Azar Azad

Federated Learning for Autonomous Vehicles: Privacy-Preserving Edge AI

Metropolia University of Applied Sciences

Master of Engineering

Information Technology

Master's Thesis

25 May 2025

PREFACE

This thesis marks the culmination of a significant chapter in my academic and personal journey. It would not have been possible without the unwavering support, encouragement, and love of the people closest to my heart.

First and foremost, I express my deepest gratitude to my beloved spouse, whose patience, understanding, and constant encouragement gave me the strength to stay focused and determined through every challenge. Your presence has been my source of peace and motivation, and your belief in me has meant more than words can express.

To my dear mother, whose boundless love and sacrifices have shaped who I am today, thank you for your endless prayers and faith in my abilities. You have always been my first teacher and my greatest source of inspiration.

And to my father, whose wisdom, guidance, and quiet strength have been invaluable throughout my life. Your confidence in me has continually driven me to aim higher and persevere through every difficulty.

This work stands as a humble token of appreciation for the love, sacrifices, and unwavering support you have each given me. I am forever grateful.

Espoo, Finland, 27/05/2025

Azar Azad

Abstract

Author:	Azar Azad
Title:	Federated Learning for Autonomous Vehicles: Privacy-Preserving Edge AI
Number of Pages:	78
Date:	25 May 2025
Degree:	Master of Engineering
Degree Programme:	Information Technology
Professional Major:	Networking and Services / Medical Technology
Supervisors:	Peter Hjort, Senior Lecturer

Federated Learning (FL) offers an appropriate approach to utilizing the huge amount of data created by Autonomous Vehicles (AVs). Besides this ability, it focuses on preserving data privacy while working on it. This thesis conducts a thorough, review-based analysis of the use of federated learning in autonomous vehicles as a privacy-preserving method of edge AI. The discussion is initiated by identifying the primary technical and ethical challenges. These issues include data heterogeneity across different driving situations, limitations in vehicle resources, and the necessity for fairness and transparency in model updates.

This work examines various privacy-enhancing techniques—Differential Privacy, Secure Multi-Party Computation, Homomorphic Encryption, and Secure Aggregation—to evaluate their trade-offs in protecting sensitive information while preserving model utility. The survey then analyses communication solutions specifically designed for automotive networks, including model compression techniques (Quantization, Pruning) and asynchronous updating methodologies, as well as Hierarchical and Decentralized Federated Learning architectures. Moreover, the survey tries to emphasize the effectiveness of these techniques in mitigating bandwidth and latency constraints.

One of the significant purposes of this survey is to emphasize the issue of non-IID data. To address this issue, the survey examines techniques such as FedProx, SCAFFOLD, and Clustered FL that are beneficial to resolve client drift and enhance convergence when local training datasets exhibit distributional discrepancies.

Another topic that has been addressed in this thesis is personalization strategies, including Transfer Learning and client-specific adaptation. These techniques are helpful to balance global knowledge with local complexity. The survey explores existing security concerns, which include adversarial and Byzantine attacks, and tries to mention some mitigating strategies through robust aggregation and blockchain-supported federated learning frameworks.

The results of these investigations lead to specific recommendations: the necessity for communication-efficient client selection, adaptive privacy controls tailored to vehicular situations, realistic simulation benchmarks (e.g., FLEXE), and standardized industry protocols. This thesis clarifies the transformational potential of FL for AVs by integrating several elements and identifies research directions, encompassing algorithms, systems, and ethics, that influence the next generation of safe, efficient, and privacy-conscious autonomous transportation.

Keywords: Federated Learning (FL), Autonomous Vehicles (AVs), Edge AI, Privacy-Preserving Machine Learning, non-IID Data, Communication Efficiency, Model Personalization, Secure Aggregation, Differential Privacy

---

The originality of this thesis has been checked using the Turnitin Originality Check service.

Contents

List of Abbreviations

Abbreviation	Definition
AFL	Adaptive Federated Multi-Task Learning
AI	Artificial Intelligence
AVs	Autonomous Vehicles
BDFL	Byzantine-Fault-Tolerance Decentralized Federated Learning
CAV	Connected Autonomous Vehicle
CCPA	California Consumer Privacy Act
CFL	Centralized Federated Learning
CFL	Clustered Federated Learning
CNN	Convolutional Neural Network
DFL	Deep Federated Learning
DFL	Decentralized Federated Learning
DNN	Deep Neural Networks
DP	Differential Privacy
DPAFL	Differential Private Asynchronous Joint Learning
E2E FL	End-to-End Federated Learning
F2L	Full-stack FL
FADNet	Federated Autonomous Driving Network
FD	Federated Distillation
FedAvg	Federated Averaging
FedIoT	Federated Learning for IoT
FedPcf	Federated Learning Framework with Multi-Level Prospective Correction Factor
FedProx	Federated Proximal
FL	Federated Learning
FLEXE	A Simulation Environment for Federated Learning Experiments in CAVs
FTL	Federated Transfer Learning
GDPR	General Data Protection Regulation
HE	Homomorphic Encryption
HFL	Hierarchical Federated Learning

Abbreviation	Definition
IID	Independent and Identically Distributed
IoT	Internet-of-Things
IoV	Internet of Vehicles
LKD	Label-Driven Knowledge Distillation
MEC servers	Mobile Edge Computing servers
ML	Machine Learning
MCS	Mobile Crowdsensing
OAC	Over-the-Air Computation
RL	Reinforcement Learning
RSUs	Roadside Units
SCAFFOLD	Stochastic Controlled Averaging for Federated Learning
SGD	Stochastic Gradient Descent
SemCom	Semantic Communication
SMPC	Secure Multi-Party Computation
UAV	Unmanned Aerial Vehicle
V2X	Vehicle-to-Everything Communication

1 Introduction

The emergence of Automotive Vehicles (AVs) represents a huge shift in transportation. It offers the potential to increase safety, enhance efficiency, and improve mobility(1). At the center of their functionality lies sophisticated Artificial Intelligence (AI) and Machine Learning (ML) algorithms, which process vast amounts of sensor data to enable perception, decision-making, and navigation without human intervention(1). By increasing the deployment of AVs, the amount of data generated by them is growing exponentially. Leveraging this huge decentralized data for training machine learning models is becoming complicated and presents both opportunities and challenges.

Traditional centralized machine-learning approaches require the collection of data in a central location, which raises severe data privacy and security risks(2). To resolve these issues, Federated Learning (FL) has emerged as a viable and promising distributed learning framework that enables collaborative model training through decentralized edge devices, such as AVs, while keeping the raw data localized on each device(2-5).

This means that each AV independently trains its local model using its sensors and computational resources at the edge, and instead of sharing raw data, only sharing model updates (e.g., gradients and model parameters) is transmitted to a central aggregator. Using a decentralized paradigm, FL leverages data from AVs in a privacy-preserving manner and reduces bandwidth consumption and development costs while ensuring compliance with data protection regulations such as GDPR and CCPA(2-6).

This thesis review-based thesis investigates the following research questions:

1. Which privacy-preserving techniques are most effective for FL in AV systems?

2. How does FL impact communication efficiency compared to traditional machine learning (ML) models?

3. What are the key challenges and solutions for maintaining model accuracy in FL for AVs with non-independent and identically distributed (non-IID) data distributions?

4. What are the technical and ethical challenges in implementing federated learning (FL) on autonomous vehicles (AVs)?

5. What are the major research gaps and future directions for FL in AV applications?

   This thesis aims to identify the gaps and propose potential future research directions that can help enhance this field.

To address these research questions, this thesis pursues the following key objectives:

1. Conduct a comprehensive review of FL applications in AVs.

2. Investigate privacy-preserving techniques in AV systems that use FL.

3. Assess FL efficiency in AV communications and computations.

4. To identify and analyze the key challenges affecting model accuracy in FL under non-IID data conditions and assess existing solutions in the literature to mitigate these challenges.

5. Provide recommendations for future research in FL-based AV systems.

2 Method and Materials

By adopting a review-based approach for this thesis as its methodology, the thesis involves systematically searching and analyzing existing literature on FL in AVs.

2.1 Research questions

The methodology that is utilized in this thesis is an approach that is based on reviews. A comprehensive search and analysis of the existing literature on FL in AVs is required to accomplish this.

This thesis, which is based on a review, intends to investigate the use of FL in the field of autonomous vehicles, with a particular emphasis on the essential component of preserving privacy using edge AI. This thesis addresses the following research questions:

1. **Which privacy-preserving techniques are most effective for FL in AV systems?**

   To further strengthen security, additional privacy-preserving mechanisms can be utilized, despite the fact that FL inherently provides a certain degree of privacy through keeping data localized. This thesis investigates the effectiveness of various methods, such as differential privacy(7-10), secure aggregation protocols, and knowledge distillation in the context of FL for AVs(11).

2. **How can FL impact communication efficiency compared to traditional ML models?**

   In distributed learning systems such as FL, communication efficiency is an essential component, particularly in networks with limited bandwidth and vehicle networks (4, 5, 7, 9, 10, 16, 17). Evaluation of the impact of FL on communication efficiency in AV application and comparing this impact with the traditional centralized training approach is one of the purposes of this thesis.

3. **What are the key challenges and solutions in maintaining model accuracy in FL for AVs with non-IID (non-independent and identically distributed) data distributions?**

   Due to variations in driving environments, driver behaviours, and sensor characteristics, data collected by Avs is likely to be non-IID(3, 10, 12-14). The shared model in FL may experience substantial challenges in terms of accuracy and convergence due to this heterogeneity. Identifying the key challenges that emerged from non-IID data in FL for AVs and assessing existing solutions proposed in the literature to mitigate these risks is another purpose of this thesis(10, 12, 14).

4. **What are the technical and ethical challenges in implementing FL on AVs?**

   The implementation of FL in autonomous vehicles (AVs) faces a number of technical challenges, including the heterogeneity of data that results from a variety of driving and environmental conditions and sensor configurations(3, 12, 13), resource limitations of edge devices in terms of computation and communication(12, 15) and the requirement to guarantee the reliability and security of the learning processes in a dynamic environment(3, 15, 16). In addition, the ethical problems that surround the use of data and the level of fairness of models require a thorough assessment.

5. **What are the major research gaps and future directions in FL for AV applications?**

   This thesis aims to identify the gaps and propose potential future research directions that can help enhance this field.

2.2 Research objectives

This thesis tries to answer these research issues by doing the following:

1. **Conduct a comprehensive review of FL applications in AVs.** This means looking at the existing research to understand how FL is being used in various AV tasks, such as lane detection(5), predicting traffic flow(11), and autonomous driving itself(1, 2).

2. **Investigate privacy-preserving techniques in AV systems that use FL.** Surveying and analysing different privacy-preserving methods that could be integrated with FL in AV systems is another objective of this thesis.
3. **Assess FL efficiency in AV communications and computations.** This goal involves analysing research examining FL's communication overhead and computing requirements in AVs. Moreover, it also focuses on finding ways to mitigate these challenges.
4. **To identify and analyze the key challenges affecting model accuracy in FL under non-IID data conditions and assess existing solutions in the literature to mitigate these challenges.** This section works on finding the challenges that working with AV's heterogeneous data and FL may face. Also, suggesting some solutions to tackle those challenges is another purpose of this thesis.
5. **Provide recommendations for future research in FL-based AV systems.** Finally, the result of those investigations, which is shown in the thesis, suggests possible directions for more research. These research topics could help FL move forward in AVs.

2.3 Research Design and Methods

The methodology consists of the following steps:

1. **Literature selection**: Related articles on the subject of utilizing FL in AVs from reputable resources are selected. These resources include academic journals, industry whitepapers, and technical reports from sources such as IEEE Xplore, Google Scholar, and ScienceDirect.
2. **Thematic categorization**: In the next step, the selected literatures are categorized into the following related concepts:
    a. Privacy-Preserving Techniques in FL for AVs
    b. Communication efficiency and network optimization
    c. Model accuracy and performance with non-IID data
    d. Security challenges and counteractions
3. **Comparative discussion and analysis**: it is conducted to:

a. Analyze and compare existing FL frameworks
b. Practical Implications for AV and FL Integration
c. propose insights for research gaps and future opportunities.

2.4 Research Outcomes

The expected outcomes of this thesis include:

1. A comprehensive review of FL's impact on privacy, security, and communication efficiency in AVs.
2. A comparative analysis of FL frameworks and techniques and their use in AV applications.
3. An overview of available frameworks for FL in AVs.
4. Recommendations for improving FL Architectures in AV Applications.
5. Identifying research gaps and open challenges in FL for AVs.

This thesis aims to provide significant perspectives into the current state of Federated Learning for Autonomous Vehicles by addressing these objectives and questions through a comprehensive examination of the literature. It offers this by highlighting the potential of these technologies for privacy-preserving edge AI as well as identifying key challenges and potential future research directions.

3 Theoretical Literature Review

This chapter presents the concepts used in this study.

3.1 Fundamentals of Federated Learning in Autonomous Vehicles

3.1.1 Overview of Federated Learning

Federated Learning (FL) is a decentralized machine learning paradigm that allows several clients (such as autonomous vehicles) to work together to train a model without sharing their raw data(1, 3, 13). This can be done through peer-to-peer communication (in Decentralized FL) or under the coordination of a central server (in Centralized FL)(1). Traditional centralized machine learning, in which all training data is gathered and kept on a single server, is in contrast to this distributed method(1). In FL, only model updates- such as gradients or model parameters – are sent to the aggregator; the training process occurs locally on each device(1). This fundamental difference is essential for preserving data privacy and security.

FL was first created to protect the privacy of each learner's data while leveraging distributed data among learners(4). When data is sensitive, proprietary, or subject to a legal restriction that limits its centralization, it is more attractive. The primary goal of FL is to use distributed data from numerous clients to train a high-quality global model(17).

Key characteristics of Federated Learning:

- **Decentralized Training**: Model training across local devices (e.g., within each autonomous vehicle) using their own data(1).
- **Local Model Updates**: Each participating client updates its local model based on its local datasets(1).
- **Model Aggregation:** To produce a new global model, the model updates received from the clients are combined by a central server (in a Centralized FL) or a peer network (in a Decentralized FL)(1, 8). Common aggregation method includes Federated Averaging (FedAvg).(4, 5, 8, 18)

- **Privacy Preservation:** To improve privacy and reduce data exposure, raw client data is kept on local devices and is never shared with the server or other clients(1, 2, 11, 17).
- **Iterative Learning**: Until the global model converges to a satisfactory performance level, the local training and global aggregation processes are repeated across multiple communication rounds(1, 18).
- **Heterogeneity Handling:** FL frameworks must interact with system heterogeneity (differences in network connectivity, computational resources) and heterogeneous data (non-IID data distributions) between clients(1, 3, 7, 13).

The overview of the FL was shown in Figure 1.(19)

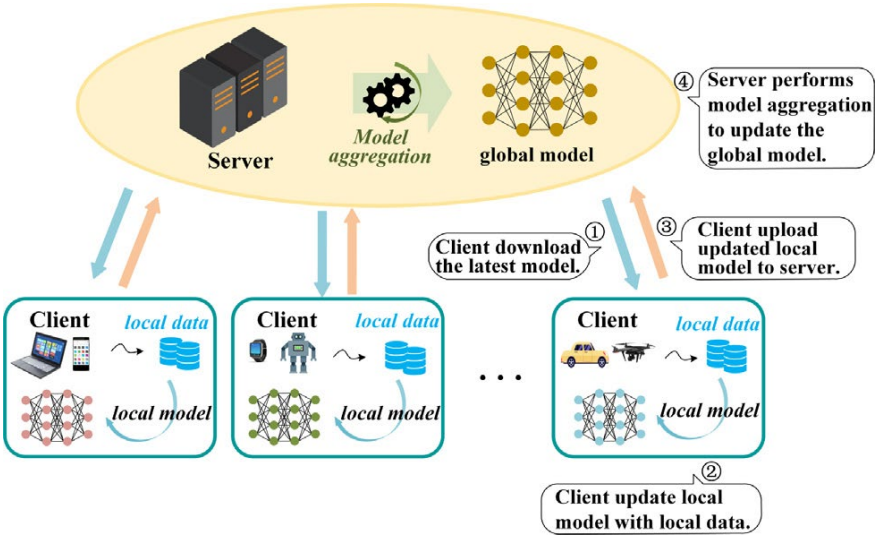

Figure 1 - The general overview of the Federated Learning Framework(19)

Federated Learning Settings:

Federated learning can be broadly categorized into two main settings(7):

- **Cross-silo FL:** This type of setting usually consists of fewer big, well-resourced participants (such as several automotive manufacturers or fleet operators). Most clients might participate in each round and maintain their state between rounds(7).
- **Cross-device FL:** This is often the more difficult scenario with many resource-constrained edge devices (such as individual autonomous

vehicles). Clients usually cannot preserve state across rounds, and just a small percentage of the clients may engage in each round(7). Given the nature of autonomous vehicles, FL in this area often fits effectively with the cross-device environment because of the dynamic network connectivity and potentially a large number of participating vehicles.

3.1.2 Motivation for FL in AVs

Various key variables are driving the usage of FL in autonomous vehicles, including the following:

Data Privacy and Security: A huge amount of sensitive data regarding locations, driving patterns, and sensor readings is generated by AVs(1, 11). FL provides a mechanism to learn from this data without affecting the privacy of fleet operators or individual vehicle owners(1, 2, 4, 11). Given growing regulatory scrutiny and user concerns about data privacy, this is especially crucial(1).

- **Handling Heterogeneous Data:** Since AVs operate under a variety of dynamic real-world circumstances, the data they gather varies significantly(4, 7, 13). Different geographic locations, weather patterns, traffic patterns, and sensor configurations are examples of these differences(4). By learning from each vehicle's local data and combining the acquired knowledge into a global model that is more resilient and generalizable under various circumstances, FL is designed to handle such non-IID data distributions(11).
- **Scalability and Adaptability:** A scalable and flexible learning framework is crucial as the number of connected autonomous vehicles(CAVs) on the road rises(1). Without needing the centralization of enormous datasets, FL enables autonomous driving systems to scale effectively, supporting a large number of vehicles and edge devices for collaborative learning(1).
- **Leveraging Edge Computing:** Modern AVs have ever-increasing processing capacity(20). By allowing model training directly on these vehicles, FL supports the trend of pushing network computing on the

edge(2, 20). This minimizes communication delay and reduces the need for a centralized server(1, 2).

- **Continuous Learning and Adaptation:** AVs operate in a continually changing environment(17). By enabling vehicles to collaboratively adjust their model in response to newly encountered data, FL promotes ongoing learning and adaptation, which eventually improves performance and safety(2, 17).

3.1.3 FL process in the context of AVs

The typical FL process can be beneficial for autonomous vehicles as follows:

1. **Global model initialization:** The global machine learning model is initiated by a central server (or, in the case of Decentralized FL, by a distributed mechanism)(11). The primary model may be initialized at random or through pre-training.
2. **Client Selection:** A subset of the available AVs is chosen to take part in the training process for each learning round(7, 11, 13). Strategies for choosing clients take into account variables including contribution history, data quality, and resource availability (communication bandwidth and computation)(15, 20).
3. **Local Training**: Using their own specific driving data, the chosen vehicles download the most recent global model and train it locally(1, 4, 11). Usually, one or more epochs of the chosen optimization algorithm, such as stochastic gradient descent, are used for local training.
4. **Model Update Transmission**: Each participating vehicle sends back to the central server (or, in the decentralized setting, to the nearby vehicles) its updated model parameters or gradients (edge devices/clients send the gradients or learnable parameters) following local training(1, 4, 11). Raw data remains on the vehicle.
5. **Global Model Aggregation**: The information sent back from the clients is not always the complete set of updated model parameters. Some clients sent back their updated model parameters(4, 21), while others sent back their gradients(2, 11), The received model updates from participating

vehicles are combined by the central server or aggregator (or, by a distributed aggregation mechanism)(1, 4, 11) to produce an updated global model(1, 2, 21). Federated Averaging (FedAvg) is a popular aggregation method in which the server calculates a weighted average of the model parameters, with weights often determined by the number of data samples on each client(4, 15).

6. **Global Model Distribution**: The participating vehicles and potentially additional available vehicles in the network receive the recently aggregated global model(11).
7. **Iteration**: Until the global model achieves the desired level of performance, steps 2-6 are carried out repeatedly for several communication rounds(1, 4).

3.1.4 Local Update Aggregation Strategies

The aggregation process combines local updates or corrections to form a better global model. There are several methods to reach this aim:

1. **Averaging (FedAvg):** This method is known as the most commonly used aggregating technique(21). The server usually begins a typical round by sending the current global model to a subset of clients that have been chosen individually(15, 21). After that, every client trains this model with its own local data, typically by carrying out many steps of stochastic gradient descent (SGD) on their own local dataset(21). The clients then communicate their updated local models back to the server after they have completed their local training(21). In FedAvg, the difference between a client's locally trained model and the initial global model is termed the local update(7). The average of these local updates across participating clients is then used to define a "pseudo-gradient", specifically its negative. The server updates the global model by applying a process equivalent to a Stochastic Gradient Descent (SGD) step using this "pseudo-gradient"(7). The server then typically takes the average of the updated local models that have been received from the clients. The contribution of each client's

update is weighted based on the amount of data it contains, which is known as a weighted average and is frequently employed(5, 9, 22).

2. **Aggregating Local Updates/Differences**: To be more specific, the server aggregates the local updates, particularly in scenarios where the emphasis is set on the efficiency of communication(23). At the beginning of the round, the global model was received, and these local updates show the difference between the locally trained model and the global model(23). The global model is then adjusted based on the aggregated update that was just presented(23).
3. **Treating Updates as Pseudo-Gradients:** For the purpose of the server-side optimization process, the aggregated local update is treated as a pseudo-gradient in advanced federated optimization frameworks such as FEDOPT(21, 23). This framework includes FedAdam, FedAdagrad, and FedYogi as special cases. According to the history of these pseudo-gradients, the server is able to employ adaptive optimization algorithms such as Adam, Adagrad, or Yogi, which adjust learning rates based on the data(21, 23). These client updates are not actual gradients, but rather "pseudo-gradients," which are difficult to analyze due to the possibility of high bias and variance(7).
4. **Combining with Server-Side Optimizers:** Frameworks such as FedPcf integrate client updates with server-side optimization. A "global prospective correction factor" is utilized by the server in order to speed up the process of aggregation and to provide direction for the global model update(24). This factor accounts for historical momentum and prospective gradient information that is produced from the process of aggregating client models. The combined historical momentum and the prospective direction that is calculated from the aggregated client models are used to construct the updated global model(24).

As mentioned, in the Global Model Aggregation phase, the server receives information that represents the corrections or local updates learned by each client's model during local training(1, 23). This information, which can be in the form of updated parameters, gradients, or compressed updates, is then aggregated. The most common method, FedAvg, averages these corrections

(or the resulting updated models)(20, 21). More advanced methods treat the aggregated correction as a pseudo-gradient for server-side optimization(21, 23) or use sophisticated mechanisms like prospective correction factors to guide the aggregation process(24). Communication efficiency techniques further focus on transmitting compressed representations of these corrections or gradients(9, 18, 21).

3.1.5 FL Frameworks and Architectures for AVs

Several FL architectures have been suggested for AV applications. Network constraints, computational resources, and security requirements all influence the selection of an architecture(25).

- **Centralized Federated Learning**
  By collecting model updates from AVs, combining them, and redistributing the refined model, a global server manages the learning process in a centralized FL. Although this method ensures effective model convergence, it is vulnerable to communication bottlenecks and single-point failures(26). Due to being vulnerable to malicious activity(17) Communication bottlenecks, intentionally dishonest aggregation, unintentional network connection failure, or unforeseen external attacks on the aggregator(17), the dependence on a single central server in Central Federated Learning causes a potential point of failure in the learning process(11).
- **Decentralized Federated Learning (DFL)**
  There is no central server in DFL. Instead, vehicles aggregate locally or collaboratively and directly share model updates with each other (e.g., through vehicle-to-vehicle (V2V) communication with other vehicles in the surrounding)(11). By eliminating the central server‘s single point of failure, DFL can increase robustness and make it more suitable for extremely dynamic vehicle networks, but it requires robust peer-to-peer communication protocols to ensure convergence(11). In Decentralized Federated Learning (DFL), failures or faulty behavior in nodes, such as unreliable communication within dynamic network topologies, can hinder

convergence and cause network latency, necessitating robust peer-to-peer protocols for parameter synchronization and adaptation to these network changes(5, 11). Regarding communication costs, DFL eliminates the central server and its associated congestion by having nodes share model updates directly with neighbors through peer-to-peer communication(11, 27). This decentralized model sharing significantly reduces the communication burden on a single central point and aims to improve efficiency compared to centralized federated learning by reducing the overall data transmission directed at a server(9, 27), with one asynchronous DFL strategy reporting a 60% bandwidth cost reduction compared to synchronous protocols(5, 27).

Figure 2 represents Centralized FL and Decentralized FL There is a server that is located in the center of a Centralized FL system (a), which results in the formation of a star network topology. This central server functions as a hub for multiple clients to connect to in order to aggregate and synchronize their models. On the other hand, a Decentralized FL system (b) does not have a central server. Instead, clients engage in peer-to-peer (P2P) communication with one another, which results in the formation of a mesh network topology within the network(19).

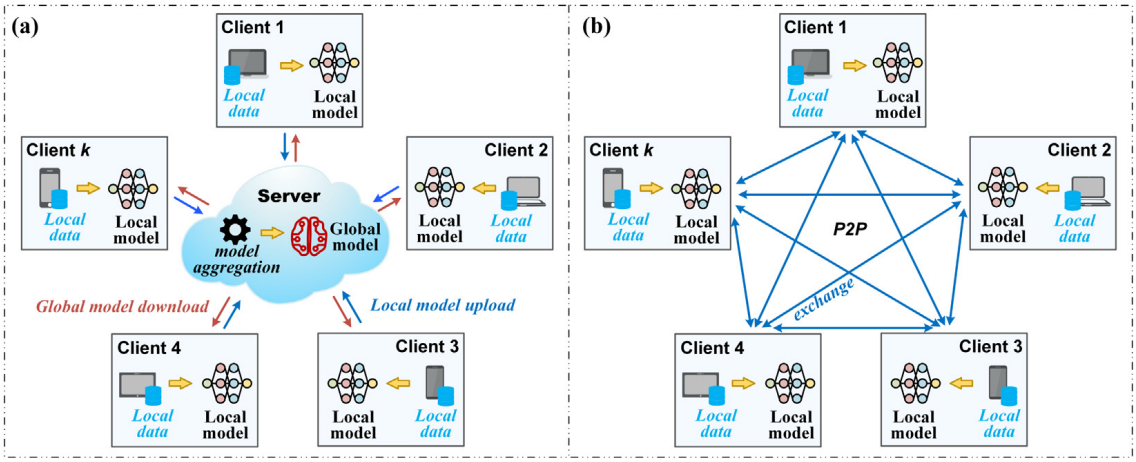

Figure 2 - Centralized Federated Learning (a), and Decentralized Federated Learning (b)(19)

- **Hierarchical Federated Learning (HFL)**

AVs as edge devices, which serve as a bridge between AVs and the global server, might aggregate their updates at a regional or cluster level (e.g., Roadside Units, or RSUs) using hierarchical FL, a multiple-tiered architecture instead of AV-to-cloud connection(10). These intermediate aggregations are then further aggregated at a global server. This can increase the efficiency and scalability, and reduce communication costs and latency of large-scale vehicular networks(7, 12).

Figure 3 shows the Hierarchical Federated Learning. The flowchart illustrates the hierarchical aggregation process that occurs through the federated Internet of Vehicles (IoV). Vehicles are grouped together into distinct clusters, and then they upload their local models to the edge servers that correspond to them (i.e., roadside units). These edge servers subsequently upload the models to the cloud in order to do additional global model updates(19).

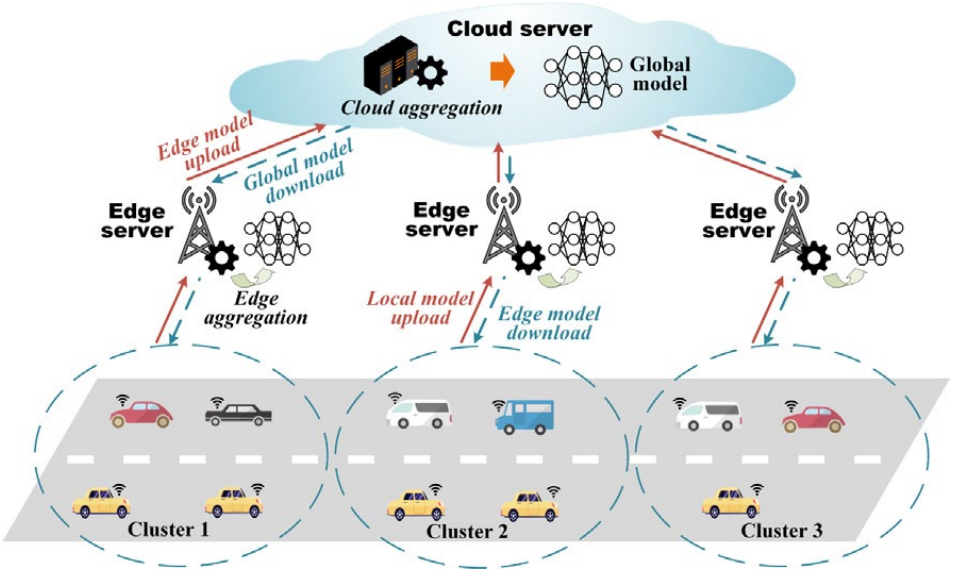

Figure 3 - Hierarchical aggregation in FL(19)

- **Federated Transfer Learning (FTL)**

  Federated Transfer Learning (FTL) is a strategic combination of the principles of both Transfer Learning and Federated Learning (FL). When applied in distributed environments, such as networks of autonomous vehicles, this strategy is especially useful since it makes it easier to

transfer pre-trained models or knowledge from one vehicle or a group of vehicles to other vehicles that are part of the federated network. The fundamental reason for this knowledge transfer is to solve the issues that are brought about by the limited availability of local data on individual vehicles. When vehicles with sparse datasets become capable of leveraging the knowledge of other participants, they are able to considerably benefit from the broader experience that is captured in the transferred models (shown in Figure 4). This results in faster convergence and improved performance across the whole vehicle fleet. The core concept of FTL is to minimize the necessity for extensive training on individual vehicles and to promote the efficient sharing of knowledge among interconnected vehicles. Furthermore, it aims to maintain the privacy advantages intrinsic to FL by avoiding the direct exchange of raw data(1, 2, 9).

The development of simulation frameworks like FLEXE supports the investigation and application of FTL techniques in the context of Connected and Autonomous Vehicles (CAVs)(20). Additional strategies for transferring knowledge, such as Federated Distillation (FD), are also being investigated within the context of CAV environments. This can potentially reduce the size of the communication and help in handling challenges such as non-IID data distributions that are common in vehicular scenarios(5). In contrast to traditional parameter-sharing in FL, FD frequently involves the exchange of model outputs (logits) rather than weights. Some techniques, such as label-driven knowledge distillation, are utilized to refine this further. These techniques concentrate on the transmission of knowledge based on the reliability of model predictions for particular data characteristics(19).

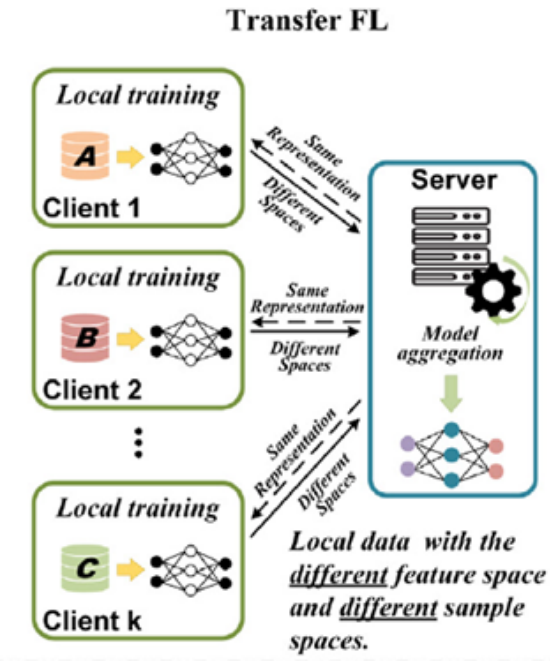

Figure 4 - Federated Transfer Learning (FTL)(19)

- **Deep Federated Learning (DFL)**

  Deep Federated Learning (DFL) is presented as a new methodology in the concept of privacy-preserving. It is specifically tailored for training deep neural networks in a decentralized environment(5, 28). It is highlighted as a groundbreaking approach, particularly for training policies in areas like autonomous driving and Connected and Autonomous Vehicles (CAVs)(5, 28).

  The DFL system is based on a peer-to-peer architecture, and by this means, it is fully decentralized(5, 28). This design is a crucial differentiation since it eliminates the requirement for a central server or orchestration for model aggregation(5, 28). This acts completely opposite to the usual star-network topology that is frequently observed in Centralized Federated Learning (FL)(28). Instead, it uses a technique that involves reaching a consensus among the devices that are participating, such as automobiles, in order to update the global parameters(28).

  The DFL idea prioritizes privacy substantially. It achieves this by assuring data localization, meaning that training occurs locally on the devices,

without data being gathered or stored on a central server. This technique effectively preserves user privacy(5, 28).

In the context of the DFL paradigm for autonomous driving, customized architectures such as the Federated Autonomous Driving Network (FADNet) are utilized(5, 28). This technique (FADNet) aims to enhance model stability, ensure convergence, and address challenges relating to imbalanced data distribution. Means that this technique was developed with the intention of being well-suited for federated training(5, 28). It is the responsibility of individual nodes, also known as clients, to train models locally using their own datasets. After that, they selectively share crucial model updates with other nodes(28).

The DFL provides a number of benefits, including improved accuracy and precision, as well as the potential to assist in reducing the risk of communication congestion(28). Experimental results, in particular those obtained with FADNet, indicate that superior accuracy is achieved compared to that of recent methods, and that communication efficiency has been improved(5, 28). In general, Decentralized Federated Learning approaches investigate ways to reduce the amount of communication overhead in large networks(1).

Figure 5 represents peer-to-peer Deep Federated Learning in which red arrows denote the aggregation process between silos. Yellow lines with a red cross indicate the non-sharing data between silos(28).

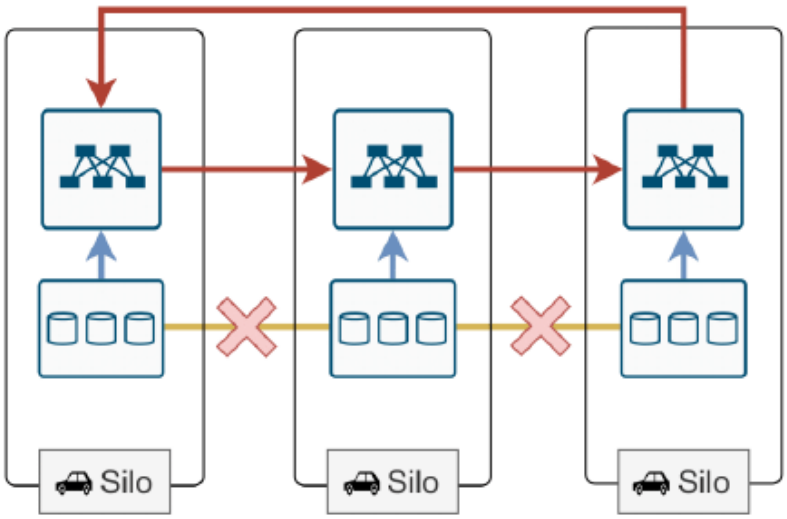

Figure 5 – Peer-to-peer Deep Federated Learning graph

- **End-to-End Federated Learning (E2E FL)**

  End-to-end federated Learning (E2E FL) is generally explored in relation to the process of training deep learning models for autonomous vehicles (AVs)(1, 29). Within the framework of Federated Learning (FL), it is described as a particular strategy that offers advantages in terms of preserving prediction accuracy and enhancing privacy(1).

  The use of E2E FL to address "classic end-to-end learning problems" in the automotive industry is a significant feature of this type of learning(29). It is used in tasks such as the prediction of steering wheel angle directly from sensory input(1, 29). This concept expands upon end-to-end learning, which is a process wherein a model directly maps raw inputs (such as photos) to control outputs, without the need for intermediate processes such as object detection or semantic segmentation(29). In the context of the E2E FL framework, this entails the utilization of deep architectures, such as a two-stream deep Convolutional Neural Network (CNN)(1), which are trained locally on the edge devices, which are the cars themselves(1, 29).

  One of the benefits of the E2E FL strategy is that it operates in real time and addresses the limitations of traditional synchronous FL methodology(29). The use of this technique makes synchronous aggregation techniques rigid for dynamic and heterogeneous environments, such as those experienced by autonomous vehicles (AVs) with fluctuating hardware and network circumstances(29). Synchronous aggregation protocols necessitate waiting for all devices. Asynchronous model aggregation procedures are incorporated into the real-time E2E FL approach(29). This involves edge vehicles computing model updates locally and selecting when to communicate based on parameters such as frequency-bound limits and model version variations. The results of using this method, especially, could be seen in the elimination of the need to wait for all participants to arrive simultaneously(29). It is possible to handle real-time streaming data with this approach, which may involve using techniques such as a sliding training window(29).

The E2E FL approach has several benefits in the AV industry. The ability to achieve prediction accuracy that is equivalent to or even better than that of centralised learning or synchronous FL, and a significant reduction in training time and bandwidth consumption, are the benefits that could be mentioned in the realm of using E2E FL in AVs(29, 30). The aim is to achieve production-ready deployment of FL for crucial automotive applications, as evidenced by focusing on incorporating asynchronous aggregation and handling real-time data(29, 30).

A solution such as Federated Learning for IoT (FedIoT) was developed to implement federated learning algorithms on IoT devices, which can be highly relevant to the edge computing component of autonomous vehicles(31).

Simulation environments like FLEXE provide an opportunity for researchers to investigate vehicular federated learning mechanisms in actual scenarios, which include the aspect of CAV (Connected Autonomous Vehicles) mobility and communication patterns among the factors to be considered(20).

Each of these architectures offers trade-offs in terms of performance, resilience, and expansibility in the availing of federated learning for automotive vehicle networks. A comprehensive overview of the Federated Learning frameworks is provided in Table 1:

Table 1 - Overview of Federated Learning Frameworks

Aspect	Centralized Federated Learning	Hierarchical Federated Learning (HFL)	Decentralized Federated Learning	Federated Transfer Learning (FTL)	Deep Federated Learning (DFL)	End-to-End Federated Learning (E2E FL)
Architecture	The central server aggregates model updates from all clients.	Multi-tier with local clients, intermediate servers (e.g., edge nodes), and a global server.	Fully distributed; clients communicate peer-to-peer (P2P) without a central server.	Combines FL with transfer learning; the central server or peers facilitate knowledge transfer.	Uses deep neural networks in a federated setup, typically with a central server for aggregation.	The entire ML pipeline (preprocessing, training, inference) is executed in a federated manner and is often centralized.

Aspect	Centralized Federated Learning	Hierarchical Federated Learning (HFL)	Decentralized Federated Learning	Federated Transfer Learning (FTL)	Deep Federated Learning (DFL)	End-to-End Federated Learning (E2E FL)
Data Handling	Data stays local; only model updates (gradients/parameters) are sent to the server.	Data stays local; intermediate servers aggregate updates before sending to the global server.	Data stays local; model updates are shared directly between clients.	Data stays local; it leverages pre-trained models or knowledge from related domains.	Data stays local; it focuses on training complex deep learning models across clients.	All data processing (including preprocessing) stays local; only model updates or encrypted outputs are shared.
Communication	Clients communicate with the central server; high server-client bandwidth is needed.	Clients communicate with intermediate servers, reducing global server load; tiered communication.	P2P communication: no central coordinator, but potentially higher latency.	Central server or peers share transferable knowledge; communication depends on the setup.	Similar to centralized FL, high communication large deep model updates.	It requires communication for all pipeline stages and often uses secure protocols to protect raw data.
Scalability	Scales moderately; bottleneck at central server for large systems.	Highly scalable due to a hierarchical structure that distributes computation.	Scales with network size but depends on efficient P2P protocols; complex in large networks.	Scales based on domain similarity and client participation depend on the FL backbone.	Limited scalability due to computational demands of deep models; server bottleneck possible.	Moderate scalability; preprocessing adds overhead, but secure protocols can streamline updates.

Aspect	Centralized Federated Learning	Hierarchical Federated Learning (HFL)	Decentralized Federated Learning	Federated Transfer Learning (FTL)	Deep Federated Learning (DFL)	End-to-End Federated Learning (E2E FL)
Advantages	- Simple to implement - Efficient for homogeneous data. - Strong global model consistency.	- Reduced communication overhead. Suits geographically distributed clients - Scalable.	- No central point of failure - Enhanced privacy - Resilient to server outages.	- Effective for heterogeneous data -Leverages existing models - Adapts to new tasks.	- Handles complex tasks (e.g., image recognition) - High model accuracy - Leverages deep learning advancements.	- Comprehensive privacy across the pipeline - Supports raw data processing - Flexible for diverse tasks.
Challenges	- Central server failure risk - High communication cost - Privacy risks from server breaches.	- Complex hierarchy management - Intermediate server reliability - Synchronization issues.	- Complex coordination - Higher P2P latency - Risk of model drift.	- Requires domain similarity. - Complex knowledge transfer - Potential overfitting.	- High computational demand - Large update sizes - Privacy for sensitive deep features.	- Complex implementation - High preprocessing overhead - Secure protocol complexity.
Privacy	Moderate; relies on encryption and trust in a central server.	High and intermediate aggregation reduce direct server exposure.	Very high; no central authority reduces aggregation risks.	Moderate to high; depends on FL structure and knowledge sharing.	Moderate; deep model updates may leak sensitive features if not encrypted properly.	Very high; encrypts all stages (preprocessing to inference), minimizing data exposure.

Aspect	Centralized Federated Learning	Hierarchical Federated Learning (HFL)	Decentralized Federated Learning	Federated Transfer Learning (FTL)	Deep Federated Learning (DFL)	End-to-End Federated Learning (E2E FL)
Computational Load	Clients train locally; the server aggregates updates.	Distributed across clients and intermediate servers, the global server load is reduced.	Fully distributed across clients; no server offloading.	Clients train locally; additional load for transfer learning adaptation.	High; clients need significant computing for deep models; a server aggregates extensive updates.	High; clients handle preprocessing and training; secure protocols add overhead.

3.1.6 Key Benefits of FL in AV Systems

Federated Learning provides a number of benefits for autonomous driving systems:

- **Enhanced Data Privacy and Security:** Without requiring a direct exchange of this raw data, FL allows for collaborative learning from the vast amount of sensitive data produced by AVs (e.g., driving patterns and sensor readings)(1, 2, 11, 17). It is essential to handle the privacy issue and adhere to the rules(1, 3).
- **Handling Heterogeneous and Non-IID Data:** Due to working AVs in a variety of real-world scenarios, their gathered data, which is non-independent and identically distributed data(Non-IID Data), varies greatly depending on the environment, sensor configurations, and driving conditions(7, 10, 11, 13). By locally training models on each vehicle and combining the learned knowledge into a more reliable global model, FL has been designed to learn from this heterogeneous data(1, 2, 11).
- **Improve Scalability and Adaptability:** FL offers a scalable framework for collaborative training that can effectively incorporate a large number of

vehicles without centralizing enormous datasets, which is important given the growing number of connected autonomous vehicles(1-3).

- **Leveraging Edge Computing Resources:** Contemporary AVs have a considerable processing capacity. By allowing model training directly on these vehicles, FL minimizes communication latency and reduces dependency on central servers, all of which are in line with the edge computing trends(2, 3, 20).
- **Continuous Learning and Adaptation:** Because of the dynamic driving environment, continuous learning is essential. FL makes it possible for AVs to work together to develop their models in response to new, locally encountered data, which continuously enhances performance, safety, and efficiency(1-3, 17).
- **Reduced Communication Overhead:** FL can significantly reduce the communication load on vehicular networks by sharing just model updates instead of a substantial amount of raw data(5, 11, 17).
- **Personalization and Adaptation to Local Environments:** Road conditions, traffic behaviors, and weather patterns vary depending on the regions in which AVs are operating. Unlike the one-size-fits-all ML models, FL enables models to adjust to local driving conditions, increasing system robustness(32).

3.1.7 Challenges in Implementing FL in AVs

Despite its advantages, FL implementation in AVs presents several challenges.

- **Non-IID Data:** Due to inherent diversity in driving conditions and sensor configuration, there are significant non-IID data distributions across vehicles, which can make it challenging for the global model's ability to converge(7, 10, 13). Various vehicles are designed to function in a variety of environments, including different types of roads, weather, and traffic conditions. These vehicles are further equipped with sensors that may change in terms of their type, number, placement, or quality during operation(5, 11, 12, 30). Because of these changes, the feature

distribution, label distribution, and overall properties of the data acquired by each vehicle are all different(12, 30). This heterogeneity indicates that the local datasets on various AVs are not statistically independent and do not have a distribution that is identical to one another(13, 22, 23, 30).

Moreover, AV data can be viewed as time series data(11, 15, 27) or data collected in dynamic environments(4, 5, 11). Data like vehicle speed, acceleration, steering angle, or sensor readings over time inherently possess temporal dependencies(27) and are not independent of previous observations (speed at time t is indeed related to speed at t-1)(3). Furthermore, the data distributions on devices can be evolving and change over time(3, 30), with data fed into the system in different rounds, potentially being Markovian(30). Clients often receive new data online and may only train on a single pass of this data(30). This issue is explored more in section 3.4.

- **System Heterogeneity:** System heterogeneity refers to the significant variations in the capabilities and resources available across the participating devices, which are the Autonomous Vehicles in this scenario(18, 30).

  System heterogeneity is the notable difference in the capabilities and resources accessible among the participating devices, the Autonomous Vehicles in this case. FL distributes the training process to many edge devices that are naturally varied, unlike training in a controlled, homogeneous data center.

  The key aspects of system heterogeneity underlined in the publications are:

  - **Computational Resources:** AVs vary greatly in their onboard computing power, including CPU, RAM availability, and maybe GPUs(30). This influences their capacity to carry out locally trained tasks requiring great computing effort(18). Vehicles with less computer capability could take much longer to finish their task(14).
  - **Communication Capabilities:** Connectivity in vehicular networks is quite dynamic and can change quickly(14). Due to mobility and environmental variables, AVs struggle with diverse bandwidth and

unstable connectivity, as well as different network access capabilities (e.g., 4G, 5G)(30). This affects the speed and reliability of sending model changes(14, 18).

- **Power Limits:** Though less important than for battery-powered devices like smartphones, the power availability of AVs might nonetheless change and perhaps influence their capacity to participate in demanding training sessions(18, 30).

- **Communication Reliability and Latency:** The interplay of different factors contributing to challenges in Federated Learning (FL) for Autonomous Vehicles (AVs). While non-IID data distributions are a significant hurdle(5, 30, 33), the mobility and network dynamics of vehicular environments are distinct, yet often interconnected, challenges that directly impact communication reliability and latency(5, 20).

  Because of mobility and network dynamics, vehicular networks often experience intermittent connectivity, fluctuating bandwidth, and latency issues, all of which could cause issues with FL training, such as communication overhead in vehicular networks(4, 5, 20).

  When it comes to autonomous vehicles (AVs), the setting of "mobility and network dynamics" indicates that:

  - **Mobility**: Autonomous vehicles are not stationary; rather, they are always moving within a dynamic environment(15). In order to accomplish this, their geographical location and closeness to other vehicles or roadside infrastructure (such as base stations or MEC servers) require continuous changes(20). Because of this movement, network nodes experience rapid and constant changes, and the amount of time it takes for vehicles to connect to network access points is very frequently very brief(4).
  - **Network Dynamics:** As vehicles move around, the communication environment that surrounds them is continuously changing(5, 20). The fluctuations in signal strength, the potential interference from other devices, the blockages caused by obstacles, and the overall topology of the network are all included in this explanation(4, 11, 20). The

characteristics of the communication channels in contexts involving vehicles are naturally dynamic(5).

The combination of CAV mobility and these dynamic network conditions results in several practical issues for FL:

- **Intermittent Connectivity:** Vehicles frequently experience unstable links, frequent drop-outs, and hand-overs between network coverage areas(4, 20). Therefore, it is possible for clients (vehicles) to become unavailable or disconnected while an FL training round is in progress(12, 18, 30).
- **Fluctuating Bandwidth and Latency:** The quality and speed of the wireless connection (bandwidth) might fluctuate quickly(20, 29). Because of this, the latency of communication is always unpredictable(4, 5, 20).
- **Increased Communication Overhead:** Because of the dynamic nature of the system, it is necessary to implement procedures that can manage connections, retransmissions, and handle drop-outs, which causes an increase in the overall communication burden(4, 5, 16). For synchronous FL, the delay caused by dropped or slow clients (the "straggler effect") can be substantial, as the system must wait for all selected participants to complete their local training and upload(30). This transmission delay can be longer than the local training time(11, 12, 14). This delay in transmission may be longer than the amount of time required for local training(20).

  Communication efficiency concerns are explored more in Section 3.3.

- **Client Selection:** In FL, where training is distributed across numerous devices like AVs, it's critical to decide which subset of vehicles to include in each training session(11, 12, 16, 18, 30). This is essential for efficiency, as including every client may not add value and increase communication costs(12). Also, some clients might not finish training, causing the entire round to fail(12). Client selection schemes aim to improve upon simple random selection(12). Various factors must be considered when choosing which AVs should participate in a given round(11, 12, 18):

- **Potential contribution:** This can relate to the quality or diversity of data held by the client(4, 5, 12, 18). Inefficiently trained models can be the result of selecting clients with inappropriate data subsets(12). It's possible that because of differences in data sizes or importance, local updates contribute in an unequal manner(5, 12).
- **Resource availability:** A variety of computational resources, including central processing units (CPUs), memory, and energy, as well as communication capabilities, are available to autonomous vehicles (AVs)(12, 18). The training procedure is both inefficient and wasteful due to limited computational resources(12). Selecting clients that have insufficient resources or unstable connections runs the risk of them not being able to complete the task or dropping out(12, 18). Another factor that leads to the "straggler effect" is the fact that the system waits for clients that are either delayed or have failed(12). Resource-aware selection takes into account a variety of factors, including time required, energy, memory, and CPU, that are necessary for tasks(9, 18).
- **Other factors:** Location and availability can also be relevant. Time needed for model distribution, training, and upload relative to a deadline is also a consideration(18). Finding the "best" client selection algorithm is not trivial, but it is an integral part of a well-functioning FL network(12). Methods exist that consider multiple criteria, such as resources and communication capabilities(18), or even dynamic factors like channel conditions and standing time(14, 15).

  These concepts are explored more in section 3.3.4.

- **Security and Privacy Risks:** FL creates new security challenges with model poisoning, free-riding on the prompt, the potential leakage of private information through shared model updates, and server vulnerabilities(11), even as they protect data privacy by not sharing raw data(3, 5, 11). Unreliable or faulty clients can also lead to undesired behaviors(16). This concept was explained more in the 3.5 section.
- **Aggregation Strategies:** It's critical to provide reliable aggregation algorithms that can efficiently merge model updates(model parameters or

gradients) from heterogeneous clients with differing data quantities and quality(15).

The aggregation strategy is extremely important since it is responsible for determining the manner in which these local updates are integrated to generate the next global model(4, 18).

- **Challenges**: Especially in situations where clients have different quantities and quality of data (non-IID data)(5, 12), one of the most significant challenges is efficiently merging updates from heterogeneous clients simultaneously(1, 15, 34). It is possible for simple averaging, such as FedAvg, to lead to problems such as client drift when the data is not identically distributed (non-IID)(24), or it may not appropriately account for clients whose data volumes and importance differ(5, 12).
- **Reliability**: The strategy must be reliable, capable of handling dynamic environments where connectivity fluctuates and clients might drop out(12, 20).
- **Approaches**: Various strategies exist beyond basic averaging. Some methods use weighting factors, for example, based on the number of samples a client provided, to ensure contributions are appropriately weighted(5). Other approaches include asynchronous or semi-synchronous aggregation to mitigate delays caused by slow clients or drop-outs(2, 4, 15, 29, 30). Selective aggregation mechanisms can be used to mitigate the effect of heterogeneity in data quality or computational ability(4, 15). Elastic parameter aggregation is mentioned in the context of balancing time and available resources in semi-synchronous frameworks(15).

- **Resource Management:** The implementation of FL on autonomous vehicles offers substantial challenges in terms of effectively managing the computational and communication resources across all of the participating vehicles(9, 11, 12, 16, 18, 30). Tasks that require a significant amount of resources include both the local training and the subsequent transmission of model updates(2, 11, 12, 16, 18, 30).

- **Computational Resources:** Automatic vehicles (AVs) possess variable levels of processing power and memory restrictions(12, 18). Because of the limited resources available on some devices, their local training may be ineffective, which may result in the creation of "stragglers" that slow down the entire FL round(12).
- **Communication Resources:** The costs associated with communication provide a significant challenge(12). Every single global model update necessitates receiving local aggregations from specific clients(12). This necessitates the consumption of bandwidth, which may be restricted in vehicular networks(11, 12). Low-energy devices have a big deal with the transmission of large models because of being energy-consuming(12).
- **Inefficient Utilization:** Challenges include ensuring that the resources of powerful edge devices are not idle, that network infrastructure is not underutilized, and that devices with weak connectivity are not neglected(11).

- **Scalability to a Large Number of Vehicles:** A critical prerequisite for using FL in real-world autonomous driving scenarios is the ability of the system to scale and handle a potentially large number of participating vehicles, maybe millions(12, 14, 28).
  - Challenges: Especially with a large client pool, traditional centralised FL methods could include sending parameters from all devices, resulting in significant network overhead, delay, and congestion(11, 28, 35). Managing several dynamic devices makes resource allocation, synchronization, aggregation, and coordination unstable(15). A direct average of parameters in aggregation could become infeasible with increasing participant numbers, especially if model structures differ(34). Major client involvement raises the processing requirements for global model updates(11).

Addressing these challenges requires innovative approaches, such as model comparison strategies, privacy-preserving techniques, and strong adversarial defenses(36).

3.1.8 Comparative Analysis of FL and Traditional ML in AVs

In the context of autonomous vehicles, FL has clear advantages over traditional centralized machine learning:

- **Data Privacy:** As explained in 3.1.2 and 3.1.6 sections, the main benefit of FL is its ability to train models without centralizing private driving data, which addresses privacy issues that come with traditional approaches that require uploading all data to a central server(1, 2, 11, 17). Traditional machine learning necessitates collecting and combining data on a central level(1, 18).
- **Decentralization:** FL avoids the requirement for the extensive infrastructure for collection and storage that centralized ML requires by utilizing the distributed nature of autonomous vehicles and their locally generated data(1, 2). In FL, training is conducted locally on individual devices(1).
- **Handling Data Heterogeneity:** Traditional machine learning models that were trained on aggregated data may not be able to adapt well to the variety of real-life scenarios that individual AVs may face. FL can provide a global model that is more robust to data heterogeneity by training locally on a variety of datasets(1, 2, 11). In AV settings, it is unlikely that training data has an independent and identical distribution (IID), which is a common assumption in traditional machine learning(4). This concept is explained more in prior sections (3.1.2 and 3.1.6) and is explained in depth in section 3.4.
- **Reduced Network Load:** Compared to traditional machine learning, which requires uploading massive amounts of data, FL has substantially less communication overhead because only model updates (which are generally much smaller than raw datasets) are sent(5, 11, 17). Some observations show that for a steering angle prediction task using a vision-based dataset in V2X-communicating vehicles, FL was found to be 250 times lighter under ideal channel conditions and 62 times lighter with channel errors than traditional centralized ML(5, 30). Another study found that an asynchronous FL approach for steering wheel angle prediction

reduced training time by approximately 70% and saved approximately 60% bandwidth compared to the traditional centralized learning process(5, 29). Beyond the fundamental benefit of sending updates, further techniques aim to minimize communication within FL(13). A federated distillation framework for regression tasks demonstrated a reduction in transmitted bits for communication load by approximately 98% compared to conventional FL by sending local average estimations per target segment instead of entire models(15). Additionally, exchanging model outputs instead of parameters has been proposed as a light-weight strategy that could achieve a latency reduction of up to 99% compared to existing FL algorithms for similar accuracy, especially with non-IID data distributions(9). During part 3.3.4, these concepts are explained more.

- **Improved Real-world Performance:** Compared to models trained on a static, centralized dataset, FL offers the potential to produce models that are more accurate and adaptable to real-world driving circumstances by continuously learning from the varied experiences of a fleet of vehicles in a privacy-preserving manner(1, 2). Fine-tuning a centralized model might make it work better with new data. However, the studies show that Federated Learning (FL) improves performance in the real world for AVs by letting them learn collaboratively from diverse and distributed experiences without the main problem that comes with traditional centralized methods.

However, it's crucial to remember that traditional machine learning may be superior in circumstances where data privacy is not the primary concern and where having access to a sizable, centralized, and carefully selected dataset makes training easier and, in specific scenarios, may result in faster convergence.

3.2 Privacy-Preserving Techniques for FL in AVs

3.2.1 Data Privacy Concerns in FL for AVs

Massive amounts of sensitive data, including photos, videos, and geolocation data, are collected and analyzed by autonomous vehicles (AVs)(1). It is crucial to protect private information from illegal access and possible misuse(1). Concerns regarding security vulnerabilities and privacy violations are raised by traditional centralized methods, which aggregate data and store it on a central server(1).
Despite the introduction of Federated Learning (FL), which attempts to protect privacy by keeping raw data locally, privacy issues still exist. It has been demonstrated through research that gradients may be applied to obtain identifiable client information(3). Nevertheless, sharing model updates with the central server of a third party during the training process may expose private data(3). A model trained on user data, for example, can be used to extract sensitive text patterns(3). Additionally, the model's memorization can result in privacy leaks(10).
An attacker may exploit local model updates obtained from IoT devices, including autonomous vehicles, inside a federated learning framework, and afterwards build their own global model and obtain private data from those devices(15). There is a difference between security and privacy, and understanding these differences is crucial. Security relates to protecting data against unauthorized access, whilst privacy refers to avoiding the unintentional disclosure of personal information(15). FL security challenges are elaborated in depth in section 3.5.

3.2.2 Differential Privacy for FL in AVs

Differential Privacy (DP) is widely used to handle these privacy issues in FL for AVs(13). By adding random noise to the data or model updates prior to transmission to the server, DP protects the privacy of the data(11). The goal of this procedure is to maintain data ownership while preventing unauthorised access to sensitive information(11). Because DP provides sample-level privacy,

the outcome is only slightly affected by the inclusion or exclusion of any single data point(13).

Adding various types of noise to the gradients or model updates (such as Gaussian, Laplacian, or Binomial noise) allows for the implementation of differential privacy(9, 13). To limit the impact of each data sample, it is typical practice to trim the gradients prior to applying the perturbation(13).

The inherent trade-off between accuracy and privacy settings is a key consideration when employing DP; while more noise provides more privacy, it can also compromise the accuracy of the model(11, 13). However, DP is still the most popular privacy metric in machine learning(13). To improve client models' privacy in the context of FL, DP can be used(10).

The application of DP in federated learning is analyzed in several research studies(3). For example, differentially private SGD has been suggested. Adaptive gradient clipping methods are also accessible to alleviate the challenges related to choosing suitable hyperparameters for DP(13). Data security has been integrated into FL for CAVs with the implementation of DP(11). The Differential Private Asynchronous Joint Learning (DPAFL) technique, which is suggested for vehicular network resource sharing, is one example. It incorporates local differential privacy to secure identity information and employs FL to protect message information(17). Additionally, the impact of adversary IoT devices (such as possibly malicious AVs) on the aggregated FL model can be reduced by using game-theoretic approaches that employ the use of differential privacy(15). Using game-theoretic approaches in Federated Learning (FL) means simulating how the different parts of the FL system, usually the server and the clients that are involved, interact and make decisions as if they were in a game(4, 9, 15). These methods are used to analyze and develop mechanisms that get certain results, usually by looking at the strategic behavior of the participants who are trying to get the most out of their productivity(9, 15).

Global privacy, in which model changes are private to all third parties save the central server, and local privacy, in which individual model updates are also private to the server, are often distinguished in the context of privacy in FL(13). Both kinds of privacy guarantees can be achieved by using differential privacy(13).

3.2.3 Secure Multi-Party Computation (SMPC)

For AVs, Secure Multi-Party Computation (SMPC) provides a further method for preserving privacy in FL. The data is encrypted and partitioned using cryptographic techniques by SMPC, allowing for collaborative computation on the data, with output results that the participating vehicles can access(11). Without disclosing the underlying model parameters, this technique makes it easier to train the model on data(11).

Despite introducing significant communication overheads, SMPC is efficient at preserving privacy and data security(11). The communication of model updates between the clients (AVs) and the server in an FL setup can be significantly encrypted via SMPC(11). Additionally, vehicles can utilize SMPC for collaborative intermediate calculations, ensuring that no useful information can be obtained from these calculations even in a scenario where the central server is compromised(11).

An SMC protocol was introduced by Bonawitz et al. to preserve individual model updates in federated learning. With this method, the central server can nevertheless observe the precise aggregated results at each round even though it has no access to any local updates. The resulting method has massive extra communication costs, even though SMPC is a lossless technique that can maintain the original correctness with an excellent privacy guarantee(13). SMPC can be used in combination with differential privacy(DP) techniques to provide even more robust privacy guarantees(13).

3.2.4 Homomorphic Encryption in FL

A powerful cryptographic method called Homomorphic Encryption (HE) enables a server to process encrypted data without requiring decryption(9, 11). This ensures data security and privacy in the context of FL for AVs by allowing the central server to aggregate model updates from vehicles, even if those updates are encrypted(11). Direct computation on encrypted data is possible with HE; the decrypted results are only accessible once the computation is complete(11).

Although HE's applications in machine learning have been restricted to specific settings, such as learning linear models or involving just a few entities, it shows great potential for privacy-preserving FL(13). In contrast to conventional machine learning-based algorithms, autonomous driving research, for example, reports better performance in terms of latency, accuracy, and security when using federated learning, blockchain, and homomorphic encryption for secure data transmission(17).

Model updates in FL can be securely aggregated using HE(15). For instance, a threshold Paillier cryptosystem can be used to aggregate participants' training models in a privacy-preserving Mobile Crowdsensing (MCS) system, which is similar to AV data collection(15). Similar to this, FedSky, an effective and privacy-preserving FL, ensures secure aggregate model updates through the use of an additive homomorphic encryption technique(15). Federated tensor mining has been proposed in the industrial IoT field, where factory data is encrypted using a homomorphic encryption mechanism and then shared with a central server for aggregation(9).

3.2.5 Federated Averaging with Secure Aggregation

For model aggregation in federated learning, Federated Averaging (FedAvg) is a widely adopted algorithm(8, 18). An improved global model is produced in this process by the central server, averaging the local model updates from a subset of participating clients (in this case, AVs)(1, 18).

Secure aggregation protocols are used in this aggregation process to further improve privacy. Instead of learning the individual updates from each device, these protocols make sure that the central server only learns the aggregated parameters from a group of devices(3), in a way that prevents it from seeing or learning about the individual model updates that each device in that group has provided(3, 11, 13). During the gathering process, cryptographic methods such as Secure Multi-Party Computation (SMPC) or Homomorphic Encryption (HE) are used to make this achievable(11, 13, 15). In this way, the server is able to obtain the final, useful aggregated result that is required for the global model

update, and the sensitive individual contributions are kept private and are not exposed to the server itself(3, 9, 11, 13, 15).

A number of practical secure aggregation techniques have been developed for machine learning that protect privacy(13, 15, 18, 33). In order to ensure that the server can only decode the total of the updates once a sufficient number of clients have contributed, these strategies frequently involve cryptographic techniques. With strategies like Turbo-aggregate seeking to overcome the quadratic aggregation barrier(3) and proposals for sparse secure aggregation to lower communication cost, recent research has also concentrated on increasing the efficiency of secure aggregation(3). Each participant and the server may possess a single partially private key in some safe aggregation algorithms, which prevents any participant from decrypting the updates of other participants(15).

3.2.6 Trade-offs Between Privacy, Security, and Performance

Trade-offs between the level of privacy and security reached and the resulting model performance and system efficiency are often necessary when implementing privacy-preserving strategies in federated learning for AVs(13). For instance, adding noise to improve privacy may result in a decrease in model accuracy even when Differential Privacy (DP) provides robust privacy assurances(13). In the same way, Secure Multi-Party Computation (SMPC) provides robust privacy and security, but it can result in significant computational and communicational costs, which could impact the overall efficacy of the FL process(11, 13).

To reconcile these competing objectives, the privacy-preserving technology and its parameters must be carefully chosen(13). For example, depending on the specific protocol being used, secure aggregation may result in communication overhead even though it can preserve the original accuracy of the aggregated model(13). The FL system designer must evaluate the system's practical vulnerability and carefully consider the trade-offs between prediction accuracy and resilience against different types of attacks(3).

To ensure efficient model performance and protect sensitive data during the FL training process for CAVs, it is essential to consider and use strong and secure

privacy preservation strategies(11). In order to promote participation and balance the resource usage related to privacy and security measures, incentive mechanisms may also be required(15). By actively participating in FL training rounds, clients are able to utilize valuable resources such as energy, computing power, and communication bandwidth(14, 15). Furthermore, clients may lack the intrinsic desire to dedicate their resources, especially if there is no direct benefit or reward(17, 34). In order to motivate enough client engagement(17) and inspire them to dedicate more resources or give high-quality data or updates for effective model training (11, 34), incentive mechanisms are needed. These mechanisms are often constructed using game theory(11-13, 15-17, 34, 37). Choosing suitable participants based on their computational capacity and the size of their local datasets can also make the system work better overall and improve its efficiency(15). When putting privacy-preserving techniques into action, it's vital to remember that data from AVs is not always the same (non-IID data), which can also affect how accurate FL-based models are(15). In the end, implementing Federated Learning for autonomous vehicles still presents a significant challenge: finding an appropriate balance between privacy, security, and performance.

3.3 Communication Efficiency in FL for AVs

During this section, the essential aspect of applying Federated Learning (FL) to Autonomous Vehicles (AVs) is examined, that is, communication efficiency. In FL, communication bottlenecks are significant challenges, particularly in the dynamic environment of vehicular networks.

3.3.1 Communication Bottlenecks in FL for AVs

Communication can be considerably slower than local computing, which makes it a crucial bottleneck in federated networks(13). This is especially true for autonomous vehicles (AVs) that operate in vehicular networks. In these situations, bandwidth is restricted, and communication channels might be

unreliable(16, 18). The large number of possible participating vehicles may make this problem worse(13).

In FL, a global model is broadcast to a selected number of clients (AVs) by the server during each training round. The clients then send back to the server their locally computed model updates for aggregation(12, 16). High communication costs, congestion, and substantial network traffic can all be caused by these interactions(16). Constraints on bandwidth, power, and energy on the client devices (AVs) cause the ongoing uplink and downlink transmissions during FL rounds to be slow(16). It affects the overall training time that can actually be counted in minutes, hours, or maybe days for the whole convergence process.(13) Using the FedAvg Algorithm's Performance Evaluation to finish training and evaluation in one instance, training large-scale automotive simulations with 1000 clients took roughly 3131.83 seconds (almost 52 minutes)(30).

One comparison revealed that while a suggested approach (ShuffleFL) took 3.50 hours to achieve target accuracy on a particular dataset, FedAvg took 16.45 hours(14, 30).

According to one source, a new FL method (FedSky) took about 0.2 hours (12 minutes) to complete one training round in image classification experiments(15).

FL performance is further impacted by mobility and communication resource issues for Connected Autonomous Vehicles (CAVs)(20). Frequent drop-outs and hand-overs may arise from the dynamic variations in CAV mobility and communication channels(20). These problems can lead to transmission delays that are significantly longer than the time it takes for devices to train their local models, which has a negative impact on the global model's convergence time(20). Another major obstacle is a lack of transmission bandwidth(20). For complicated perception tasks such as vision and LiDAR processing, which are widely used by autonomous vehicles (AVs), machine learning models (such as Deep Neural Networks) can still contain a significant number of parameters. This results in the model updates and gradients being substantial in size, even if they are smaller than the raw data(11, 15).

The "straggler effect," where the slowest vehicle limits the network speed, can result from the synchronous nature of some FL techniques, where a training

round finishes only once all chosen devices have transmitted their models(12). Communication costs are still the key issue, even with algorithms that choose clients more quickly(12).

3.3.2 Model Compression Techniques

Several strategies concentrate on minimizing the size of the messages sent during each communication round in order to alleviate the communication bottleneck(13). Sparsification, Subsampling, and Quantization are examples of model compression methods that can significantly minimize the size of these messages(13, 15).

- **Quantization:** It involves decreasing the model parameters' precision (for example, by representing weights with fewer bits)(15, 21).
- **Pruning:** It is the process of eliminating less important connections or neurons from the model in order to reduce the number of parameters(15). To reduce communication overhead, for instance, one study suggests combining IoT device selection with model pruning(15). Adaptively Federated Multi-Task Learning (AFL) makes sparse shareable structures that can be pruned over and over again to make model sharing more efficient(15).
- **Sparsification:** It entails transmitting only the update matrix's non-zero values(15, 18).
- **Lossy compression:** It can be applied to minimize communication between servers and devices(13).
- **Error Accumulation and Correction:** It is possible to ensure that the overall impact of updates is preserved throughout rounds by utilizing techniques such as error accumulation alongside compression(9, 18). This is the case even if individual updates are compressed. Control variates are used alongside local gradients by algorithms such as SCAFFOLD, which are designed to adjust the update direction in order to get it closer to the true global optimum(14, 33). These control variables are also updated, and it is possible to view them as contributing to guiding the corrections(33).

By constructing a sub-model with fewer parameters, Federated Dropout is a method to shrink the size of the global model(18). After that, this sub-model can be delivered to clients after being lossily compressed on the server side(18). Additionally, clients compress the updates they generate before returning them to the server(18). Other compression techniques that are mentioned include error accumulation and ternarization(18).

Sparsification, ternarization, error accumulation, and optimal Golomb encoding are all used in one communication-efficient protocol to compress both uplink and downlink communications(9). A different method, FetchSGD, uses a Count Sketch data structure to reduce gradients(9).

3.3.3 Asynchronous vs. Synchronous FL Approaches

FL can be carried out both synchronously and asynchronously.

- **Synchronous FL:** Before aggregating them to update the global model, synchronous FL waits for all (or a certain percentage of) selected clients to complete their local training and upload their model modifications(4, 12). This strategy is vulnerable to the straggler effect(12).
    - **Settings:** Synchronous methods are considered to be simple and promise a serial-equivalent computational paradigm(13). The most commonly used method in FL, the Federated Averaging (FedAvg) algorithm, follows a general operation that corresponds with a synchronous approach, including random selection of clients, aggregation of local updates, and generating a global model after every iteration round(30). This implies synchronous methods are often used as a baseline or default approach due to their simplicity and theoretical guarantees in ideal settings. They are discussed in the context of vehicular networks(4).
    - **Drawbacks and Suitability:** A major limitation of synchronous FL is its susceptibility to stragglers (slow clients) in the face of device variability, as the server must wait for the slowest client to complete its task, which can cause significant communication delays, especially when multiple vehicles upload simultaneously(4, 13). Some sources

suggest synchronous aggregation protocols are unsuitable for real-world heterogeneous hardware like that found in vehicles(29). The approach also often presumes that training data comes from an independent and identical distribution (IID), which is not always guaranteed in real-world vehicular networks(4).

- **Asynchronous FL:** It allows clients to upload model updates whenever they're ready, and the server updates the global model without waiting for all of the chosen clients to submit(4). This could mitigate the issue of stragglers. For resource allocation in vehicular communications, one study created both synchronous (SY) and asynchronous (ASY) algorithms(4).
    - **Settings:** Asynchronous systems are considered an attractive way to mitigate stragglers in heterogeneous settings(13). Particularly in automobile environments, they have been suggested and confirmed for real-time End-To-End FL(5, 29). This covers a particular asynchronous version-based aggregating technique shown for steering wheel angle prediction in self-driving vehicles(5). This method was found to outperform older synchronous protocols in prediction accuracy and significantly reduce bandwidth costs and training time by allowing clients to update their models more frequently(5).
    - **Suitability and Challenges:** Asynchronous methods are seen as more suitable for real-world heterogeneous hardware than synchronous protocols(29). They are also mentioned in the context of addressing the high mobility and uncertainty of CAVs in IoV networks through semi-synchronous frameworks(15). However, classical asynchronous schemes often rely on assumptions about bounded delays, which can be unrealistic in federated settings where delays might be very long or even unbounded(13). This suggests that novel models of asynchrony, such as those involving event-triggered device interactions, may be needed for typical federated networks where devices are often undedicated(13). Asynchronous aggregation is also explored as a potential scheme for hybrid FL scenarios(14).
- **Semi-SynFed:** A semi-synchronous FL framework that has been suggested as a solution to the high mobility and volatility of CAVs(15). To handle the high

mobility and uncertainty of CAVs in IoV networks(15), the Semi-SynFed framework combines elastic parameter aggregation with a dynamic waiting time scheme. This hybrid technique attempts to balance the strict waiting of synchronous techniques and the flexibility of asynchronous ones to manage device and network heterogeneity.

- **Purpose:** Balancing the time consumption and available computation resources depends much on the "elastic parameter aggregation"(15). This enables the aggregation process to adjust to variations in how fast different vehicles can train and upload their updates.
- **Mechanism:** In this system, the global model w_{t+1} at round t is updated using the formula: $w_{t+1} = w_t + \sum \alpha_k^t \nabla w_k^t$(15). Here, ∇w_k^t is the gradient of client k at round t(15). The phrase mixing hyperparameter is used to describe α_k^t(15). This parameter makes the aggregation "elastic" since it relates to the linear function of staleness and the multiplication of the proportion of training samples(15).
- **Meaning of "Elastic":** This mixing hyper-parameter α_k^t, not being a fixed value for all clients, leads to the "elastic" aspect. It dynamically adjusts the weight or influence of a client's update (∇w_k^t) on the global model based on variables including:
  - **Proportion of training samples:** Clients with larger datasets may contribute more, as is usual in weighted averaging (as in FedAvg)(15).
  - **Staleness:** The "staleness" probably relates to how old the update from client k is when the server combines it(15). Some clients may take longer to finish their local training and upload their models in heterogeneous or dynamic settings(29).

  An elastic aggregation can change the contribution depending on this staleness, so possibly assigning less weight to notably delayed updates or managing them so that they don't much impede the general training progress, which is a main concern in strictly synchronous FL because of stragglers(29).

3.3.4 Network Optimization for FL in AVs

In AVs, optimizing the network for FL requires several types of strategies:

- **Selection of Clients:** To achieve the maximum FL performance, including training duration and accuracy, as well as to improve communication efficiency, it is essential that suitable clients (AVs) be chosen for each round(12, 16). For choosing a client, variables such as each client's number of local data, computational capacity, and network resources can be taken into account(12). It was recommended that choosing clients as late as possible can increase efficiency(12). More clients may participate if incentive mechanisms are established(12, 15). According to one study, clients that have high-quality models have a higher chance of connecting to the base station (server)(18). Caching resource utilization can be optimized through mobility-aware vehicle selection based on standing time, local data, and channel conditions(15).
- **Resource Allocation:** Learning performance can be significantly improved by the effective use of communication resources(16). Due to high communication costs, techniques for joint client selection and resource allocation have been developed to shorten convergence times(12). The best subset of clients to choose for participation can be found via Reinforcement Learning (RL)(12). Communication costs can also be reduced by using distributed client selection algorithms in which clients take part in aggregation(12). Model aggregation through beamforming can be accelerated by techniques such as over-the-air computation (OAC or AirComp)(15).
- **Hierarchical Federated Learning (HFL):** To improve communication efficiency, HFL splits clients into clusters and aggregates updates on different levels (e.g., edge servers before a central server)(12, 14). Research is being done to determine the best thresholds for separating these clusters(12). A Full-stack FL (F2L) framework suggests Label-Driven Knowledge Distillation (LKD) to handle heterogeneity and employs a hierarchical network architecture for scalability(10).

- **Decentralized Federated Learning (DFL):** DFL might reduce communication overhead by enabling multiple vehicles to work together to train a model without the need for a central server(1, 11). Individual nodes selectively share important model updates with other nodes and train models locally(1).
- **Edge Computing:** By placing computation and storage closer to vehicles (edge devices), latency, energy, and bandwidth are decreased(9, 16).
- **Semantic Communication (SemCom):** By integrating FL with SemCom, IoT devices can send just relevant semantic features rather than complete data, which could reduce the volume of communication(15).

3.3.5 AV Communication Performance: FL vs. ML

Studies on AVs have shown that FL can improve communication efficiency when compared to typical centralized machine learning. One study testing steering angle prediction found that FL significantly reduced network load and maintained performance with less training time than centralized ML. This was especially true under ideal channel circumstances(5). FL frameworks save communication resources by just transmitting model data, which is less than the user data needed by centralized approaches(11).

It's crucial to remember that techniques like secure aggregation, which FL uses for privacy, may result in communication overhead(3). The amount of model updates and the number of communication rounds determine the overall communication cost in FL algorithms(7). By making the transmitted messages smaller, model compression approaches try to mitigate this(13).

3.4 Model Accuracy and Non-IID Data Challenges in FL for AVs

This section focuses on the key challenges that non-independent and non-identically distributed (non-IID) data creates on Federated Learning (FL) models, and the solutions addressing these issues, particularly in the Autonomous Vehicles (AVs) domain. It also discusses strategies and algorithms that have been developed for handling the above-mentioned problems.

3.4.1 Impact of Non-IID Data on FL Model Convergence

The heterogeneity of client datasets poses a significant problem in federated learning(7, 8, 12, 13). As mentioned in previous sections, in FL, training occurs on the local devices of clients, such as AVs, which inherently contain diverse datasets and differing availability(12). The variation in data distribution among clients is referred to as non-IID data, which can adversely affect the learning efficacy of FL models when clients are selected randomly(12).

Federated Learning contains several types of data heterogeneity, including feature distribution skew, label distribution skew, same labels with different features, identical features with different labels, and quantity skew(12). In the case of autonomous vehicles, this could lead to different driving conditions, traffic patterns, or sensors with different capabilities, resulting in changes in the data collected.

The existence of non-IID data may result in a divergence in local optima(22). As each client (AV) trains its local model on its own, sometimes biased data, the resulting local models may deviate considerably(23). The aggregation of these divergent local models at the server may slow down the convergence of the global model and diminish its accuracy(10, 12, 21-24). Label distribution skew, reflecting the varying types of objects encountered by distinct autonomous vehicles (e.g., one mainly sees cars while another primarily encounters pedestrians), has been identified as an important parameter contributing to performance degradation(12).

Furthermore, non-IID data may introduce biases from some customers into the training of the global model, leading to a reduction in accuracy(12). There is a critical necessity for client selection protocols that guarantee data impartiality in Federated Learning to resolve this issue(12). The weight gap between a federated learning model trained on non-IID data and a traditional centralized learning model trained on IID data mostly depends on the differences in data distribution(15).

Client drift happens when data is not IID, which makes local models deviate from the best global solution during training(24). This drift could make the model slower to converge and not learn well enough during model aggregation(24).

3.4.2 FL Algorithms for Handling Non-IID Data

A multitude of algorithms have been proposed to tackle the issues presented by non-IID data in Federated Learning. These methodologies often seek to either diminish the variance among client updates or alter the aggregation process on the server(21, 23).

- **Clustered Federated Learning (CFL)** has been proposed as an effective method for dealing with non-IID data by utilizing device heterogeneity to optimize client selection according to round latency and bandwidth(12). This method is effective in specific scenarios, particularly in IoT networks with inherently clustered devices(12). Another technique involves identifying clusters of clients with nearly independent and identically distributed (IID) data by becoming distribution-aware(12).
- Methods to mitigate **Label Distribution Skew** involve assessing the similarity between the aggregated data distribution of chosen clients and the global distribution(12) or assigning each client an irrelevance score to enhance data distribution uniformity(12). Grouping clients by data classes and then randomly selecting one client from each group is an appealing method(12). It is additionally suggested to incorporate diversity in client selection by assessing how a sample of clients can collectively reflect the entirety when aggregated(12).
- Algorithms such as **FedProx** (Federated Proximal) adjust the loss function on the client side by adding a penalty for the gap between the local model and the global model, to improve convergence with non-IID data(11, 13, 21).
- **SCAFFOLD** (Stochastic Controlled Averaging for Federated Learning) suggests adding a control variate into the model updates to align the client model's routing more closely with the optimal server route(10, 33).
- **FedNova** utilizes gradient scaling elements within the model update function in order to prevent the aggregated model from conforming to highly biased models(10).
- **Aggregation Delayed Federated Learning** incorporates redistribution rounds that postpone the aggregation of local models on the server by

transmitting local models to clients several times, demonstrating enhanced performance on non-IID data(21). This framework can additionally integrate importance sampling for client selection(21).

- **Adaptive Federated Multi-Task Learning (AFL)** generates sparse shared structures that may be iteratively pruned to tailor models for various tasks, potentially useful for heterogeneous data(15).
- By reducing client drift, **FedPcf** enhances communication efficiency and resilience on non-IID data by integrating multi-level prospective correction factors into the training process of the server and clients(24).
- **ShuffleFL** adds a user layer between devices and the server to allow data shuffling among devices belonging to the same user, aiming to match local data more closely and balance data distribution(14).

3.4.3 Transfer Learning and Knowledge Distillation for Non-IID FL

In the presence of non-IID data in FL, Transfer Learning and Knowledge Distillation are two strategies that can be utilized to increase the accuracy of the model.

- **Transfer Learning**, as was explained in section 3.1.5, can be accomplished by first training a global model on some shared proxy data and then fine-tuning it on individual clients(13). This allows for the personalization of models while utilizing a common base.
- The process of **knowledge Distillation** involves the transmission of knowledge from a teacher model (such as the global model or regional aggregated models in a hierarchical setup) to a student model (such as local client models)(10).
- **Label-Driven Knowledge Distillation (LKD)** is a strategy that has been developed that enables a student model to absorb meaningful knowledge from various teacher models. This technique also helps to reduce the divergence that exists amongst client models when non-IID data is present(10). It is possible to integrate LKD at the global server with hierarchical FL frameworks such as Full-stack FL (F2L). This allows the

frameworks to leverage the knowledge gained from regional aggregated models in order to handle non-IID challenges and enhance scalability(10).

3.4.4 Personalization and Client-Specific Model Adaptation

Personalization of FL models is essential because of the inherent heterogeneity of data, which is present in autonomous vehicles (AVs), as well as the probable requirement for models to perform optimally in certain driving situations or for individual vehicles(3, 13).

- The goal of personalized learning is to learn a **collection of client-specific models** that reduce test errors beyond what is possible with a single global model(3).
- Existing work on personalized learning frequently aims to **regularize the difference between local and global parameters**. This is in contrast to the goal of training a single global model(3).
- Methods such as **Multi-Task Learning** can be adapted to the federated environment in order to develop distinct but related models for each device while simultaneously utilizing a shared representation(13).
- Additionally, cyclic patterns in data samples can be addressed by a **pluralistic approach** that makes an adaptive choice between a global model and device-specific models. This solution can be implemented during federated training(13).

3.4.5 Empirical FL Performance Evaluation on AV Datasets

An empirical evaluation of the performance of FL models on real-world AV datasets is absolutely necessary in order to validate the efficacy of various algorithms and methods in the management of non-IID data. This evaluation must be carried out in order to ensure that the FL models are performing as expected.

- Findings of several studies reveal that FL has the potential to significantly improve the performance of tasks in autonomous vehicles (AVs), such as lane segmentation, in comparison to baseline models, which would result in improved stability and accuracy(5).

- FL has also been shown to be useful for learning spatio-temporal features for predicting directions in vehicular networks while still keeping the user's privacy protected(11).
- It is generally accepted that measuring accuracy is a better way to assess the performance of FL algorithms in comparison with communication rounds. This is because testing accuracy shows how well learning is successful, and it is also a better way to reduce the number of communication rounds because of their costs(12).
- When different FL approaches under IID and non-IID conditions are compared, it can be seen that more biased data makes the model converge more slowly. This emphasizes that the robustness of models is important while working on non-IID data(24).

To fully evaluate the performance of different FL approaches on a range of AV datasets and in real-world non-IID situations, further study is needed. Meanwhile, considering some important factors is crucial. These factors include changing sensor quality, the driving environment, and the behavior of participants. Moreover, to successfully use FL in self-driving cars, the trade-offs between making models more personalized, making communication more efficient, and protecting customers' privacy should be considered.

3.5 Security Challenges and Mitigation Strategies

This chapter focuses on the challenging security issues that Federated Learning (FL) systems face when used with autonomous vehicles (AVs). It also looks at the different ways that are beneficial to deal with these problems.

3.5.1 Threat Landscape in FL for AVs

In FL, the distributed structure is good for privacy, but when it comes to self-driving cars, it may cause the threat landscape to get more complicated (3, 9, 11).

- **Privacy Concerns:** Despite the inherent characteristics of FL to preserve users' privacy, research has demonstrated that sensitive information can still be discovered from gradients(3, 9, 13). Therefore, techniques that

preserve the client's privacy are required(3, 13). Regulations such as the General Data Protection Regulation (GDPR) apply to the data that is collected by various sensors in autonomous vehicles (AVs)(11).

- **Malicious Attacks:** Federated learning (FL) training can be vulnerable to malicious attacks, which involve compromised individuals transmitting false parameters in order to impair the performance of the global model(3, 11). At the same time, the central server itself is vulnerable to attacks(11).
- **Data Poisoning Attacks:** These involve adversaries injecting small amounts of poisoned data into the training phase of distributed learning models. This can result in inaccurate inference results(9, 15, 27). On the client side, this can take place through the modification of data features or the injection of inaccurate subsets of data in order to incorporate backdoors into the model(9).
- **Model Poisoning Attacks:** It is possible for attackers to modify local model updates before they are transmitted to the server. This type of attack is known as model poisoning(9).
- **Model Stealing, Membership Inference Attacks, and Model Inversion Attacks:** They are all examples of potential security vulnerabilities that can arise in distributed learning(15).
- **Evasion Attacks:** For the purpose of fooling a trained model, adversarial inputs might be purposefully produced(9, 15).
- **Free-riding:** It refers to the situation in which certain participants may contribute very little or nothing at all, while nevertheless experiencing the benefits of the global model(15).
- **Communication Security:** It is possible to intercept the transmission of model parameters between vehicles and the server(9, 11).
- **Compromised Clients:** It is possible for individual vehicles that are part of an FL system to be compromised and behave in a malicious manner(9, 11).

3.5.2 Adversarial Attacks on FL Models

AVs are vulnerable to adversarial attacks, which pose a substantial threat to FL models(3, 9, 11, 27).

- By poisoning training data, adversaries can attempt to disrupt the training process and cause disruptions(27).
- They are also able to make an effort to avoid being detected by trained models by using inputs that have been carefully crafted(27).
- Additionally, one of the objectives of adversarial attacks is to extract information about models(27).
- Generating "fake" data that is difficult to differentiate from real data can also be considered a tactic(27).
- Because of the limited resources available, individual AVs (also known as UAVs in a swarm) can be especially vulnerable to cyberattacks(27).
- It is essential for the development of attack detection techniques within the FL architecture in order to ensure that operations are both reliable and secure(9). Among these processes is the identification of abnormal updates through the verification of their low-dimensional embeddings(9).
- The identification and removal of malicious nodes is of vital importance for safe FL(9). In order to discover abnormal client behaviors, this may include the utilization of anomaly detection models that have been pre-trained(9). Detecting adversarial clients can also be achieved through the use of FL models themselves(9).

3.5.3 Byzantine-Resilient FL

In FL, there is a risk of Byzantine attacks, which occur when participants send updates that are arbitrary and could potentially be malicious(3, 4, 11).

- The goal of **Byzantine-resilient FL** is to guarantee that the global model converges correctly even when there are clients that are either faulty or malicious(11).
- A **Byzantine-Fault-Tolerance Decentralized Federated Learning (BDFL)** approach is intended to ensure Byzantine Fault Tolerance,

protecting against arbitrary or malicious client actions. This approach works within a Decentralized Federated Learning architecture. In decentralized systems, model updates can be included through aggregators that may operate on technologies such as blockchains. This particular BDFL methodology is designed for Autonomous Vehicles(11, 37).

3.5.4 Secure Aggregation Protocols

Within the context of the aggregation process that takes place on the server, secure aggregation protocols are absolutely necessary for the purpose of preserving the privacy and integrity of model updates(3, 9, 11, 13).

- The main idea is to allow the server to only learn aggregated parameters over groups of devices, rather than learning a single device(3).
- Computations can be performed on encrypted data without the need for decryption, thanks to techniques such as **Homomorphic Encryption(HE)**, which ensures the privacy and security of the data(9, 11).
- When it comes to protecting data throughout the training and inference stages, **Secure Multi-Party Computation (SMPC)** provides robust mechanisms(1, 13).
- **Differential privacy (DP),** as mentioned earlier in part 3.2.2, by adding noise into model updates or gradients, restricts the impact of malicious Internet of Things (IoT) devices and protects against privacy leaks(3, 9, 13). On the client side, this can be applied locally, and on the server side, it can be deployed globally(9, 13). On the other hand, this frequently comes at the cost of decreased system efficiency or model performance(13).
- By encrypting local updates and key sharing among users and the server for the purpose of fair data verification and key agreement during the aggregation process, a secure aggregation technique that makes use of a double-masking structure can protect clients(9).

3.5.5 Blockchain-Based FL for Enhanced Security

One of the beneficial technologies, which includes several features that improve the trustworthiness and security of FL systems for AVs, is Blockchain technology(9, 11, 17).

- **Decentralization:** The technology behind blockchain has the ability to take the place of the centralized aggregator that is utilized in traditional FL. Under these circumstances, the risk of a single point of failure would be eliminated, and the amount of trust in a central server would be decreased(9). It is possible to accomplish the development of serverless FL frameworks by utilizing blockchain technology(15).
- **Transparency and Immutability:** The blockchain can store model updates and other important information in immutable ledgers, which makes them transparent and tamper-resistant(9, 11). It helps check the model's parameters and ensure data integration(9).
- **Secure Model Exchange and Aggregation:** Blockchain technology makes it possible for participants in decentralized FL systems to communicate and share model updates in a secure manner(9). It is possible to offload model updates to the blockchain, which allows for secure exchange and aggregation of data(9).
- **Access Control and Authentication:** Blockchain technology has the potential to provide powerful authentication and access control during the training process. This can protect against malicious threats and data breaches(9). There are procedures for verifying transactions and calculating the averaged global model that can be implemented on consortium blockchains, which can be used to store permanently updated FL models(9).
- **Incentive Mechanisms:** Management and verification of incentive mechanisms for participants who contribute to FL can be accomplished through the use of blockchain technology(15).
- The possibility of combining blockchain technology and FL is investigated in several studies. The aim is to ensure the protection of users' privacy

while facilitating the sharing of data in a variety of IoT and vehicular network scenarios(9, 11, 15, 17).

To maintain the security of federated learning for AVs, it is essential to address many vulnerabilities. As explored before, some issues that should be considered are privacy leakage, adversarial assaults, and Byzantine clients. To mitigate these problems, some solutions or even a combination of solutions must be evaluated. The solutions that are mentioned before are secure aggregation protocols, leveraging the security features of blockchain technology, and employing privacy-preserving methods such as DP and HE. Moreover, further study is required to evaluate and enhance these methodologies, to make sure the implementation of federated learning in autonomous driving contexts is secure and reliable.(3, 11).

4 Critical Synthesis and Comparative Discussion

After in-depth investigations of the prior studies, it is time to combine and critically assess the concepts, frameworks, and methods found in Federated Learning (FL) for Autonomous Vehicles (AVs). This section attempts to provide a brief but detailed and critical look at these issues, key trade-offs, synergies, and current challenges in the integration of FL into vehicular networks. It especially focuses on privacy preservation, communication efficiency, system heterogeneity, and security.

4.1 Privacy-Preserving Techniques: Strengths and Trade-offs

All the reviewed literature demonstrated that the inherent characteristic of FL enhances data privacy. However, there is still the risk of inference attacks during model updates. All mentioned models, including Differential Privacy (DP), Secure Multi-Party Computation (SMPC), and Homomorphic Encryption (HE), are beneficial in protecting the client’s data. Moreover, all of them are good ways to protect sensitive data. As mentioned before, DP, widely employed for its statistical privacy guarantees, integrates controlled noise into gradients or model updates. On the other hand, the added noise could affect model accuracy,

exposing an intrinsic trade-off between privacy robustness and predictive efficiency.

SMPC ensures strong privacy through data partitioning and safe joint computations; nonetheless, it results in significant communication overhead and latency, which in turn pose difficulties in high-mobility vehicular networks. Searching on HE shows that it facilitates operations on encrypted data while preserving confidentiality; yet, it suffers from computational inefficiencies and scalability constraints in deep learning applications within edge-based systems.

A comparative advantage exists in the integration of various strategies. For example, combining DP with secure aggregation or incorporating lightweight HE techniques with DP can achieve a balance between performance and security, although practical applications are still rare in the automotive sector.

4.2 Communication Efficiency: Addressing Network Constraints

The communication overhead continues to be a significant obstacle to the effective implementation of FL in AV networks. The research underscores significant delays induced by synchronous aggregation techniques, especially in scenarios of network instability and client mobility.

Traditional synchronous FL, which is used in FedAvg, suffers from the “straggler effect”, as mentioned before, it causes slower clients to delay global model update.

In contrast, asynchronous FL and hierarchical FL architectures have demonstrated improved adaptability. Asynchronous FL permits clients to upload model updates independently, mitigating delays but raising concerns about model convergence and consistency. To enhance scalability and reduce bandwidth consumption, HFL controls the volume of data transferred to the central server. HFL leverages intermediate aggregation at roadside units (RSUs) to achieve this goal. Despite this solution, it should be mentioned that HFL involves coordination complexity and potential failure points at intermediary nodes.

Compression techniques, including Quantization, Pruning, and Sparsification, reduce communication burden. These techniques, however effective, must

mitigate any reductions in model accuracy, requiring compensatory measures such as error accumulation or adaptive compression methods.

4.3 Handling Non-IID Data and System Heterogeneity

Non-independent and identically distributed (non-IID) data continues to provide significant challenges in vehicular FL. The examined methods, such as FedProx and SCAFFOLD, try to alleviate client drift by adjusting local training or integrating control variates, respectively. Although successful in controlled trials, its practical viability in dynamic vehicular environments requires more verification.

The diversity of systems—differences in computational capacity, communication reliability, and power availability among autonomous vehicles—adds to the problem. The research illustrates the necessity of resource-aware client selection as an optimization, given that current approaches are inadequate for addressing highly dynamic, large-scale networks.

Federated Transfer Learning (FTL) serves as an effective method for enabling knowledge transfer from well-trained clients to resource-constrained nodes. Despite its effectiveness, it should be considered that its dependence on task similarity and availability of pre-trained models limits its generalizability across diverse AV applications.

4.4 Security and Ethical Considerations: Emerging Priorities

Recurrent threats in federated settings are security challenges such as adversarial attacks, model poisoning, and Byzantine behaviors. Although blockchain-backed FL strategies and secure aggregation techniques provide resilience, their incorporation into latency-sensitive AV networks is still mostly theoretical.

Especially, the examined literature provides little interaction with ethical aspects like model fairness, consent management, and regulatory compliance. Ethical governance systems have to follow innovations in federated learning as AV systems more and more affect urban mobility and public safety.

4.5 Open Challenges in FL for AVs

Regarding the implementation of FL for AVs, there are a number of unresolved difficulties that cover the technical, ethical, and practical aspects, including the following:

- **Technical Challenges:**
  - **Iterative Communication Overhead:** FL training requires a substantial amount of communication between vehicles and the server, which can be a bottleneck, particularly when considering the dynamic nature of vehicular environments(2, 5).
  - **Device Heterogeneity:** Autonomous vehicles (AVs) possess a wide range of computational capabilities and resources, which raise issues for efficient and equitable participation in FL(2). The heterogeneity of the system is one of the primary challenges in implementing FL(12).
  - **Data Heterogeneity (Non-IID Data):** There is a possibility that the data gathered by various vehicles have non-identical and non-independent distributions (non-IID), which might have an impact on the convergence and accuracy of the global model. Therefore, it is important to consider the possibility of data heterogeneity (16). One of the important challenges is to find the best way to keep the accuracy of the model without any changes when dealing with non-IID data distributions.
  - **Scalability:** The management of a large number of participating vehicles in FL can lead to an increase in solution time and memory utilization, which in turn requires a greater amount of processing resources for global model updates(11).
  - **Real-time Learning:** There are still lots of obstacles to real-time learning. The utmost importance of it is to ensure that FL is capable of supporting the real-time decision-making requirements of autonomous driving(1, 2).
  - **Communication Efficiency:** It is of greatest importance in FL networks due to the restricted communication capacity and the periodic availability of clients. This makes it necessary to pick clients efficiently

and share updates. It should also not be forgotten that there is a need to address the issue of communication latency(11, 16).

- **Resource Utilisation:** It is of utmost importance, as it involves effective management and distribution of computing and communication resources among the vehicles that are participating(11).
- **Model Heterogeneity:** Due to the fact that there are differences between the architectures and capacities of models available for various vehicles, new challenges have emerged(1, 11).
- **Maintaining Model Synchronisation:** The process of maintaining model synchronization can be challenging because it is difficult to keep local models across various vehicles aligned with the global model(11).

- **Ethical Challenges:**
  - **Fairness:** It is an ethical responsibility to make certain that the global model functions in a manner that is fair across various user groups and driving situations that are represented by the vehicles that are participating(1, 11). The mathematical measurement of long-term fairness through limitations has been achieved(12).
  - **Digital Ethics Issues:** Those that arise from the huge volumes of data involved are of significant concern since they present substantial privacy and security issues that need to be addressed without affecting the accuracy of the model(11).
- **Practical/Deployment Challenges:**
  - **User Adoption:** It can be challenging to encourage involvement, engagement, and adoption from vehicle owners in FL systems, which is a challenge with regard to user adoption(2).
  - **Standardised Frameworks:** In the absence of standardized frameworks that are specifically designed to allow FL's integration into the automotive industry, widespread adoption is affected(2).
  - **Interference in Wireless Mobile Networks:** In wireless mobile networks, interference can have an effect on the quality of communication during the process of uploading local results. This

interference can potentially result in the loss of data and a reduction in the reliability of the global model(4).

- **Data Quality:** It is of utmost importance, since it is essential to guarantee the accuracy and consistency of the data obtained from various vehicle sensors(4).
- **Capability Diagnostics:** It is necessary to conduct capability diagnostics in order to evaluate the appropriateness and capabilities of various vehicles for the purpose of participating in FL(11).
- **Vehicle Selection and Resource Allocation:** When it comes to dynamic vehicular environments, the difficulty of efficiently selecting appropriate clients and allocating resources remains unresolved(11, 16).

4.6 Emerging Trends in FL for Edge AI in AVs

The future of FL for edge AI in autonomous vehicles is highlighted by several emerging trends, including the following:

- **Cooperative Perception:** FL enables vehicles to increase their perception capabilities through collaborative efforts by fusing data from numerous sources while preserving the vehicles' privacy(1).
- **Real-time Federated Learning:** It is essential for real-time applications in autonomous vehicles to adapt FL in order to have the capability of achieving faster convergence and being responsive to dynamic driving situations(1). Techniques such as asynchronous FL might be utilized in this context(15).
- **Integration with Edge Computing Infrastructure:** Leveraging the constantly increasing processing capacity of edge computing nodes to support FL processes in vehicular networks can enhance efficiency and reduce latency(1, 11). This can be accomplished by integrating with edge computing infrastructure. The convergence of AI and edge computing for unmanned aerial vehicles (UAVs) is also being investigated, which has implications for autonomous vehicles (AVs)(27).

- **Sixth Generation (6G) and Vehicle-to-Everything (V2X) Technologies:** The introduction of 6G and the development of Vehicle-to-Everything (V2X) communication technologies have the potential to dramatically boost FL in AVs by offering higher bandwidth and lower latency(1, 11).
- **Clustered Federated Learning:** Investigating clustered FL designs, in which vehicles create groups for the purpose of local aggregation before connecting with the central server, can help reduce the amount of communication overhead while improving scalability(11).
- **Federated Transfer Learning (FTL):** The utilization of FTL to use knowledge from related activities or domains has the potential to enhance the performance of FL models in AVs, particularly when there is a limited amount of local data(9).
- **AI-native UV Networks:** The idea of AI-native unmanned vehicle (UV) networks, in which AI is extensively incorporated into network functionalities, including learning, has the potential to inspire similar developments in AV networks that leverage FL(27).

4.7 Comparative Summary

Table 1 in the literature provides a valuable comparative overview of FL architectures, highlighting the trade-offs between Centralized, Decentralized, Hierarchical, and Deep Federated Learning models. Centralized FL offers simplicity but risks single-point failures; Decentralized and Hierarchical models improve resilience and scalability at the cost of coordination complexity. While increasing model capacity, Deep FL imposes significant computational and communication loads. Although encouraging for latency-sensitive applications, End-to-End FL needs more fine-tuning in real-time, asynchronous environments. All things considered, although notable developments in privacy-preserving federated learning for AVs have been achieved, issues with communication efficiency, data heterogeneity, system heterogeneity, and security remain. Dealing with them requires integrative solutions combining several privacy-preserving mechanisms, asynchronous communication systems, adaptive aggregation techniques, and ethically focused system designs.

5 Discussions and Conclusions

According to what is investigated in previous chapters, this chapter presents the findings of the thesis in relation to the predefined research questions, underlines the practical consequences for using Federated Learning (FL) in Autonomous Vehicles (AVs), identifies key limitations and ethical issues, and suggests future research directions.

5.1 Answers to Research Questions

As a result of the investigation, some essential aspects of applying FL to AVs have been brought forward:

- **Research Question 1: Which privacy-preserving techniques are most effective for FL in AV systems?**
  The most efficient privacy-preserving strategies for federated learning (FL) in autonomous vehicle (AV) systems, as this paper shows, are Differential Privacy (DP), Secure Multi-Party Computation (SMPC), Homomorphic Encryption (HE), and secure aggregation protocols. Each approach has its own benefits and trade-offs. For instance, DP often sacrifices model accuracy but offers high statistical privacy by introducing calibrated noise to gradients. By comparison, SMPC and HE allow encrypted model updates or computations, hence improving confidentiality but at higher computational and communication costs. Particularly, HE struggles with scalability in real-time vehicular networks.
  Especially recent developments like Turbo-Aggregate and sparse secure aggregation, secure aggregation methods are quite appropriate for AV settings. By guaranteeing that only aggregated updates are observable to central servers, they provide the optimal balance between privacy and efficiency. A key benefit in bandwidth-limited vehicular networks is that these protocols also reduce communication overhead (as covered in Section 4.1). Therefore, combining secure aggregation with lightweight DP

methods seems to be the most feasible way to preserve privacy in FL-based AV systems.

- **Research Question 2: How does FL impact communication efficiency compared to traditional ML models?**

  By sending model updates rather than raw data, communication overhead is reduced significantly through using FL, hence lowering network load by up to 250 times in ideal conditions and around 62 times in the presence of channel faults when compared to centralized learning frameworks. FL, on the other hand, creates new communication problems. Although easier, synchronous update systems are affected by the straggler effect, in which slow clients postpone global model updates.

  Asynchronous FL and hierarchical FL (HFL) systems have been developed to solve this. HFL, especially with the integration of Roadside Units (RSUs) for local aggregation, improves scalability and reduces latency. Furthermore, model compression methods—such as Quantization, Pruning, and Sparsification—are efficient in further lowering communication loads with little influence on model performance. Operationalizing FL in high-mobility, resource-constrained vehicle networks requires these approaches, investigated thoroughly in Section 4.2.

- **Research Question 3: What are the key challenges and solutions for maintaining model accuracy in FL for AVs with non-IID data distributions?**

  AV networks naturally contain non-independent and identically distributed (non-IID) data because of different geographic, environmental, and sensory characteristics. Model convergence and generalization suffer from this diversity. Effective solutions include algorithmic interventions such as FedProx and SCAFFOLD, which reduce client drift by adding proximal terms or control variates.

  Federated Transfer Learning (FTL) also improves convergence by using shared representations from related activities; smart client selection policies, depending on resource availability, data quality, and earlier contributions, help to guarantee consistent learning even more. These

answers, however, face operational limits in very dynamic networks with fluctuating connectivity.

Current techniques show theoretical potential, as described in Section 4.3, but they must be modified for scalability and real-time responsiveness in vehicular settings.

- **Research Question 4: What are the technical and ethical challenges in implementing FL on AVs?**

  FL deployment on AVs is technically challenging due to system heterogeneity, intermittent connectivity, high latency, limited onboard resources, and, in addition, vulnerability to adversarial threats such as data poisoning and model inversion attacks. These issues need robust aggregation methods, dynamic model compression, and adaptive scheduling frameworks.

  Ethically, the discourse is still in its early stages. Current FL literature fails to address issues such as fairness in model personalization, informed permission for data usage, transparency of model behavior, and accountability in public road situations. As highlighted in Section 4.4, the lack of comprehensive ethical governance frameworks exposes a fundamental gap.

  Thus, technical and ethical factors must be addressed simultaneously, especially when FL is used in systems that directly affect public safety and social trust.

- **Research Question 5: What are the major research gaps and future directions in FL for AV applications?**

  The studies show several unresolved issues. There aren't many empirical studies that prove FL works under large-scale, real-time AV situations. There hasn't been much research on how to use advanced privacy-preserving approaches (such as combined DP-HE frameworks) in real-world AV systems.

  There are still lots of interesting subjects to investigate, such as Federated Reinforcement Learning (FRL), customized FL algorithms for individual vehicle behavior, and blockchain-based FL systems for secure audit trails. But there are not enough sources to delve into them. Another topic that is

important and should not be neglected is the development of simulation tools such as FLEXE for assessing FL models under realistic mobility patterns.

At last, the ethical governance of FL systems—especially in the public road setting— requires a dedicated research agenda. It is a good idea to investigate and focus on papers that include values like fairness, explainability, and social accountability in the design of FL for future research.

5.2 Practical Implications for AV and FL Integration

The research in this thesis has implications for using Federated Learning (FL) in communities for Autonomous vehicles (AVs). The finding showed that using communication-efficient designs, like Hierarchical Federated Learning (HFL), can make vehicle networks much less limited by latency and bandwidth. This decrease is especially clear when limited aggregation is made easier by Roadside Units (RSUs). This shows that they are moving away from centralized models and toward scalable, decentralized systems that can handle frequent learning updates in changing network settings.

Second, privacy-protecting tools, especially safe aggregation protocols, need to be made to work with the mobility and computing limits of AVs. These must be combined with adaptive Differential Privacy methods that change the amount of privacy noise based on how sensitive the data is and how risky it is in that area. This means that privacy settings that take into account the context need to be built into the firmware or operating system of self-driving cars.

In the end, Federated Transfer Learning (FTL) customization procedures can be quite helpful for building global models that work well for all types of cars in all kinds of driving conditions. This helps drivers make better decisions in a wider range of traffic situations, which makes autonomous vehicle fleets safer with more ability to find their way around. This means that people in the business world and people who put together systems need to come up with flexible, interoperable federated learning frameworks that can work with a variety of hardware platforms and still follow new rules about AI and privacy.

5.3 Limitations and Ethical Considerations

This thesis, being a review-based study, is limited by the availability and variability of published results. Empirical performance metrics, especially in large-scale real-time AV networks, remain sparse (5, 18). Furthermore, while security concerns are well-documented, ethical challenges such as model fairness, data consent, and the social implications of privacy-preserving AV systems receive insufficient attention in current literature (3, 19). Future work should explicitly integrate ethical AI governance principles into FL design for AV applications.

5.4 Future Work and Research Agenda

Although there are still many areas to explore regarding the use of FL in AVs, some of the crucial areas for future studies may include the following:

- **Advancing Communication-Efficient FL and Intelligent Client Selection in Vehicular Networks**
  working on developing more communication-efficient FL algorithms and client selection techniques that have been tailored for the dynamic and resource-constrained nature of vehicular networks.
- **Addressing Statistical Heterogeneity in AV Data for Improved Global Model Performance**
  Investigating more advanced methods to efficiently manage the inherent statistical heterogeneity (non-IID data) in autonomous vehicle (AV) data in order to guarantee that global models reach high accuracy and fairness across a wide range of driving scenarios.
- **Integrating FL with Emerging Technologies for Next-Generation Autonomous Mobility**
  Exploring the possibility of integrating FL with other cutting-edge technologies, such as edge computing, 6G, and advanced V2X communication, in order to develop autonomous systems that are more effective and responsive.
- **Developing Realistic Simulation Platforms and Evaluation Benchmarks for FL in AVs**

The creation of realistic simulation environments and benchmarks for the purpose of conducting an extensive assessment of the performance and resilience of various FL techniques in a variety of AV use scenarios.

5.5 Final Thoughts

Federated Learning is a revolutionary approach that creates the possibility of enabling collaborative AI for autonomous vehicles while preserving users' privacy. This thesis addresses persistent technical and ethical gaps that need to be solved in order to ensure that autonomous vehicle (AV) systems are safe, fair, and efficient. Substantial progress has been achieved in addressing challenges related to privacy, communication efficiency, and model accuracy. For the purpose of developing FL toward scalable, real-world vehicular applications that are in line with both technical and societal standards, it is essential to continue this type of interdisciplinary research.

References

1. Wang W. Empowering safe and secure autonomy: federated learning in the era of autonomous driving. Applied and Computational Engineering. 2024;51:40-4.
2. Ma T. Federated Learning-based neural networks for autonomous driving. Applied and Computational Engineering. 2024;49:192-8.
3. Ding J, Tramel E, Sahu AK, Wu S, Avestimehr S, Zhang T, editors. Federated learning challenges and opportunities: An outlook. ICASSP 2022-2022 IEEE international conference on acoustics, speech, and signal processing (ICASSP); 2022: IEEE.
4. Tan K, Bremner D, Le Kernec J, Imran M, editors. Federated machine learning in vehicular networks: A summary of recent applications. 2020 international conference on UK-China emerging technologies (UCET); 2020: IEEE.
5. Eid Kishawy MM, Abd El-Hafez MT, Yousri R, Darweesh MS. Federated learning system on autonomous vehicles for lane segmentation. Scientific Reports. 2024;14(1):25029.
6. Kairouz P, McMahan HB, Avent B, Bellet A, Bennis M, Bhagoji AN, et al. Advances and open problems in federated learning. Foundations and trends® in machine learning. 2021;14(1–2):1-210.
7. Reddi S, Charles Z, Zaheer M, Garrett Z, Rush K, Konečný J, et al. Adaptive federated optimization. arXiv preprint arXiv:200300295. 2020.
8. Paul S, Sengupta P, Mishra S. FLaPS: Federated Learning and Privately Scaling. 2020 IEEE 17th International Conference on Mobile Ad Hoc and Sensor Systems (MASS). 2020:13-9.
9. Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Poor HV. Federated learning for internet of things: A comprehensive survey. IEEE Communications Surveys & Tutorials. 2021;23(3):1622-58.
10. Nguyen M-D, Pham VQ, Hoang DT, Tran-Thanh L, Nguyen DN, Hwang WJ, editors. Label driven Knowledge Distillation for Federated Learning with non-IID Data2022.
11. Chellapandi VP, Yuan L, Brinton CG, Żak SH, Wang Z. Federated learning for connected and automated vehicles: A survey of existing approaches and challenges. IEEE Transactions on Intelligent Vehicles. 2023;9(1):119-37.
12. Smestad C, Li J, editors. A systematic literature review on client selection in federated learning. Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering; 2023.
13. Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: Challenges, methods, and future directions. IEEE signal processing magazine. 2020;37(3):50-60.
14. Zhu R, Yang M, Wang Q. ShuffleFL: Addressing Heterogeneity in Multi-Device Federated Learning. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies. 2024;8:1 - 34.
15. Le M, Huynh-The T, Do-Duy T, Vu T-H, Hwang W-J, Pham Q-V. Applications of distributed machine learning for the internet-of-things: A comprehensive survey. IEEE Communications Surveys & Tutorials. 2024.

16. Hayyolalam V, Özkasap Ö. Communication Efficient Client Selection in Federated Learning.
17. Mishra A, Jefferson OPJ, Kapur P, Kannur K, Agarwal P, Arya A. Swarm Learning In Autonomous Driving: A Privacy Preserving Approach. Proceedings of the 2023 15th International Conference on Computer Modeling and Simulation. 2023.
18. AbdulRahman S, Tout H, Mourad A, Talhi C. FedMCCS: Multicriteria client selection model for optimal IoT federated learning. IEEE Internet of Things Journal. 2020;8(6):4723-35.
19. Qi P, Chiaro D, Guzzo A, Ianni M, Fortino G, Piccialli F. Model aggregation techniques in federated learning: A comprehensive survey. Future Generation Computer Systems. 2024;150:272-93.
20. Lobato W, Costa J, Souza AMd, Rosário Dd, Sommer C, Villas LA. FLEXE: Investigating Federated Learning in Connected Autonomous Vehicle Simulations. 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall). 2022:1-5.
21. Xue Y, Klabjan D, Luo Y, editors. Aggregation delayed federated learning. 2022 IEEE International Conference on Big Data (Big Data); 2022: IEEE.
22. Son HM, Kim MH, Chung T-M. Comparisons where it matters: Using layer-wise regularization to improve federated learning on heterogeneous data. Applied Sciences. 2022;12(19):9943.
23. Nguyen H, Phan L, Warrier H, Gupta Y. Federated learning for non-iid data via client variance reduction and adaptive server update. arXiv preprint arXiv:220708391. 2022.
24. Zang Y, Xue Z, Ou S, Long Y, Zhou H, Du J, editors. Fedpcf: An integrated federated learning framework with multi-level prospective correction factor. Proceedings of the 2023 ACM International Conference on Multimedia Retrieval; 2023.
25. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA, editors. Communication-efficient learning of deep networks from decentralized data. Artificial intelligence and statistics; 2017: PMLR.
26. Lyu L, Yu H, Yang Q. Threats to federated learning: A survey. arXiv preprint arXiv:200302133. 2020.
27. Liu G, Huynh NV, Du H, Hoang DT, Niyato DT, Zhu K, et al. Generative AI for Unmanned Vehicle Swarms: Challenges, Applications and Opportunities. ArXiv. 2024;abs/2402.18062.
28. Nguyen A, Do T, Tran M, Nguyen BX, Duong C, Phan T, et al., editors. Deep federated learning for autonomous driving. 2022 IEEE Intelligent Vehicles Symposium (IV); 2022: IEEE.
29. Zhang H, Bosch J, Olsson HH, editors. Real-time end-to-end federated learning: An automotive case study. 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC); 2021: IEEE.
30. Murthy SL. Towards production-ready end-to-end Federated Learning for automotive applications: University of Stuttgart; 2022.
31. Zhang T, He C, Ma T, Ma M, Avestimehr S. Federated Learning for Internet of Things: A Federated Learning Framework for On-device Anomaly Data Detection.(June 2021). arXiv preprint arXiv:210607976. 2021.

32. Shalev-Shwartz S, Shammah S, Shashua A. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:161003295. 2016.
33. Karimireddy SP, Kale S, Mohri M, Reddi SJ, Stich SU, Suresh AT, editors. SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. International Conference on Machine Learning; 2019.
34. Shojaee P, Zeng Y, Wahed M, Seth A, Jin R, Lourentzou I. Task-Driven Privacy-Preserving Data-Sharing Framework for the Industrial Internet. 2022 IEEE International Conference on Big Data (Big Data). 2022:1505-14.
35. Liu S, Yang S, Zhang H, Wu W. A federated learning and deep reinforcement learning-based method with two types of agents for computation offload. Sensors. 2023;23(4):2243.
36. Fallah A, Mokhtari A, Ozdaglar A. Personalized federated learning: A meta-learning approach. arXiv preprint arXiv:200207948. 2020.
37. Lee C, Heiss J, Tai S, Hong JW-K, editors. End-to-end verifiable decentralized federated learning. 2024 IEEE International Conference on Blockchain and Cryptocurrency (ICBC); 2024: IEEE.