Journal of Internet Services and Applications, 2025, 16:1, doi: 10.5753/jisa.2025.5170

Synthetic Driving Conditions Data Generation Using Federated Generative Adversarial Networks

David de Melo A. dos Reis [**Universidade Estadual de Campinas** | ***d203757@dac.unicamp.br***]
Allan M. de Souza [**Universidade Estadual de Campinas** | ***allanms@unicamp.br***]

Instituto de Computação, Universidade Estadual de Campinas, Av. Albert Einstein, 1251 - Cidade Universitária, Campinas, SP, 13083-852, Brasil.

Received: 06 November 2024 • **Accepted:** 02 May 2025 • **Published:** 19 June 2025

Abstract Road safety remains a global challenge, especially in scenarios where behavioral and environmental factors heavily influence drivers' decision-making. Machine learning models play a crucial role in enhancing safety and informed decision-making by learning effective actions based on traffic conditions. However, training these models requires access to user data, which can compromise drivers' privacy and expose sensitive information. To address this issue, this study proposes a solution for generating synthetic driving condition data using a Federated Learning approach combined with Generative Adversarial Networks (GAN). This method enables model training across multiple federated learning clients, preserving data privacy by avoiding direct data sharing. By leveraging the Harmony dataset, similarity metrics such as Euclidean Distance and KL-Divergence were integrated into the GAN loss function to improve the quality of the generated synthetic data. The results demonstrate that the proposed approach successfully generates realistic driving condition data, supporting centralized model training while maintaining user privacy, showcasing its potential in privacy-conscious road safety applications.

Keywords: Federated Learning, Generative Adversarial Networks, Synthetic Data Generation, Driving Conditions

1 Introduction

Road safety remains a global challenge, particularly in contexts where behavioral and environmental variables directly influence drivers' decision-making processes [de Souza *et al.*, 2020; M. *et al.*, 2024; de Souza *et al.*, 2017]. In this context, research indicates that over 90% of traffic accidents result from human error, suggesting that connected autonomous vehicles represent a promising solution for reducing these rates [Dai *et al.*, 2023]. In this regard, studies such as those by Schwarting *et al.* [2018]; de Souza *et al.* [2022] highlight that machine learning models play a crucial role in enhancing safety and informed decision-making by learning effective actions based on driving conditions consequently addressing human errors in driving.

The development of a machine learning model to address this problem holds significant promise; however, its training process may face notable challenges, including limited access to comprehensive datasets and potential compromises to drivers' privacy, exposing sensitive information [Guo and Jiang, 2024]. While the study conducted by Tavakoli *et al.* [2021] benefited from adequate data volume, real-world applications involving more complex decision-making models may be hampered by the lack of sufficiently diverse data representing various driving conditions. This deficiency can arise from difficulties in capturing rare events, privacy constraints, data imbalances, and obstacles related to label generation [Guo and Jiang, 2024]. Insufficient or imbalanced data can lead to problems such as low sensitivity for underrepresented classes, stemming from inadequate reward feedback during training when the model correctly identifies these classes. Moreover, this limitation may distort the representativeness of performance metrics like accuracy, which, by emphasizing overall model performance, can obscure nuanced behaviors and specific model shortcomings [Chawla *et al.*, 2002].

To address these challenges, various studies have investigated data augmentation and generation techniques aimed at overcoming issues related to data imbalance and lack of diversity. Methods such as the Synthetic Minority Oversampling Technique (SMOTE) have proven effective in balancing structured datasets, while image manipulation strategies can be valuable for handling unstructured data [Chawla *et al.*, 2002; Lecun *et al.*, 1995]. Additionally, generative models that learn input data behavior to produce synthetic data have been explored as promising solutions [Goodfellow *et al.*, 2014; Kingma and Welling, 2022].

A significant factor contributing to data scarcity is privacy concerns. Certain categories of data may be restricted or inaccessible due to regulatory frameworks, such as the General Data Protection Regulator (GDPR), or may include sensitive personal information that users are unwilling to share [McMahan *et al.*, 2023].

Federated Learning (FL) is a machine learning technique that enables collaborative model training without requiring direct data sharing. This approach is particularly useful in scenarios where data privacy is a primary concern [McMahan *et al.*, 2023]. FL was introduced by McMahan *et al.* [2023] as a solution to train deep learning models efficiently using distributed data without centralization. In a typical FL scenario, various devices or *clients* (such as smartphones or hospitals) collaborate to train a global model under the coordination of a central server, which aggregates locally generated model updates [Konečný *et al.*, 2016; McMahan *et al.*, 2023].

Given the inherent challenges in scenarios where data sharing with third parties is unfeasible due to strict privacy constraints, this work proposes a FL strategy aimed at replicating conditions close to a real-world environment for generating driving conditions data with a Generative Adversarial Network (GAN). This federated model allows the final training to occur in a centralized manner, with all necessary preprocessing steps and model hyperparameters fine-tuning applied.

GANs constitute a class of generative models used for the creation of synthetic data, learning to replicate the behavior of training data through an adversarial training process between two models [Zhang *et al.*, 2023]. In the context of FL, their use can be advantageous as it allows a more in-depth study of data in the construction of a final model, enabling access to the behavior of data from different users without requiring identification or personalization. In this way, the main contributions of this work include: *(i)* an efficient approach for generating synthetic driving condition data preserving privacy and sensitive information about drivers; *(ii)* the exploration of different similarity metrics to guide the training of the generative model; and *(iii)* a federated training approach, enabling improved model and data evaluation for a centralized entity.

The rest of this paper is structured as follows. Section 3 and Section 2 introduces the related work and background, providing a review of foundational concepts which are pivotal to our proposed method. Section 4 describes the federated approach for training the generative model and some aspects related to the driving condition data. Section 5 details the methodology, data processing, and metrics considered for the evaluation of the machine learning models. Section 6 presents the empirical results, including the centralized baseline model, and the federated GAN model. It provides a breakdown of similarity-based losses incorporated to enhance synthetic data quality, benchmarks the performance of the federated GAN model against the centralized baseline, and analyzes the effectiveness of the proposed similarity metrics. Finally, Section 7 concludes the paper, summarizing the contributions and potential impact of our federated approach on privacy-preserving data generation in road safety applications.

2 Background

This section presents the fundamental definitions addressed throughout this work. We begin with an explanation of the functioning of FL, followed by a description of the generative model used in this study. Then, the cost functions metrics used to train the GAN are presented, including: Kullback–Leibler divergence, Jensen–Shannon divergence, Maximum mean discrepancy and Kernel Density Estimation. Finally, we explain CTGAN, the reference implementation used in this work.

2.1 Federated Learning

FL is a collaborative learning framework with three stages: *(i) Client selection:* in this stage, the server, following a policy, chooses the best clients for training and sends the initial model. *(ii) Local training:* the selected clients train the received model on their local data. At the end of training, the model weights are sent to the server. *(iii) Model aggregation:* in this phase, the server receives the trained models and aggregates the learned knowledge into a single model, following an aggregation policy. Originally, FedAvg is used, but other methods can be applied. These three stages together are known as communication rounds, and during the federated process, this occurs T times until a stopping condition is met.

FedAvg is a traditional algorithm that aggregates models through a weighted average and random client selection [McMahan *et al.*, 2023]. Given a set of clients C, an objective function $\mathcal{L}$, and a sequence of training rounds $t \in 1, 2, \ldots, T$, each client i has a dataset $d_i \in \mathcal{D}$, where $\mathcal{D} = \cup_{i \in C} d_i$. The selected clients represent a subset $K \in C$ that receive the global model w_g^t and train it on their local data d_i. Thus, the global model is updated according to Equation 1.

$$w_g^{t+1} = \sum_{i=1}^{|K|} \frac{|d_i|}{|D|} \mathcal{F}(w_g^t) \tag{1}$$

Thus, FL aims to minimize the function $f(\cdot)$, which represents the collective objective, as shown in Equation 2:

$$\min_w f(w) = \sum_{i=1}^{|C|} p_c \mathcal{L}_i(w) \tag{2}$$

The training rounds and client selection continue until convergence occurs or the model's performance reaches a predefined level.

One of the main challenges of FL is managing non-IID (*non-independent and identically distributed*) data, where different clients may have significantly diverse data distributions. This can lead to degradation in the performance of the global model [Kairouz *et al.*, 2021]. To mitigate these effects Li *et al.* [2020] proposed techniques such as weighted averaging , customized optimization algorithms, client selection [de Souza *et al.*, 2024; Souza *et al.*, 2023]. Furthermore, privacy and security issues, such as inference attacks and model poisoning, are significant concerns in FL. Various defense strategies, including the use of differential privacy techniques and robust client authentication, have been explored to address these challenges [Geyer *et al.*, 2018].

2.2 Generative Adversarial Networks

GANs were introduced by Goodfellow *et al.* [2014], revolutionizing the field of machine learning with a novel approach to synthetic data generation. These networks are composed of two competing neural networks, a generator and a discriminator as illustrated in Figure 1. The generator network aims to create data indistinguishable from real data, while the discriminator attempts to differentiate the generated data from authentic data [Goodfellow *et al.*, 2014].

Despite their success, GANs face significant challenges, such as instability during training and mode collapse, where the generator produces a limited variety of outputs. Various

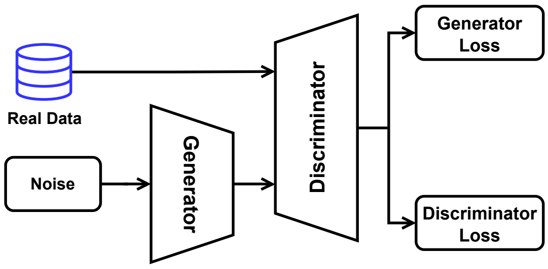


Figure 1. GAN operation.

studies have focused on techniques such as self-regulated training and modifications to the loss function to address these issues [Salimans *et al.*, 2016].

Since their introduction, numerous GAN variants have been proposed to improve both training stability and the quality of generated data. Among these variants are Weighted Gradient Generative Adversarial Networks (WGANs), which utilize weighted gradient calculations, relying on the Wasserstein distance to define the cost function. This approach not only enhances training stability by preventing gradient vanishing issues but also contributes to more consistent and higher-quality data generation, reducing the phenomenon of mode collapse [Arjovsky *et al.*, 2017].

Building upon the advancements of WGANs, CTGAN extends their capabilities to tabular data, incorporating specialized treatments tailored to different data types [Xu *et al.*, 2019]. Among the primary treatments, it includes the application of one-hot encoding for categorical data, transforming categorical columns into sparse columns for better representational accuracy. Additionally, the model employs a Bayesian Gaussian Mixture, which generates distinct labels for each numerical column through clustering, resulting in a normalized data value with a corresponding class, represented in one-hot encoding notation.

In addition to the applied transformations, CTGAN utilizes a latent vector to condition the output of the generator model, as well as a learning strategy based on subsampling a random class during training to prevent mode collapse, which could adversely affect classes with fewer data.

2.3 Bayesian Gaussian Mixture

The Bayesian Gaussian Mixture Model (BGM) is a probabilistic modeling technique that combines Bayesian inference with Gaussian distributions to identify complex patterns in data. In BGM, the parameters defining the Gaussian distributions are treated as random variables, with uncertainties expressed through prior distributions, which guide the model in identifying patterns in a more adaptable manner. This means that the number of distributions required to represent the data does not need to be specified in advance, as the model automatically adjusts the count based on observed variability [Bishop, 2006].

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{3}$$

where, K is the number of components in the mixture, π_k represents the weight of component k, such that $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$, and $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the Gaussian probability density with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$.

With this approach, BGM better adapts to contexts where data structure is complex or where there is uncertainty regarding the number of patterns. Additionally, the use of prior distributions imposes natural regularization, reducing the risk of overfitting by penalizing irrelevant components. The model employs approximate inference methods, such as variational inference, to find the most probable distribution of parameters, promoting a more robust and interpretable data representation [Bishop, 2006].

2.4 Symmetry Functions

S. Kullback and R. A. Leibler Kullback and Leibler [1951] introduced the Kullback-Leibler divergence (KL), a similarity measure between two probability distributions that quantifies the extent to which a distribution Q diverges from a reference distribution P.

To address the asymmetry of KL divergence, the Jensen–Shannon divergence (JSD) was introduced as a symmetric and smoothed alternative for measuring similarity between distributions [Lin, 1991]. Essentially, the JSD computes the weighted average of two distributions P and Q and then calculates the KL divergence of P and Q with respect to M.

While KL and JSD focus on measuring distributional divergence, other approaches compare distributions using different mathematical frameworks. One such approach is the Maximum Mean Discrepancy (MMD), which measures differences through kernel methods. The MMD metric is a statistical measure that evaluates the difference between two probability distributions by comparing the means of their representations in a reproducing kernel Hilbert space (RKHS), defined by a kernel function. By mapping the samples to this space, MMD measures the distance between the means of the transformed samples from the two distributions.

Beyond direct distributional comparisons, another essential statistical tool is Kernel Density Estimation (KDE), which focuses on reconstructing probability density functions from finite samples. KDE achieves this by centering a kernel function at each sample point and aggregating these contributions to generate a smoothed density curve. Each kernel is a symmetric function that defines the weight around each point, which diminishes as the distance from the point increases [Parzen, 1962].

3 Related Works

This section presents literature works that tackles federated and augmentation methods for generating synthetic data. Studies conducted by Tavakoli *et al.* [2021] provided an insightful approach for analyzing and identifying various traffic events and driver behaviors using data collected from simple devices such as cameras, smartwatches, and microphones. Building upon this foundation, works like the one proposed by Khoa *et al.* [2024] develop machine learning methods within privacy-preserving scenarios, leveraging federated learning to prevent the sharing of sensitive data. These methods aim to identify and potentially predict traffic events, such as sudden braking by the leading vehicle, unsafe lane changes, and driver distractions that may compromise attention to the road.

However, as observed by Khoa *et al*. [2024], the class imbalance is a significant issue when using traffic data, as common traffic situations occur much more frequently than adverse events. This imbalance negatively impacts model training, particularly in federated learning scenarios. Additionally, data collection can be challenging due to privacy concerns related to data sharing and the high cost of acquisition [Tamayo-Urgilés *et al*., 2024].

To address this issue, Khoa *et al*. [2024] opted to use normalization and SMOTE as class re-balancing techniques as part of the pre-processing stage before distributing data among the federation participants. In this approach, the federated nodes receive already balanced and normalized data prior to training. While the results of this method are promising, its practical applicability is limited. This is because, in real-world scenarios, data preprocessing must be performed locally on each user's data rather than on the entire dataset before distribution, meaning that normalization and class rebalancing would be applied only to each user's subset, rather than the complete dataset.

Given this, studies such as Zhao *et al*. [2024a], Tamayo-Urgilés *et al*. [2024], and Miller *et al*. [2025] focus on the generation of synthetic datasets using traffic simulators and generative models.

The approach proposed by Zhao *et al*. [2024a] relies on the CARLA simulator to generate realistic traffic scenes along with their semantic segmentation, combined with text-to-image models. This enables the training of models capable of generating scenes from textual inputs.

Alternatively, Tamayo-Urgilés *et al*. [2024] trains and compares GANs using data collected during 50 minutes of driving a vehicle equipped with various sensors. To assess the quality of the generated data, they compare the performance of accident risk prediction models. While the results are promising, the dataset is limited in diversity due to the high cost of data acquisition, which constrains the monitoring of different drivers in various vehicles. Finally, Miller *et al*. [2025] uses a synthetic image data set generated by CARLA for pre-training, followed by training on a real dataset to develop an autonomous driving model that predicts the speed and angle of the vehicle. Although the results indicate performance improvements when pre-training with synthetic data, privacy concerns arise due to the use of real images and driving data.

Another promising approach for synthetic data generation is presented by Shaker *et al*. [2024], which explores the decentralized training of a GAN. In this framework, data remains on devices referred to as Workers, which are responsible for training the GAN's discriminator, while the generator is trained by a central entity. During training, the central entity generates synthetic data, which is sent to the Workers. The Workers then train the discriminator using their respective portions of real and synthetic data and return the computed loss to the central entity, which continues the training process of the generator.

Some authors, such as Beitollahi *et al*. [2024], propose a one-shot federated learning strategy to enhance communication efficiency during federated training. This approach leverages a Foundation Model to extract features from the data, which are then used to train a parametric model. The trained parametric model is subsequently sent to the central entity, which utilizes it to sample features and train the corresponding model in a centralized manner. This method preserves user data privacy while completing the federation process in a single round.

In parallel, authors such as Li and Wang [2019], Lin *et al*. [2020], and Zhu *et al*. [2021] have proposed knowledge distillation in federated learning, relying on external datasets or generative models to produce synthetic samples. However, these approaches face limitations due to their dependence on high computational resources on the server and the challenge of ensuring the quality of the generated data. To address these issues, Zhao *et al*. [2024b] introduced FedF2DG, which distinguishes itself by entirely eliminating the need for external datasets or generative models. Instead, it leverages local models for data generation, dynamically adapting both the label distribution and the volume of synthetic data.

Other studies, such as Ghavamipour *et al*. [2023], focus on data privacy by training GANs in a federated manner. Specifically, this work proposes the use of Homomorphic Encryption to encrypt each client's GAN models before sending them to the centralized server, which performs the aggregation step without the need to decrypt the models or expose client updates. Additionally, they employ differential privacy, a technique that introduces noise into the training gradients, allowing for granular control over the level of data privacy at the cost of model quality.

Thus, the studies in the literature present certain challenges that will be addressed by our proposal in the following chapters, specifically for our application. These challenges include: the need for rebalancing and normalizing all data together before the start of federated training; performance degradation when applied to non-IID data; the requirement of data in the centralized server for training a global foundation model; and performance degradation when applied to tabular data, as existing approaches were primarily designed for image-based applications.

4 Federated Data Generation

The proposed FL approach is based on training a federated GAN to generate synthetic driving conditions data on a centralized server, enabling the subsequent training of a machine learning model, as illustrated in the pipeline in Figure 2. In this example we each client represent a vehicle and driver and passengers inside the car in which are producing data from wearable devices and smartphone as well as sensor and cameras inside the car sensing the urban environment and capturing driving conditions.

In the context, data considered for describing the driving conditions collected by devices can be sensitive, given that some sensors include internal and external cameras, GPS, and microphones, driving patterns. Information generated from this data includes user location, presence of people inside the car, driving speed and acceleration, presence of surrounding vehicles, ambient music, driver gaze direction, and physiological data such as heart rate. Therefore, FL becomes a viable solution for training machine learning models that can provide value to the user without compromising privacy.

With this approach, the driving conditions data enables the creation of models capable of alerting users to events on the road while driving. Specifically, we aim to label different driving events to identify categories established by Harmony, such as sudden braking of vehicles in front, intersection crossings, normal driving actions like overtaking and lane changes, and non-driving actions such as side conversations or eating.

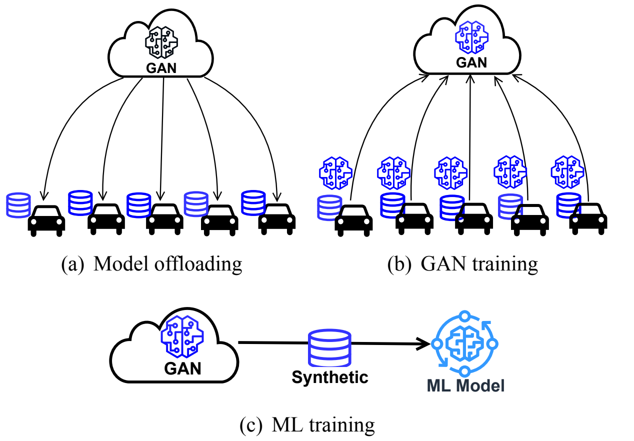


Figure 2. Federated data generation and ML training: *step 1* the server offloads the GAN model to the clients; *step 2* each client train the GAN model on its own data and send the trained parameters to the server; *step 3* the server uses the aggregated GAN model to produce synthetic data to train a machine learning model.

More specifically, clients (i.e., vehicles and drivers) will locally train a GAN, which employs a generator model and a discriminator model for each possible class described in the driving conditions. The FL aggregation process will combine all generators and discriminators corresponding to pattern. As training rounds progress, the global model will gradually learn to reproduce the behavior of each user's data for each driving condition.

In the next stage, the server, possessing the trained GAN, will perform balanced synthetic data generation. At this point, the federation server gains access to a representation of the users' data without compromising their privacy. With this data representation, the federation server can carry out various processes, including data preprocessing, model benchmarking, hyperparameter tuning, and model training. Finally, the federation server can send the fully trained and fine-tuned model to federation users for their respective applications.

5 Methodology

In this section, we present the methodology employed and the results obtained in our study. We begin by detailing the processing of the Harmony dataset [Tavakoli *et al.*, 2021] used to provide the various driving conditions data, including how the driving condition events were labeled based on heart rate variations of the driver and which sensor features were selected for model training. We then establish a centralized baseline by training classical classification models—such as Logistic Regression, Random Forest, Support Vector Machines, Gradient Boosting, and XGBoost—and evaluate their performance using cross-validation. Subsequently, we introduce our federated approach, which involves training a modified federated Generative Adversarial Network to generate synthetic data, incorporating similarity metrics like Euclidean Distance, Kullback-Leibler Divergence, Jensen-Shannon Divergence, and Maximum Mean Discrepancy into the generator's loss function. This approach aims to enhance the quality of the synthetic data and to evaluate the performance of models trained on these data in comparison to the centralized baseline.

5.1 Data processing

The dataset used in this study are sourced from the Harmony study dataset, as presented in Tavakoli *et al.* [2023], which comprises information processed with the aid of machine learning algorithms from various sensors to describe various driving conditions. This information includes the detection of nearby vehicles and pedestrians from camera recordings, the driver's gaze direction, facial expressions, and heart rate, as well as vehicle accelerometer and gyroscope data, lighting conditions, and ambient noise [Tavakoli *et al.*, 2021].

In this work, the labels were generated based on the events proposed by Tavakoli *et al.* [2021], utilizing their methodology to identify events through variations in heart rate data. The number of occurrences and their respective events are presented in Table **??** and the categories are defined as follows: *(i)* primary tasks, representing driving-related actions such as lane changes and overtaking; *(ii)* secondary tasks, representing non-driving-related actions such as handling a mobile phone or eating; *(iii)* lead vehicle, representing braking events of the vehicle ahead; and *(iv)* intersection, representing occurrences of passing through intersections. Additional characteristics of the dataset, such as data types and variable information, are also presented in Table 1. For the classification models, we employed the dataset sampling, which already incorporates the handling of different sensor frequencies. The solution adopted by the authors involved using the average frequency, standardizing the sensor sampling at 10 Hz.

The driving condition event identification through heart rate uses Bayesian Change Point Detection (BCP), which employs Bayesian inference to determine the probability that a change has occurred in a time series. The BCP method is applied here to identify the occurrence of events based on change points in heart rate data. For more details about this method, refer to Tavakoli *et al.* [2021].

Regarding the driving conditions considered in this work, we selected indicators of the presence of pedestrians, cyclists, and surrounding vehicles; the distance to the vehicle ahead; indicators of the driver's gaze direction; gyroscope and accelerometer values and magnitudes; as well as lighting levels and the driver's heart rate. For training, the dataset was divided into a proportion of 80% for training data and 20% for test data. Thus, this data will be use to train the generative model and also to train the machine learning model to learn the driving events classification.

Table 1. Dataset Characteristics

Category	Occurrences
intersection	350
lead_vehicle	733
primary_tasks	683
secondary_tasks	971
Dataset Statistics	**Valor**
Number of variables	95
Number of observations	2737
Missing cells (%)	0.0%
Duplicate rows (%)	0.0%
Variable Types	**Quantidade**
Numeric	88
Categorical	7

5.2 Metrics Evaluated

This section describes the metrics used to evaluate the machine learning models trained with the real data and also with the synthetic one generated by the federated approach.

- **Accuracy:** Measures the overall correctness of a model by showing the proportion of correctly predicted instances out of the total number of instances. It's a general indicator of model performance.

$$\text{Accuracy} = \frac{\text{correctly classified classes}}{\text{all classifications}}$$

- **Precision:** Indicates how many of the positively predicted instances are actually true positives. It focuses on the reliability of positive predictions and is crucial when false positives carry a significant cost.

$$\text{Precision} = \frac{\sum_{i=1}^{N} \text{Correct Predictions for Class } i}{\sum_{i=1}^{N} \text{Total Predictions for Class } i}$$

- **Recall**: Shows the ability of the model to correctly identify all actual positive cases.

$$\text{Recall} = \frac{\sum_{i=1}^{N} \text{Correct Predictions in Class } i}{\sum_{i=1}^{N} \text{Total Actual Instances in Class } i}$$

- **F1-Score**: The harmonic mean of precision and recall, balancing the two. It's useful when there's an uneven class distribution or when you need a balance between precision and recall.

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Considering these metrics we can evaluate the performance of each machine learning. To do so, we first define a baseline approach to compare with the machine learning model trained with the synthetic federated GAN. It is important to notice that the machine learning model aims to classify the driving event produced by the labeled data present in the dataset [Tavakoli *et al.*, 2023]

6 Performance Analysis

This section presents the results achieved by the proposed solution. First, we stablish the baseline as a centralized approach trained on the real data. Then we apply the proposed solution for generating synthetic data and check its performance in comparison with the real one.

6.1 Centralized Baseline

In this step, we aim to establish a performance benchmark for the federated model. Since the proposal involves generating synthetic data from a federated generative model, we will use the same classical models established in this section for centralized training but with synthetic data generated by the federated model. These models will then be evaluated using real test data, also allowing an assessment of the quality of the generated synthetic data by comparing the performance of classical models trained on synthetic data with those trained on real data.

To establish a baseline for comparison with the performance of federated training, we opted for a grid search over different hyperparameters in various classical classification models. This grid search consists of the Cartesian product of all hyperparameters, enabling an effective approximation of the best configuration for a more precise validation of the generalization capacity of each tested model, including Logistic Regression, Random Forest, SVM, Gradient Boosting, and XGBoost.

For performance evaluation of the different models, we used the average Weighted F1-Score from a 5-fold cross-validation.

The 5-fold cross-validation is a technique that divides the data into five equally sized parts, performing in five distinct stages the alternation of each part for validation, while the remaining parts are used for training.

The models utilized are as follows: *Logistic Regression*, which estimates the probability of a class by applying the sigmoid function to the weighted linear combination of independent variables, mapping the output to values between 0 and 1; *Random Forest*, which combines multiple independent decision trees, where each tree is trained on a random sample of the data and a random selection of attributes, with the final decision made by majority vote among the predictions of all trees; *SVM*, which seeks to find an optimal hyperplane that maximizes the margin between distinct classes in the data, relying on support vectors (data points closest to the margin); *Gradient Boosting*, which builds strong predictive models by combining multiple weak models, where each new tree corrects the errors of the previous ones by fitting to the gradient of the loss function; and *XGBoost*, an advanced implementation of Gradient Boosting that enhances efficiency, speed, and performance by incorporating regularization, parallel processing, and overfitting reduction techniques.

Figure 3 shows the results achieved by each machine learning models after performing hyperparameter tuning with cross-validation across various models, we observed that tree based models achieve better performance than Logistic Regression and SVM. In particular, they achieved an f1-score

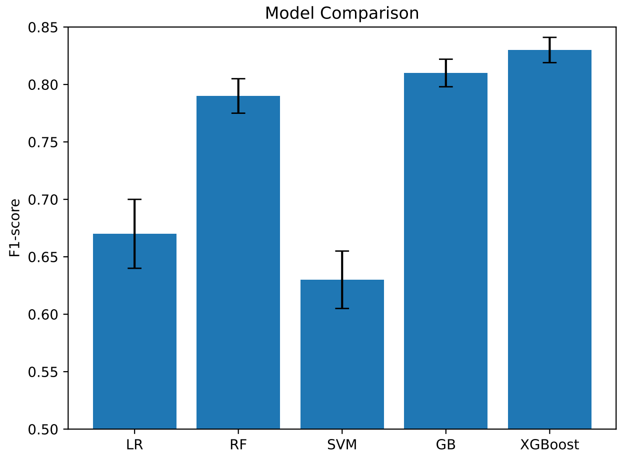


Figure 3. Comparison of centralized models: Logistic Regression (LR), Random Forest (RF), Support Vector Machine (SVM), Gradient Boosting (GB) and XGBoost. In the first stage, we define the models and the range of hyperparameters. In the second stage, a grid search is performed, combining all possible hyperparameter configurations for each model. In the third stage, a 5-fold cross-validation is conducted to calculate the average accuracy across the training and validation batch combinations. Finally, the hyperparameters yielding the highest average accuracy are selected.

of 65%, while tree-based models achieved f1-scores higher than 80%. The model with the highest score was the XGBoost, achieving a 83%, more details about the classification of this model will be present in the next section.

6.2 Federated GAN Evaluation

The federation was conducted with five clients, each training their respective four generators and four discriminators, corresponding to each data class, over 10 epochs per federation round, using FedAvg as the aggregation strategy.

In this work, we modified CTGAN version of Xu *et al.* [2019], which implements various guidance methods for the generator and incorporates a strategy based on using a GAN per class rather than a conditional latent vector for class representation. Additionally, we modified the data transformation approach proposed by Xu *et al.* [2019], which employs a Bayesian Gaussian Mixture with a dynamic number of components based on the probability of a component being relevant. Instead, we adopted a fixed number of components to ensure that the data maintains the same shape across all users, thereby enabling the federated averaging of generators with a consistent shape.

Another modification involved the use of a dedicated generator and discriminator for each class, which became necessary after initial experiments failed to optimize our similarity metric and the conditional vector from Xu *et al.* [2019]. In this scenario, a performance bottleneck was observed during training due to the high bandwidth consumption required for communication between all models, the users, and the federation server.

The Gaussian Mixture Model is employed as a data preprocessing step, serving as a feature augmentation technique proposed by Xu *et al.* [2019] to handle the non-Gaussian distribution of continuous columns in tabular datasets. The method proposed by Bishop [2006] and implemented in CTGAN involves categorizing each column's data separately and adding, to the normalized column, a one-hot encoding corresponding to the calculated class. To enable model federation, we adopted an approach with a fixed number of classes calculated by the Gaussian Mixture Model, whereas CTGAN uses a variable class count strategy. This modification was necessary because different class distributions in distinct federations could yield conflicting numbers of new features, resulting in incompatible GANs for the model aggregation process.

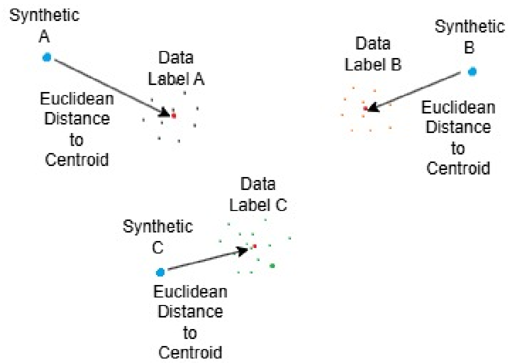


Figure 4. Demonstration of Euclidean Similarity

The GAN architecture used was WGANGP, as proposed by Gulrajani *et al.* [2017], preserving the model architecture and hyperparameters from Xu *et al.* [2019]. The training modification included the addition of a loss component, based on various similarity metrics, between the generated and real data, which was incorporated into the generator model's loss function. We propose the addition of this similarity-based loss ($\mathcal{L}_{sim}$) to enhance the quality of the generated data, aiming to minimize the degradation in accuracy of a classification model trained on these synthetic data.

To improve the fidelity of synthetic data in relation to real data, we integrated a similarity measure between synthetic and real data into the generator model's loss function, applying the following formula:

$$\mathcal{L}_g = -score_{\text{fake}} + \mathcal{L}_{sim} \tag{4}$$

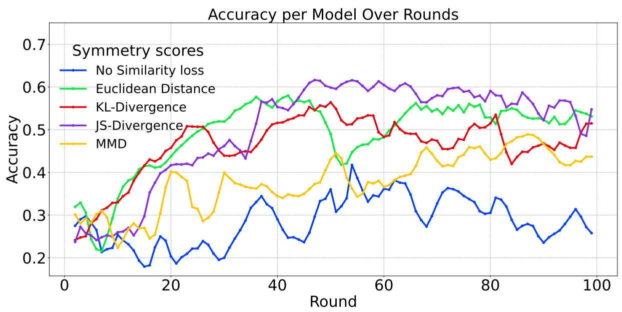
In this formulation, the term $-\text{score}_{\text{fake}}$ represents the adversarial component, which aims to maximize the discriminator's score for synthetic data, encouraging the generator to produce samples that closely resemble real data. The second term, $\mathcal{L}_{\text{sim}}$, acts as a regularizer, enforcing a similarity measure that guides synthetic data toward the characteristics of real data.

Among the similarity measures, we used Euclidean Distance, KL-Divergence, JS-Divergence, and MMD. The Euclidean distance is calculated between the generated data and a central point of the real data for each category as illustrated in Figure 4. For JS-Divergence and MMD, the calculations were applied between the synthetic data generated at each training epoch and the real data of the respective category. For KL-Divergence, it was necessary to apply KDE on the real and synthetic samples to estimate the probability distribution of the data before calculating the KL-Divergence.

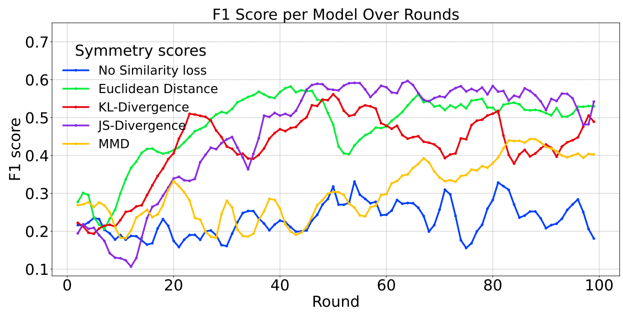
We compared each similarity method proposed to check which one would perform better in this scenario. Table 2 shows the accuracies for each similarity-based loss applied

Table 2. Accuracy Across Different Similarity Metrics and Fine Tuned Models

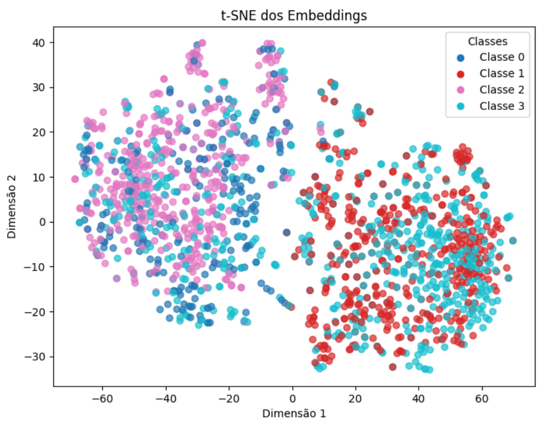
Similarity Metric	Logistic Regression	Random Forest	SVM	Gradient Boosting	XGBoost
No Similarity loss	0.37	0.38	0.40	0.37	0.32
Euclidean Distance	0.42	0.48	0.44	0.44	0.48
KL-Divergence	0.54	0.57	0.55	0.57	0.59
JS-Divergence	0.53	**0.66**	0.58	**0.65**	**0.65**
MMD	0.41	0.46	0.42	0.46	0.45



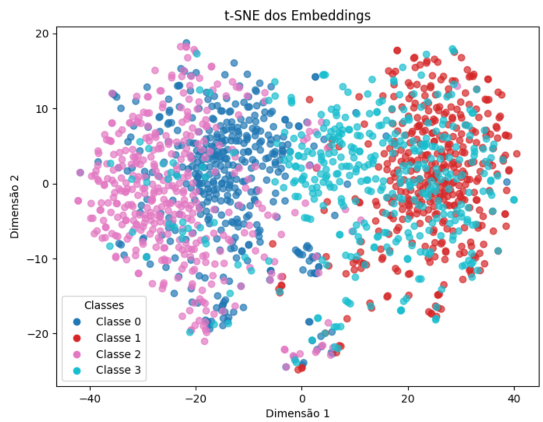
(a) Accuracy score over federation rounds.



(b) F1 score over federation rounds.



(c) Real Data Latent Space Plots



(d) Synthetic Data Latent Space Plots

Figure 5. Analysis of accuracy (a) and F1-score (b) metrics over the communication rounds for the federated GAN training and model classification performance considering trained with real data (c) and trained with synthetic data (d).

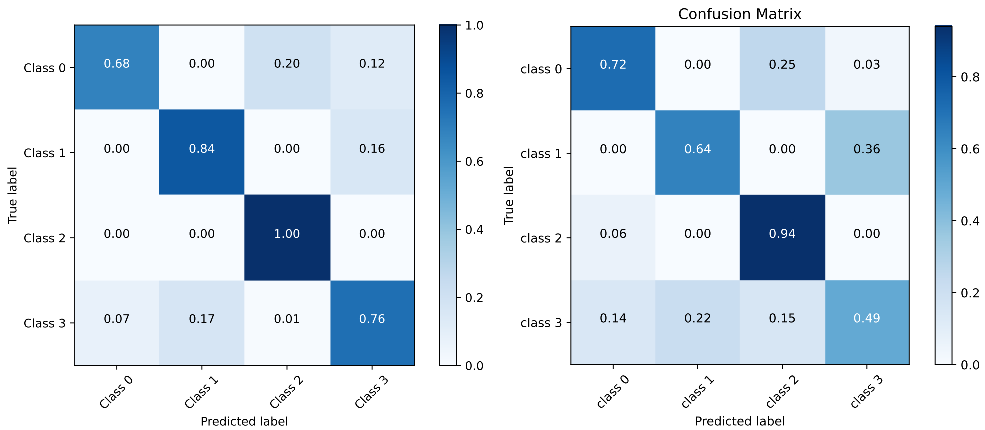
to each machine learning model. As we can see, the JS-Divergence enabled a better representation for the synthetic data. After conducting evaluations with new hyperparameter adjustments on the machine learning models, we observed a performance improvement by incorporating the similarity metric into the generator's loss function in Table 2. Notably, a superior improvement was achieved when using the Jensen-Shannon divergence with the Random Forest model, reaching an accuracy of 66%.

6.3 Analysis Model Performance considering Synthetic Data vs Real Data

In this section, we evaluate the performance of the federated model over the communication rounds and compare the classification performance of a model trained on real data with one trained on synthetic data generated by the federated model to analyze the efficiency of the federated data generation.

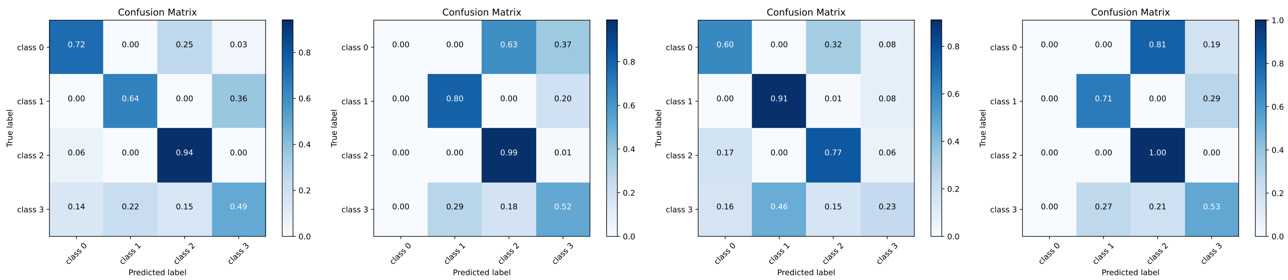
Figure 5 presents the accuracy (Figure 5(a)) and F1-score (Figure 5(b)) metrics for an XGBoost model trained on synthetic data generated at the end of each communication round. The results reveal some instability among the models, with the GAN using Jensen-Shannon divergence showing the most consistent improvements, as confirmed by the final evaluation of the adjusted classification models. Notably, the XGBoost model trained on synthetic data achieved an accuracy exceeding 60% across the 100 communication rounds used for training the federated generative model.

To gain deeper insights into the model's performance using synthetic data, we generated a visualization based on training a convolutional neural network to learn a latent representation of the data, which was subsequently projected using T-SNE. Figure 5(c) illustrates the distribution of the latent representation for real data, whereas Figure 5(d) depicts the latent representation of synthetic data. These visualizations reveal an overlap between certain classes, which adversely affects the accuracy of data classification.



(a) confusion matrix of the model trained on real data (b) confusion matrix of the model trained with synthetic data

Figure 6. Comparison of Training Results with Real vs Synthetic Data. Where Class 0: intersection, Class 1: lead vehicle, Class 2: primary tasks, Class 3: secondary tasks



(a) Our Proposal (b) Federated Learning (c) FedPFT (d) Safety is our Friend

Figure 7. Comparison of results between different training methodologies

Figure 6 illustrates the accuracy per class for models trained on both real and synthetic data, emphasizing the challenges in distinguishing certain classes. The model trained on synthetic data closely approximates the performance of the model trained on real data for classes 0, 1, and 2, even improving the performance for class 1. However, there is a noticeable decrease in performance for class 3 compared to the model trained on real data.

This analysis demonstrates that the federated generative model can produce synthetic data capable of training an effective classifier while maintaining data privacy. Although the current model performs well in classifying driving condition events, further exploration with alternative models could enhance performance and reduce discrepancies, particularly for challenging classes like class 3. However, finding the optimal model for driving condition classification is beyond the scope of this work.

The federated training of GANs can achieve performance comparable to centralized training, the quality of the generated data remains suboptimal for achieving the desired performance levels. In particular, data transformations necessary for handling tabular data, which involve clustering dataset columns, may present challenges in federated contexts. This is especially true when individual users' samples are not representative of the overall distribution, resulting in incomplete or fragmented clusters.

Consequently, there remains a need to investigate alternative data preprocessing methods that are more compatible with federated environments and better capture the underlying characteristics of the data.

6.4 Comparison with Related Work

In this study, we will replicate methodologies applied in related works to compare their potential results when applied to our dataset, in relation to our proposed methodology. Our approach will be represented by the classifier and similarity metric with the best performance, specifically JS Divergence with Random Forest. Among the related works, we will implement federated learning with a CNN and apply data normalization at each federation node to represent the federated learning scenario. Another alternative is the Safety is our Friend [Khoa *et al.*, 2024], which applies SMOTE at the federated nodes for class rebalancing. Finally, we will implement FedPFT [Beitollahi *et al.*, 2024], which was adapted by creating an additional partition of the dataset for training a CNN as a foundation model. This foundation model extracts features from the data at the federated nodes, which are then used to train a GMM. The GMM samples these features at a centralized server, where a XGBoost is trained. To ensure a fair comparison between methods, the training, validation, and test datasets will remain the same across all approaches. All models will use 60% of the data for training, 20% for validation, and will be evaluated using the remaining 20% for testing. The results of each evaluation are presented in Table

Table 3. Weighted Scores Across Different Methodologies

Methodology	Precision	Recall	F1-Score	Accuracy
Our Proposal	0.66	0.66	0.65	0.66
FedAvg	0.58	0.65	0.60	0.65
FedPFT	0.61	0.59	0.56	0.59
Safety is our Friend	0.43	0.56	0.47	0.56

3.

Based on these results, we can conclude that Our Proposal offers a balanced complexity while effectively handling non-IID data, a critical capability for real-world distributed systems. However, it comes with high communication and computation costs. Despite these trade-offs, it achieves the highest overall performance, leading across all key metrics. In contrast, FedAvg struggles with non-IID data, which impacts its model performance, resulting in moderate outcomes. FedPFT offers an attractive trade-off with low communication cost and support for non-IID data but delivers lower performance compared to our proposal. Finally, Safety is Our Friend suffers from high complexity, high communication cost, and its inability to handle non-IID data, all of which severely limit its effectiveness. In summary, while Our Proposal provides the most favorable balance between system attributes and model performance, FedPFT and FedAvg still offer specific advantages depending on system constraints.

7 Conclusion

This research conducted experiments to assess the feasibility of training GANs within a federated learning framework for generating synthetic data of driving conditions, with the objective of performing final model training in a centralized setting. This methodology aims to preserve user privacy while facilitating effective data processing and the exploration of multiple model architectures.

The simulations revealed the potential of a federated GAN to approximate the behavior of real data without requiring users to share their data with a centralized server. This approach enables effective data handling and broadens the scope for exploring various models to classify these synthetic data.

In a scenario with data distributed across five distinct users, it was demonstrated that a federated-trained GAN could synthesize data closely resembling the real dataset. This finding suggests further opportunities for refining techniques to achieve a more precise approximation of real data.

Declarations

Acknowledgements

The authors would like to thank projects #23/00673-7, São Paulo Research Institute (FAPESP), CNPq project #405940/2022-0, CAPES project #88887.954253/2024-00 and FAEPEX/PIND UNICAMP project #519.287 for the support.

Authors' Contributions

Both authors contributed equally to all aspects of the work involved in the development of this article, including the conception of ideas, execution of experiments, and writing of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of Data and Materials

The dataset used in this research is publicly available in the OSF repository at https://osf.io/zextd/. The source code is available at https://github.com/DaveAlmR/Synthetic-Driving-Conditions-Data-Generation-Using-Federated-Generative-Adversarial-Networks.

References

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein gan. DOI: 10.48550/arXiv.1701.07875.

Beitollahi, M., Bie, A., Hemati, S., Brunswic, L. M., Li, X., Chen, X., and Zhang, G. (2024). Parametric feature transfer: One-shot federated learning with foundation models. DOI: 10.48550/arxiv.2402.01862.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer. DOI: 10.1117/1.2819119.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357. DOI: 10.1613/jair.953.

Dai, Y., Wang, C., and Xie, Y. (2023). Explicitly incorporating surrogate safety measures into connected and automated vehicle longitudinal control objectives for enhancing platoon safety. *Accident Analysis Prevention*, 183:106975. DOI: 10.1016/j.aap.2023.106975.

de Souza, A. M., Braun, T., Botega, L. C., Villas, L. A., and Loureiro, A. A. F. (2020). Safe and sound: Driver safety-aware vehicle re-routing based on spatiotemporal information. *IEEE Transactions on Intelligent Transportation Systems*, 21(9):3973–3989. DOI: 10.1109/TITS.2019.2958624.

de Souza, A. M., Brennand, C. A., Yokoyama, R. S., Donato, E. A., Madeira, E. R., and Villas, L. A. (2017). Traffic management systems: A classification, review, challenges, and future perspectives. *International Journal of Distributed Sensor Networks*, 13(4):1550147716683612. DOI: 10.1177/1550147716683612.

de Souza, A. M., Maciel, F., da Costa, J. B., Bittencourt, L. F., Cerqueira, E., Loureiro, A. A., and Villas, L. A. (2024). Adaptive client selection with personalization for communication efficient federated learning. *Ad Hoc Networks*, 157:103462. DOI: 10.1016/j.adhoc.2024.103462.

de Souza, A. M., Oliveira, H. F., Zhao, Z., Braun, T., Loureiro, A. A., and Villas, L. A. (2022). Enhancing sensing and decision-making of automated driving systems with multi-access edge computing and machine learning. *IEEE Intelligent Transportation Systems Magazine*, 14(1):44–56. DOI: 10.1109/MITS.2019.2953513.

Geyer, R. C., Klein, T., and Nabi, M. (2018). Differentially private federated learning: A client level perspective. DOI: 10.48550/arxiv.1712.07557.

Ghavamipour, A. R., Turkmen, F., Wang, R., and Liang, K. (2023). Federated synthetic data generation with stronger security guarantees. In *Proceedings of the 28th ACM Symposium on Access Control Models and Technologies*, SACMAT '23, page 31–42, New York, NY, USA. Association for Computing Machinery. DOI: 10.1145/3589608.3593835.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks. DOI: 10.1145/3422622.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved training of wasserstein gans. DOI: 10.48550/arxiv.1704.00028.

Guo, W. and Jiang, P. (2024). Weakly supervised anomaly detection with privacy preservation under a bi-level federated learning framework. *Expert Systems with Applications*, 254:124450. DOI: 10.1016/j.eswa.2024.124450.

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Eichner, H., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and Zhao, S. (2021). Advances and open problems in federated learning. DOI: 10.1561/9781680837896.

Khoa, T. A., Trac, N. D., Tinh, V. P., Nam, N. H., Dang, D. N. M., Son, H. H., and Lam, P. D. (2024). Safety is our friend: A federated learning framework toward driver's state and behavior detection. *IEEE Transactions on Computational Social Systems*, 11(2):2340–2358. DOI: 10.1109/TCSS.2023.3273727.

Kingma, D. P. and Welling, M. (2022). Auto-encoding variational bayes. DOI: 10.48550/arXiv.1312.6114.

Konečný, J., McMahan, H. B., Ramage, D., and Richtárik, P. (2016). Federated optimization: Distributed machine learning for on-device intelligence.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86. DOI: 10.1214/aoms/1177729694.

Lecun, Y., Jackel, L., Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Muller, U., Sackinger, E., Simard, P., and Vapnik, V. (1995). *Learning algorithms for classification: A comparison on handwritten digit recognition*, pages 261–276. World Scientific. Available at: `https://www.eecis.udel.edu/~shatkay/Course/papers/NetworksAndCNNClasifiersIntroVapnik95.pdf`.

Li, D. and Wang, J. (2019). Fedmd: Heterogenous federated learning via model distillation. DOI: 10.48550/arxiv.1910.03581.

Li, T., Sahu, A. K., Talwalkar, A., and Smith, V. (2020). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60. DOI: 10.1109/msp.2020.2975749.

Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151. DOI: 10.1109/18.61115.

Lin, Y., Bilotti, E., Bastiaansen, C. W., and Peijs, T. (2020). Transparent semi-crystalline polymeric materials and their nanocomposites: A review. *Polymer Engineering & Science*, 60(10):2351–2376. DOI: 10.1002/pen.25489.

M., J. J., K., D., F., K., P., A., D., G., and K., P. (2024). Enhanced generalization performance in deep learning for monitoring driver distraction: A systematic review. *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 48(1):137–151. DOI: 10.37934/araset.48.1.137151.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. (2023). Communication-efficient learning of deep networks from decentralized data. DOI: 10.48550/arxiv.1602.05629.

Miller, L., Navarro, P. J., and Rosique, F. (2025). Edgenet: An end-to-end deep neural network pretrained with synthetic data for a real-world autonomous driving application. *Sensors*, 25(1). DOI: 10.3390/s25010089.

Parzen, E. (1962). On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076. DOI: 10.1214/aoms/1177704472.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. DOI: 10.48550/arxiv.1606.03498.

Schwarting, W., Alonso-Mora, J., and Rus, D. (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 1(1):187–210. DOI: 10.1146/annurev-control-060117-105157.

Shaker, B., Pietro Rosati Papini, G., Saveriano, M., and Liang, K.-Y. (2024). Generating synthetic vehicle data using decentralized generative adversarial networks. *IEEE Access*, 12:138076–138085. DOI: 10.1109/ACCESS.2024.3428369.

Souza, A., Bittencourt, L., Cerqueira, E., Loureiro, A., and Villas, L. (2023). Dispositivos, eu escolho vocês: Seleção de clientes adaptativa para comunicação eficiente em aprendizado federado. In *Anais do XLI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 1–14, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbrc.2023.499.

Tamayo-Urgilés, D., Sanchez-Gordon, S., Valdivieso Caraguay, □. L., and Hernández-Álvarez, M. (2024). Gan-based generation of synthetic data for vehicle driving events. *Applied Sciences*, 14(20). DOI: 10.3390/app14209269.

Tavakoli, A., Kumar, S., Guo, X., Balali, V., Boukhechba, M., and Heydarian, A. (2021). Harmony: A human-centered multimodal driving study in the wild.

IEEE Access, 9:23956–23978. DOI: 10.1109/ACCESS.2021.3056007.

Tavakoli, A., Kumar, S., Guo, X., Balali, V., Boukhechba, M., and Heydarian, A. (2023). Harmony. Available at: `https://osf.io/zextd` Retrieved from osf.io/zextd.

Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling tabular data using conditional gan. In *Advances in Neural Information Processing Systems*. DOI: 10.48550/arxiv.1907.00503.

Zhang, C., Yu, S., Tian, Z., and Yu, J. J. Q. (2023). Generative adversarial networks: A survey on attack and defense perspective. *ACM Comput. Surv.*, 56(4). DOI: 10.1145/3615336.

Zhao, H., Wang, Y., Bashford-Rogers, T., Donzella, V., and Debattista, K. (2024a). Exploring generative ai for sim2real in driving data synthesis. In *2024 IEEE Intelligent Vehicles Symposium (IV)*, pages 3071–3077. DOI: 10.1109/IV55156.2024.10588493.

Zhao, S., Liao, T., Fu, L., Chen, C., Bian, J., and Zheng, Z. (2024b). Data-free knowledge distillation via generator-free data generation for non-iid federated learning. *Neural Networks*, 179:106627. DOI: 10.1016/j.neunet.2024.106627.

Zhu, Z., Hong, J., and Zhou, J. (2021). Data-free knowledge distillation for heterogeneous federated learning. DOI: 10.48550/arxiv.2105.10056.