the best thesis in 1993. Prof. Braun has experience as a visiting scientist at institutions such as INRIA in France and Bell Labs in the USA. At the University of Bern, he has held several roles, including serving as Director of the Institute of Computer Science. He was a Research Scholar at the University of California Los Angeles and Boston University between 2017 and 2019.

DENIS ROSÁRIO graduated in Computer Engineering from the Instituto de Estudos Superiores da Amazônia in 2007. He earned his master's in Automation and Systems Engineering from the Federal University of Santa Catarina (UFSC) in 2010, spending some time at the CISTER Research Unit of the Instituto Superior de Engenharia do Porto in Portugal (2009). By 2014, he obtained his Ph.D. in Electrical Engineering from the Federal University of Para (UFPA) and in Computer Science from the University of Bern.

ANTONIO DI MAIO completed his Ph.D. in communications engineering from the University of Luxembourg in 2020, focusing on routing and content dissemination in Software-Defined Vehicular Networks. Currently, he is a Postdoctoral Researcher with the Communications and Distributed Systems Group at the University of Bern, Switzerland. His research encompasses modeling and optimization of mobile (ad-hoc) networks' performance, resilience, positioning, and resource allocation through both analytical and data-driven approaches.

EDUARDO CERQUEIRA received the Ph.D. degree in informatics engineering from the University of Coimbra, Portugal, in 2008. He is currently an Associate Professor at the Faculty of Computer Engineering and Telecommunications, Federal University of Para (UFPA), Brazil. His publications include five edited books, five book chapters, four patents, and over 250 articles in national/international refereed journals/conferences. His research interests include mobility, distributed machine learning, and wireless networks.