

Capítulo

2

Aprendizado Federado Veicular: da Teoria à Prática

Lucas Airam Castro de Souza (UFRJ e INRIA), Guilherme Araujo Thomaz (UFRJ), Mateus da Silva Gilbert (UFRJ), Vinicius de Oliveira Avena (UFRJ), Felipe Gomes Táparo (UFRJ), João Victor Dias Sobrinho (UFRJ), Fernando Dias de Mello Silva (UFRJ), Nadjib Achir (INRIA), Miguel Elias Mitre Campista (UFRJ), Luís Henrique Maciel Kosmowski Costa (UFRJ)

Resumo

Veículos conectados e automatizados (Connected and Autonomous Vehicles - CAVs) aprimoram a experiência de condução e o conforto dos passageiros por meio de diversas aplicações de segurança e entretenimento, muitas delas baseadas em aprendizado de máquina. Diante da necessidade de utilizar dados pessoais para o aperfeiçoamento dessas tecnologias, o aprendizado federado surge como uma alternativa para treinar modelos de aprendizado de máquina com garantia de privacidade. Entretanto, devido ao ambiente dinâmico com alta mobilidade, o aprendizado federado veicular (Vehicular Federated Learning – VFL) apresenta desafios além dos inerentes ao FL tradicional. Este minicurso contextualiza a execução básica e a relevância do VFL, introduzindo sua arquitetura, detalhando estratégias para superar os desafios de latência e mobilidade enfrentados no cenário veicular. Além disso, o minicurso discute a infraestrutura de rede necessária para o VFL, analisando o plano de dados e o plano de controle na transmissão de modelos e a orquestração de tarefas. O conteúdo também abrange as tendências propostas para a superação dos desafios práticos, incluindo a heterogeneidade de dispositivos dos clientes, as limitações de largura de banda do canal sem fio e a distribuição estatística de dados entre os veículos. Ainda, o presente minicurso pretende aprofundar-se e elevar a experiência dos participantes de forma prática, demonstrando como implementar e testar soluções. Por fim, o minicurso discute pontos em aberto e futuras direções de pesquisa.

2.1. Introdução

Veículos conectados e automatizados (Connected and Autonomous Vehicles - CAVs) transformam a mobilidade por meio da integração de sensores e atuadores, com-

binando com a inteligência artificial. Os veículos autônomos são divididos em 6 níveis de acordo com as suas capacidades, conforme exibido na Figura 2.1. No nível zero, o carro não possui nenhuma automação e necessita de intervenção constante do condutor. Já no nível L1, o condutor recebe informações de assistência na direção, porém este continua responsável pelo controle do veículo. Por outro lado, veículos parcialmente automatizados realizam o controle da direção e da aceleração sem intervenção humana para evitar acidentes. No nível 3, o veículo realiza mais de forma independente, como a coleta de informações sobre o ambiente, sendo o controle do veículo assistido pelo condutor, que deve permanecer atento a eventuais falhas ao longo de toda a viagem, pois o controle pode ser revertido ao usuário. Veículos com classificação L4 estão um nível abaixo dos veículos completamente automatizados, necessitando apenas da observação do usuário no controle do veículo. Por fim, os veículos L5 dispensam qualquer intervenção humana para o seu funcionamento, podendo ser supervisionados remotamente.

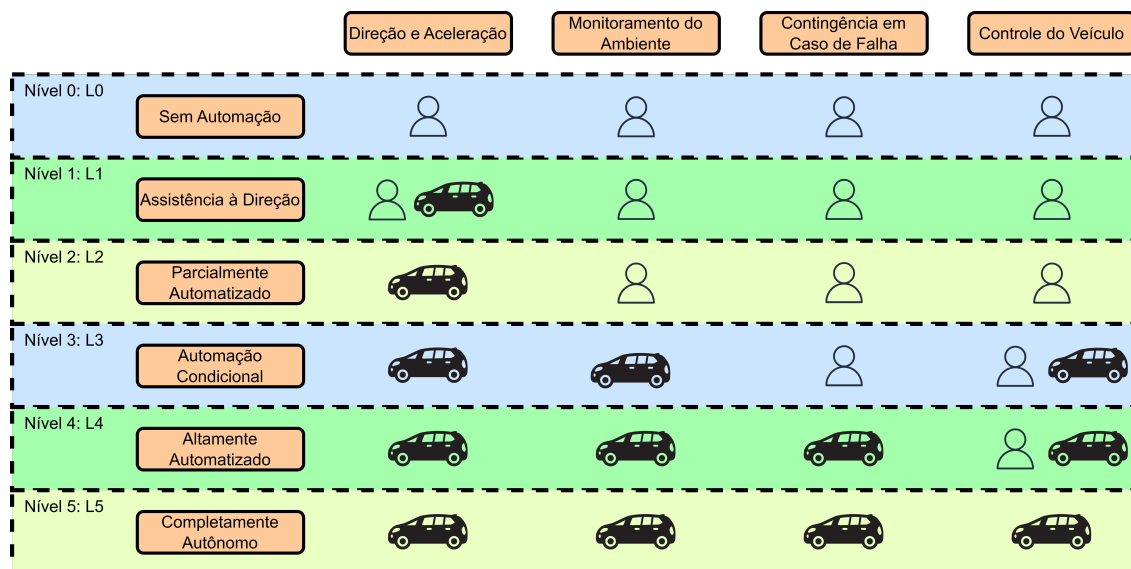


Figura 2.1. Classificação de veículos autônomos de acordo com as tarefas executadas. O ícone de usuário indica que a atividade necessita de intervenção humana, o veículo com o usuário indica que a tarefa é parcialmente automatizada e necessita de supervisão, enquanto o veículo indica que a tarefa é completamente automatizada.

O pilar dessas inovações reside em modelos de inteligência artificial, que exigem um treinamento com dados para oferecer alto desempenho em funções críticas, como a previsão de situações de risco futuras e o controle automatizado de parâmetros internos do automóvel. Diante da necessidade de utilizar dados reais e pessoais do usuário para o aperfeiçoamento dessas tecnologias, surge o desafio de proteger a segurança do condutor, o que torna o uso do aprendizado federado (*Federated Learning - FL*) [McMahan et al., 2017] uma solução indispensável. Essa abordagem permite que o treinamento dos modelos de inteligência artificial ocorra de forma eficiente, evitando a transferência direta de dados sensíveis dos usuários para centros de processamento externos, garantindo assim a privacidade das informações coletadas pelo veículo. Além disso, o compartilhamento de modelos, em vez de dados, reduz a quantidade de informação

transmitida, considerando que os clientes estão constantemente coletando novas amostras de alta dimensionalidade.

Tradicionalmente, o FL utiliza o FedAvg [McMahan et al., 2017], um algoritmo de treinamento federado que atualiza o modelo global com a média ponderada dos pesos do modelo de diferentes clientes, ponderada pela quantidade de dados de cada cliente. Cada rodada de atualização é realizada de forma síncrona, na qual o servidor aguarda a resposta de todos os clientes selecionados por um determinado tempo limite para atualizar o modelo global com as respostas recebidas. A fim de evitar sobrecarga computacional para os clientes, os autores propõem selecionar uma porcentagem para atualizar o modelo em cada rodada. No entanto, o aprendizado federado apresenta desafios, como a dificuldade de convergência devido à heterogeneidade dos dados dos clientes [McMahan et al., 2017, Li et al., 2020]. Como o modelo global é treinado indiretamente, quando os clientes têm distribuições de dados distintas, as atualizações do modelo tendem a ser discordantes dependendo de quais clientes participam do treinamento. Portanto, a seleção do subconjunto de clientes que participa de uma rodada de treinamento pode impactar a qualidade do modelo treinado [Kairouz et al., 2021].

Em um cenário veicular onde os clientes estão em constante movimento, esse desafio é agravado pela heterogeneidade do canal dos clientes [Bao et al., 2021, Buyukates e Ulukus, 2021, de Souza et al., 2025], que pode gerar longos períodos de desconexão ou falhas de comunicação. Além disso, os dispositivos em um ambiente veicular são heterogêneos [ElectronicsMaker, 2024] e a suposição de que o treinamento deve ser realizado em um modelo comum para o cálculo da média torna-se irrealista. Ainda, desafios práticos de aplicações veiculares, como a falta de rótulos para o treinamento de modelos de detecção de objetos, tornam o cenário mais desafiador [Li et al., 2022a].

O objetivo principal do minicurso é contextualizar a execução básica e a relevância do aprendizado federado veicular (*Veicular Federated Learning – VFL*), apresentando como o arcabouço do VFL permite o treinamento de modelos de inteligência artificial, garantindo a privacidade dos dados dos usuários. Dessa forma, introduz-se a arquitetura do VFL, detalhando estratégias de agregação, como as abordagens centralizadas, hierárquicas e descentralizadas, para superar os desafios de latência e mobilidade. Além disso, abordam-se os esquemas de coordenação semissíncronos [Wu et al., 2021a] e assíncronos [Xie et al., 2019], que buscam adequar o FL a cenários dinâmicos, relaxando a necessidade de sincronismo e evitando problemas de atraso.

Outro objetivo é discutir a infraestrutura de rede necessária para o VFL, utilizando uma abordagem de divisão em planos de dados e controle para otimizar a transmissão de modelos e a orquestração de tarefas. Para isso, comparam-se tecnologias de transmissão física como 5G & *Beyond*, enlaces laterais e satélites, além de protocolos de aplicação (MQTT, HTTP), transporte (TCP, UDP) e rede (IP, ICN). O curso também identifica e oferece caminhos para a superação de desafios práticos, incluindo a heterogeneidade de hardware dos clientes, as limitações de largura de banda do canal sem fio, a distribuição estatística de dados entre os veículos e a escassez de dados rotulados.

Para consolidar o conhecimento teórico, o minicurso capacita os participantes por meio de uma atividade prática de simulação realista de um ambiente veicular. A simulação inclui a definição de heterogeneidade de dados, dispositivos, mobilidade e o canal de

comunicação. Ainda, os dados gerados são integrados ao arcabouço Flower para execução do aprendizado federado veicular de forma realista, avaliando o impacto de diferentes parâmetros no desempenho do modelo.

Finalmente, o curso pretende motivar o desenvolvimento de pesquisas sobre tendências futuras, como a personalização de modelos e a destilação de conhecimento para arquiteturas heterogêneas, visando a implantação eficiente de soluções inteligentes em ambientes de alta mobilidade.

2.2. Aprendizado Federado e seu uso em Redes Veiculares

Tradicionalmente, o Aprendizado de Máquina (*Machine Learning – ML*) requer que um conjunto de dados representativo seja reunido em um único local, o que representa um desafio para redes veiculares, dado o volume de dados produzido por cada veículo e a sensibilidade inerente a esses dados [Chellapandi et al., 2023, Zhang et al., 2023]. Por exemplo, foi estimado que Veículos Conectados e Automatizados (*Connected and Autonomous Vehicles – CAV*) produzem entre 20 e 40 TB de dados por dia [Chellapandi et al., 2023], contendo informações que permitem a caracterização dos veículos e dos condutores. Alternativamente, o FL é um paradigma de Aprendizado Descentralizado de Máquina (*Decentralized Machine Learning – DML*) em que um conjunto de clientes, por exemplo, dispositivos móveis, executa o aprendizado colaborativo de um modelo sem compartilhar seus dados [McMahan et al., 2017]. Assim, cada cliente possui uma cópia local do modelo de ML a ser ajustado, sobre o qual aplica modificações com base em seus dados privados. Periodicamente, os clientes compartilham as atualizações feitas aos modelos locais para que sejam agregadas pelo servidor. Por sua vez, o servidor combina as atualizações dos clientes em um modelo global, que, posteriormente, é difundido aos clientes da federação, para que o processo de aprendizado seja repetido. O ciclo de uma atualização do modelo global é denominado rodada, enquanto as atualizações locais dos clientes são denominadas épocas locais. Sob essa estrutura de aprendizado, o FL permite o treinamento de modelos, mitigando problemas que surgem em um aprendizado exclusivamente local, como a baixa representatividade do conjunto de dados disponível em relação à distribuição real do fenômeno a ser aprendido [McMahan et al., 2017, Yuan et al., 2024].

Embora o FL aborde o problema da privacidade de dados e a redução dos custos de comunicação associados ao ML em redes veiculares [Posner et al., 2021], sua configuração tradicional implica hipóteses e restrições que podem dificultar seu emprego efetivo nessas redes. Um dos principais desafios à implementação de FL em redes veiculares é a alta mobilidade dos clientes. A federação pode sofrer modificações decorrentes da inclusão e da remoção de clientes, uma vez que veículos podem entrar e sair da área de cobertura do servidor de agregação [Fittipaldi et al., 2025]. Além disso, as informações enviadas pelos membros da federação podem chegar com atrasos distintos devido às condições variáveis do canal de comunicação e às distâncias entre clientes e agregadores [Zhang et al., 2023]. Por fim, diferenças no ambiente em que cada cliente federado se encontra, bem como na qualidade dos equipamentos de coleta empregados, provocam heterogeneidade nos dados coletados [Chellapandi et al., 2023]. Por consequência, as estratégias de agregação de informação e de coordenação dos clientes veiculares devem considerar as características de cada rede veicular, a fim de garantir o sucesso do processo

de aprendizado. No que segue, descreveremos sistematicamente as principais estratégias de agregação e de coordenação de clientes federados.

2.2.1. Estratégias de agregação

No FL clássico, as informações aprendidas pelos clientes, como os parâmetros ou gradientes do modelo, são enviadas a um único nó-servidor, responsável por agregá-las, difundir os parâmetros do modelo global e orquestrar a federação [McMahan et al., 2017]. Devido à centralização da etapa de agregação, essa variação é conhecida como Aprendizado Federado Centralizado (*Centralized Federated Learning – CFL*) [Beltrán et al., 2023], e o nó-agregador é referido como servidor de agregação. Essa centralização traz alguns problemas que a tornam inadequada a alguns cenários encontrados em redes sem fio [Beltrán et al., 2023, Yuan et al., 2024]. O primeiro deles é a existência de um único ponto de falha, que pode interromper o processo de aprendizado quando o servidor fica incomunicável ou torna-se inoperante [Li et al., 2021b, Yuan et al., 2024]. Adicionalmente, a largura de banda do servidor pode ser um gargalo quando múltiplos clientes enviam atualizações simultaneamente, retardando a operação de agregação, e as distâncias entre clientes e servidor podem aumentar os custos de comunicação [Posner et al., 2021, Chen et al., 2023a]. Além disso, centralizar a operação de agregação requer a confiança dos clientes no servidor da federação [Kalra et al., 2023], uma vez que servidores maliciosos podem extrair informações sensíveis dos dados recebidos [Huang et al., 2021].

O Aprendizado Federado Descentralizado (*Decentralized Federated Learning – DFL*), também conhecido como FL Distribuído ou sem servidor (*Serverless FL*) [Beltrán et al., 2023], é uma variação em que a etapa de agregação ocorre de maneira distribuída nos próprios clientes [Beltrán et al., 2023, Yuan et al., 2024]. No DFL, os clientes federados compartilham suas atualizações diretamente com seus parceiros, por meio de comunicação ponto a ponto (*Peer-to-Peer – P2P*), difusão (*broadcast*) ou fofoca (*gossiping*) [Yuan et al., 2024]. Isso traz duas vantagens em relação ao CFL. Uma comunicação mais direta entre nós permite reduzir os custos associados à transmissão de dados e evita que muitos clientes se comuniquem simultaneamente com um único nó [Chen et al., 2023a, Yuan et al., 2024]. Por fim, o DFL permite uma maior personalização de modelos do que o CFL, pois os próprios clientes federados podem decidir com quem interagir durante o processo de aprendizado e como agregar as atualizações recebidas [Jeong e Kountouris, 2023, Liu et al., 2024b].

Se, por um lado, o DFL permite a remoção de um único ponto de falha e maior liberdade para definir como os clientes interagem entre si, essa variação pode apresentar desvantagens em relação ao CFL quanto ao tempo de convergência e à qualidade dos modelos aprendidos. A topologia da federação, que diz respeito a com quem cada cliente federado se comunica durante o processo de aprendizado, desempenha um papel fundamental na taxa de convergência da federação e na capacidade de generalização do modelo aprendido [Shi et al., 2023, Sun et al., 2025]. Isso ocorre porque os dados de veículos próximos seguem uma distribuição semelhante, que pode divergir da distribuição real. Logo, o modelo resultante de sua interação pode sobreajustar-se a essa tendência. Além disso, agregar modelos muito distintos reduz o desempenho e aumenta o número de iterações até a convergência do modelo [Shi et al., 2023]. Topologias densas, nas quais clientes que

participam de DFL estabelecem conexões com um grande número de membros da federação, tendem a apresentar modelos que generalizam melhor. Entretanto, essa configuração pode acarretar altos custos de comunicação [Sun et al., 2025], particularmente em redes de grande escala.

Considerando as vantagens e desvantagens de CFL e DFL, o Aprendizado Federado Híbrido ou Semi-DFL (SDFL) apresenta-se como um conjunto de estratégias de agregação que combina características de ambas as abordagens anteriores [Beltrán et al., 2023]. Dentre essas estratégias, uma das principais é o Aprendizado Federado Hierárquico (*Hierarchical Federated Learning – HFL*) [Abad et al., 2020, Liu et al., 2020]. Nessa variação, constrói-se uma hierarquia de nós agregadores, na qual clientes que formam a base da hierarquia se organizam em pequenos aglomerados (*clusters*) e geram um modelo do nível hierárquico. Os modelos são agregados em múltiplas etapas, seguindo a estrutura hierárquica [De Rango et al., 2021, Azimi-Abarghouyi e Fischione, 2025].

Assim, o HFL intercala momentos de agregação intra-aglomerado (*intra-cluster*), semelhante ao CFL clássico, com momentos de agregação inter-aglomerado para estabelecer um modelo global [Abad et al., 2020]. Em redes veiculares, estações radio-base ou Unidades de Borda de Estrada (*Roadside Units – RSU*) podem ser exploradas como servidores de agregação locais próximos aos veículos, cujos modelos são enviados a um servidor em nuvem, para gerar o modelo global [Xie et al., 2026]. Outra variação, mais próxima do DFL, consiste em eleger periodicamente um dos clientes como servidor de agregação [Beltrán et al., 2023]. Essa abordagem permite reduzir a necessidade de infraestrutura dedicada à agregação na borda, além de mitigar a sobrecarga persistente em um único nó agregador.

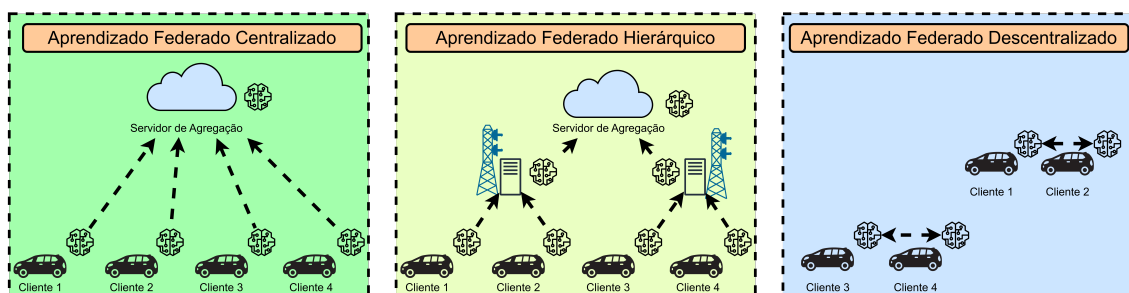


Figura 2.2. Comparação entre as diferentes estratégias de agregação.

De modo geral, pode-se considerar o CFL e o DFL como dois extremos de estratégias de agregação, e as demais variações combinam e privilegiam características de ambos, conforme os requisitos operacionais de cada implementação de FL, como mostrado na Figura 2.2. Outro fator importante para o sucesso do FL é definir quando a agregação deve ocorrer, o que é abordado pelas diferentes estratégias de coordenação.

2.2.2. Estratégias de coordenação

Enquanto as estratégias de agregação definem o local de agregação dos modelos, as estratégias de coordenação definem o instante em que os modelos serão agregados. Como no caso anterior, em que temos um conjunto de estratégias cujos extremos são CFL

e DFL, pode-se classificar as estratégias de coordenação em síncronas, semissíncronas e assíncronas. O FL clássico emprega uma estratégia de coordenação em que a operação de agregação ocorre somente quando o servidor recebe as atualizações de todos os clientes, após esses executarem o mesmo número de épocas locais até um tempo limite (*timeout*) [McMahan et al., 2017]. Isso implica que, a cada operação de agregação, todos os clientes devem estar sincronizados na mesma rodada e que o servidor aguarde que todos encerrem seus processos de aprendizado locais e transmitam suas atualizações. A exceção são os clientes que eventualmente estejam desconectados ou em um estado de falha.

Essas premissas podem ser problemáticas quando a federação é composta por dispositivos heterogêneos e a transmissão de atualizações ocorre por meio de redes sem fio. As diferenças de recursos entre dispositivos podem provocar variações no tempo de execução dessas operações locais, e clientes com uma conexão ruim com o servidor de agregação impactam negativamente o tempo de início da operação de agregação [Wang et al., 2022]. Em um FL síncrono, como descrito, o servidor e os dispositivos que executam o aprendizado local rapidamente devem aguardar os dispositivos retardatários (*stragglers*). Isso impacta o tempo de convergência da federação e torna o aprendizado ineficiente devido a ociosidade dos clientes com mais recursos computacionais [Xie et al., 2019, Xu et al., 2023].

Para mitigar o efeito de clientes retardatários, pode-se relaxar o requisito que obriga o servidor a esperar todas as atualizações dos clientes da federação, permitindo que esse agregue modelos treinados em rodadas diferentes, após receber um número fixo de atualizações, ou ainda sempre que recebe uma nova atualização. O último desses é conhecido como FL assíncrono [Xie et al., 2019], enquanto os dois primeiros são geralmente classificados como estratégias de FL semissíncrono [Stripelis et al., 2022, Nguyen et al., 2022].

No FL assíncrono, procura-se minimizar ao máximo os estados ociosos, gerando um novo modelo global a cada atualização, eliminando completamente o atraso decorrente da espera aos clientes retardatários. Entretanto, o emprego efetivo do FL assíncrono requer atenção quanto ao algoritmo de agregação. Uma vez que clientes em uma mesma federação utilizam versões distintas do modelo global, há um impacto negativo tanto a taxa de convergência quanto na qualidade do modelo aprendido. Em particular, clientes retardatários possuem modelos obsoletos que degradam o modelo global se agregados com o mesmo peso que os demais [Liu et al., 2024a].

Dessa forma, o FL semissíncrono propõe uma solução intermediária em que a operação de agregação ocorre de forma semelhante ao FL síncrono, porém elimina a restrição de aguardar que todos os clientes enviem suas atualizações [Stripelis et al., 2022]. Para mitigar a obsolescência dos modelos locais, o FL semissíncrono define momentos de sincronização entre um número mínimo de clientes. Porém, clientes retardatários não têm seus modelos descartados, como ocorre no caso síncrono. Caso o servidor receba um modelo de um cliente que chegue após a agregação, deve-se determinar o atraso em relação ao modelo global. O objetivo é decidir se o modelo defasado pode contribuir com o modelo ou se deve ser descartado para preservar o desempenho do modelo global. O grau de defasagem tolerável e o peso na agregação dos modelos defasados são hiperparâmetros

do FL semissíncrono. Entretanto, o FL semissíncrono também está sujeito a problemas semelhantes aos do FL assíncrono, como o problema de convergência [Islam et al., 2025].

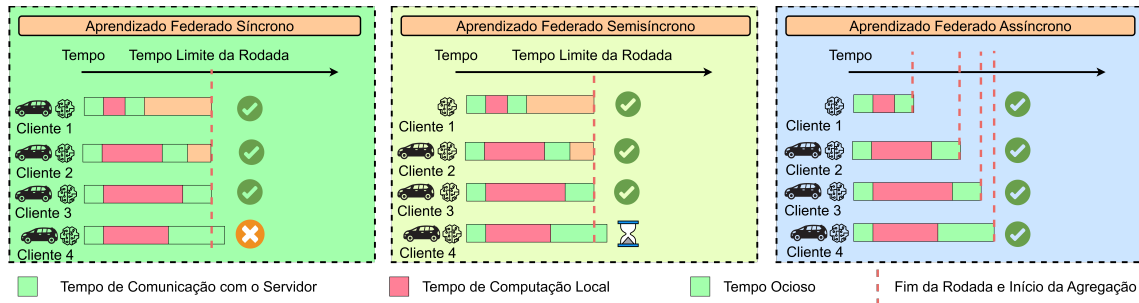


Figura 2.3. Comparação entre as diferentes estratégias de coordenação.

A Figura 2.3 apresenta as diferenças entre as principais estratégias de coordenação. Cada estratégia de coordenação possui vantagens e desvantagens, relacionadas às diferentes tecnologias de comunicação dos clientes para a realização do treinamento. Assim, a seguir, abordam-se as principais características da infraestrutura, detalhando os principais protocolos utilizados nas camadas inferiores da arquitetura.

2.3. Infraestrutura da Rede do Aprendizado Federado Veicular

A infraestrutura da rede do FL no cenário veicular é formada por componentes de *hardware* e *software*, cuja implantação é dividida em três setores: borda veicular, núcleo e a computação em nuvem. A borda veicular inclui os dispositivos nos quais os dados são gerados e os dispositivos com os quais eles se comunicam diretamente por um enlace. O núcleo inclui toda a infraestrutura tradicional de rede utilizada para interconectar as redes de acesso na borda aos servidores em nuvem. A computação em nuvem inclui as redes de *data centers* e os servidores de alta capacidade computacional, responsáveis pelo gerenciamento do FL e pela agregação de modelos.

O projeto da infraestrutura de rede para FL veicular em todos estes setores deve levar em consideração três frentes principais, discutidas ao longo das seguintes subseções. A Subseção 2.3.1 discute a frente de tecnologias físicas de transmissão utilizadas para implementar os enlaces da borda veicular. Em seguida, a Subseção 2.3.2 apresenta a frente de protocolos de comunicação utilizados ao longo de toda a rede, que têm por objetivo a garantia de desempenho e resiliência a falhas distribuídas. Por fim, a Seção 2.3.2 foca na frente de gerenciamento de sistemas em rede com o objetivo de suportar a inicialização, a finalização, o monitoramento e a atualização das tarefas de FL.

A Tabela 2.1 elenca as principais tecnologias utilizadas em diversas camadas de rede para suporte ao FL, incluindo protocolos, padrões, arquiteturas e algoritmos. As tecnologias em verde já são adotadas em VFL, e serão discutidas ao longo das Seções 2.3.1 e 2.3.2. As tecnologias em violeta são propostas para resolver problemas emergentes em comunicações para VFL, sendo discutidas em mais detalhes na seção de desafios desse minicurso, Seção 2.4.3.

Tabela 2.1. Comparativo de protocolos e tecnologias por camada, onde a cor verde denota tecnologias já adotadas em VFL, enquanto que a cor violeta denota propostas para solucionar desafios.

Camada	Protocolos/Tecnologia	Pontos fortes	Limitações
Aplicação	HTTP+REST	Não mantém estado de conexão	Sobrecarga com URI
	CoAP	Adequado para IoT	Fragmentação e segurança
	gRPC	Resiliente a redes desafiadoras	Dependência de conexão persistente
	MQTT/AMQP	Assincronismo	Maior latência devido ao broker
Transporte	TCP	Confiabilidade e controle de congestionamento	Sobrecarga de controle e conexão
	UDP	Evita sobrecarga de controle e conexão	Perdas e congestionamento
	PCON	Controle ativo de fila de roteamento	Exige modificações na camada de rede
Rede	IP	Compatível com a infraestrutura atual da Internet	Melhor esforço (poucas garantias)
	ICN NDN	Cache, multicast, mobilidade e ECN	Camada de adaptação ao IP traz sobrecarga
Acesso ao Meio (MAC) e transmissão física	Redes celulares (4G, 5G)	Alcance, vazão e latência	Custo com plano de serviço das operadoras
	IEEE 802.X	Equipamentos de baixo custo e consumo	Baixo alcance e interferência
	Redes space-air-ground	Cobertura em certas regiões	Complexidade da infraestrutura
	Novas técnicas de DSP (NOMA)	Maior eficiência espectral	Equipamentos novos e mais caros
	RIS	Aumento do alcance com baixa energia	Necessidade de instalação física

REST: *Representational State Transfer*

URI: *Uniform Resource Identifier*

CoAP: *Constrained Application Protocol*

gRPC: *gRPC Remote Procedure Calls*

AMQP: *Advanced Message Queuing Protocol*

PCON: *Practical Congestion Control*

ICN: *Information-Centric Networking*

NDN: *Named Data Networking*

ECN: *Explicit Congestion Notification*

DSP: *Digital Signal Processing*

NOMA: *Non-Orthogonal Multiple Access*

RIS: *Reconfigurable Intelligent Surface*

2.3.1. Tecnologias físicas de transmissão e acesso

Existem muitas propostas que usam FL para otimização de sistemas de comunicação sem fio veiculares. Entretanto, a discussão nesta subseção foca em tecnologias físicas de transmissão utilizadas para suportar o FL veicular. O objetivo é criar enlaces entre os *hardwares* presentes na borda veicular, ou seja, os dispositivos nos quais os dados são gerados e outros dispositivos em borda.

Na borda veicular, os dispositivos que geram dados estão presentes nos veículos, nos *smartphones* dos pedestres e passageiros e em equipamentos IoT. A rede local dos veículos inclui desde dispositivos simples, tais como sensores, até dispositivos com maior poder computacional, como computadores de bordo e Unidades Eletrônicas de Controle (*Electronic Control Units – ECUs*). No contexto de transmissão de dados em redes veiculares, o foco é em uma dessas ECUs, chamada de ECU primária, ou *gateway*, que provê conectividade sem fio com o restante da infraestrutura [Kornaros et al., 2020]. Existem também dispositivos dedicados ao provisionamento de redes de acesso, permitindo o en-

caminhamento de mensagens para outros dispositivos nas proximidades ou para a rede de núcleo, visando alcançar servidores em nuvem. Em termos de processamento, alguns dispositivos que geram dados podem ser incapazes de treinar modelos localmente, delegando essa tarefa para unidades de borda devidamente equipadas, como as RSUs (*Road Side Units*) [Magdum et al., 2021] localizadas na proximidade.

A transmissão dos dados pode ser direta entre veículos (*Vehicle-to-Vehicle – V2V*), entre o veículo e os outros dispositivos na infraestrutura de borda (*Vehicle-to-Infrastructure – V2I*) ou entre o veículo e o restante da rede (*Vehicle-to-Network – V2N*). Todas estas abordagens fazem parte do guarda-chuva de comunicações V2X (*Vehicle-to-Everything*). A abordagem tradicional do FL veicular depende de comunicação V2N, uma vez que o servidor de agregação está em nuvem. Algumas propostas evidenciam ganhos de latência ao posicionar o servidor de agregação no ponto de acesso e utilizar comunicação V2I, se servindo do padrão em MEC (*Multi-access Edge Computing*). Por fim, as abordagens de FL descentralizado dependem da troca direta de modelos entre os veículos por meio de V2V [Zejun et al., 2024].

A comunicação sem fio entre estes dispositivos tipicamente utiliza padrões comuns em redes celulares, como o 4G (*Long-Term Evolution – LTE*) e o 5G (*New Radio – NR*). Esses padrões, definidos pela 3GPP (*3rd Generation Partnership Project*) usam tecnologias de acesso por rádio com alta vazão e baixa latência, necessárias para a transmissão de modelos grandes no FL veicular. O acesso ao meio compartilhado e a modulação utilizam o OFDMA (*Orthogonal Frequency Division Multiple Access*), no qual cada usuário recebe um bloco de recurso (*Resource Block – RB*), que consiste em uma faixa de frequência e *slot* de tempo no qual ocorre a transmissão do sinal modulado com os dados. Alguns *slots* são dedicados para *uplink* (equipamento do usuário para estação base) e outros são dedicados para *downlink* (estação base para equipamento do usuário). A implantação das redes de acesso pro rádio já está bastante madura, garantindo uma cobertura com ampla penetração no território urbano [Magdum et al., 2021].

As redes de quarta e quinta geração incluem padrões para comunicação direta entre equipamentos do usuário (*User Equipments – UE*). Entretanto, a maioria das implantações é configurada para comunicação direta com as estações rádio base, desmotivando o uso de enlaces laterais (*side-links*). Ainda assim, a inclusão de padrões mais específicos de V2X pode reduzir a latência de sistemas de FL que exigem comunicação entre componentes em borda. Outras tecnologias promissoras na camada física incluem o padrão IEEE 802.11p para V2V, permitindo também comunicações diretas de curto alcance (*Dedicated Short-Range Communications – DSRC*) [Magdum et al., 2021].

2.3.2. Protocolos de rede e sistemas distribuídos

Na maioria das propostas de FL veicular, a troca de parâmetros entre servidor e clientes ocorre através de mensagens que trafegam pela Internet. Isso se deve ao fato de que a rede de acesso inclui apenas os dispositivos na borda veicular, enquanto que o servidor de agregação é uma máquina em nuvem. Nesses cenários, além das tecnologias físicas de transmissão sem fio, outra frente que deve ser considerada no projeto da infraestrutura de rede do FL veicular é a escolha dos protocolos. Para estudar as possibilidades de protocolos para FL veicular, será considerada a pilha de protocolos da arquitetura em

camadas da Internet (TCP/IP), a menos que dito o contrário. O foco será nas camadas superiores (aplicação e transporte) A camada de rede utiliza o IP para encaminhar os pacotes entre redes de forma interoperável, segundo o modelo de ampulheta (*cintura fina*) da Internet. As camadas de enlace e física são discutidas em outras seções.

A camada de aplicação é a mais flexível a nível de projeto, uma vez que o desenvolvedor tem a liberdade de construir suas próprias máquinas de estados finitas (*Finite-State Machine* – FSM) no nível de *software* de aplicação. A FSM define quais mensagens serão enviadas, quando serão enviadas e o que será feito quando elas forem recebidas, tanto no cliente quanto no servidor. Em geral, essa troca de mensagens nas aplicações envolve o *download* de parâmetros no início de uma rodada, o *upload* de parâmetros no final da rodada e mensagens de controle utilizadas pelos clientes para ingressar em uma federação (no início) e sincronizar o seu estado com o servidor (periodicamente). Com um projeto adequado de protocolo de aplicação, é possível resolver problemas típicos de sistemas distribuídos, como tolerância a falhas e garantia de sincronismo.

Outro aspecto importante no desenvolvimento da aplicação é o formato das mensagens trocadas entre nós no FL. Heydarishirayeh implementa uma ferramenta de FL que permite a escolha de diferentes protocolos de aplicação tradicionais para troca de mensagens [Heydarishirayeh, 2023]. Os experimentos utilizaram quatro opções (REST+HTTP, CoAP, gRPC+HTTP2 e MQTT/AMQP), a serem resumidas a seguir. Os autores introduziram condições de rede desafiadoras, simulando cenários com muitas perdas, baixa vazão e alta latência. Esse cenário é comum nos enlaces sem fio da rede de acesso em certos locais com sombras e interferência.

O paradigma REST (*Representational State Transfer*) permite a criação de APIs *web* sem estado (*stateless*) utilizando mensagens HTTP, sendo usada na ferramenta de FL FLoBC [Ghanem et al., 2022]. Entretanto, essa abordagem introduz uma sobrecarga significativa em termos de *bytes* de controle devido ao fato de que todas as mensagens incluem um cabeçalho com URI (*Uniform Resource Identifier*). O CoAP (*Constrained Application Protocol*) é uma alternativa baseada em REST voltada para dispositivos IoT leves, mas que apresenta alta latência em condições de rede desafiadoras. A necessidade de fragmentação em pacotes menores que 1 kB, a falta de recursos de segurança e a possibilidade de aumento do congestionamento pelo uso de UDP desestimulam o uso desse protocolo em aplicações reais. O gRPC (*Google Remote Procedure Call*) permite com que um cliente chame uma função em um servidor, abstraindo em uma biblioteca, chamada *stub*, a formatação dos parâmetros e da resposta em XML e o envio da mensagem por HTTP2. Essa é a abordagem utilizada pelas ferramentas mais populares: Flower e TensorFlow Federated (TFF). Nos experimentos realizados em [Heydarishirayeh, 2023], esse protocolo foi o menos afetado por condições de rede desafiadoras. Por fim, o MQTT (*Message Queuing Telemetry Transport*) e o AMQP (*Advanced Message Queuing Protocol*) são protocolos que seguem o paradigma *publish-subscribe*, no qual consumidores se inscrevem em filas de mensagens para receber mensagens de produtores. O gerenciamento de filas, feito por um *broker*, garante assincronia na transferência de mensagens, motivando o seu uso na ferramenta FedML [He et al., 2020]. Entretanto, a implantação da infraestrutura de rede nesse cenário é mais complexa e introduz maior latência devido à dependência do intermediário.

Na camada de transporte, o TCP é preferível na maior parte dos casos devido à criação de um canal de comunicação confiável e ao controle de congestionamento. Os protocolos de aplicação baseados em MQTT/AMQP e HTTP se servem de conexões TCP, apesar de ser possível também o uso direto de soquetes TCP sem esses protocolos previamente citados. As implementações mais comuns do TCP são o BIC e o CUBIC, utilizadas no *kernel* do Linux. Entretanto, Amadeo *et al.* indicam o uso do TCP Westwood para melhorar a conexão dos enlaces sem fio do FL e do *multipaht* TCP (MPTCP) para evitar congestionamento em pontos específicos da rede de núcleo quando múltiplos clientes enviam parâmetros concorrentemente [Amadeo et al., 2025]. Cleland *et al.* demonstram que, apesar da menor sobrecarga de controle, o UDP não reduz a latência na transferência de modelos em relação ao TCP no FL em redes de borda 4G e 5G [Cleland et al., 2022]. Apesar do TCP desempenhar pior em redes mais limitadas (3G), a ausência de mecanismos de confirmação e retransmissão inviabiliza a convergência do modelo com o uso do UDP nessas condições.

2.3.3. Gerenciamento do aprendizado federado

Embora no ML centralizado típico os profissionais possam facilmente explorar diferentes modelos e hiperparâmetros, no FL, os operadores na nuvem não têm controle sobre os dispositivos onde o treinamento é realizado. Portanto, o desenvolvimento de arquiteturas para automatizar a criação, implantação, modificação e monitoramento de tarefas de FL é desejável. As operações de FL incluem o desenvolvimento de código, implantação em clientes compatíveis selecionados, monitoramento e tomada de decisão automática ou manual.

Alguns trabalhos na literatura propõem ferramentas para reduzir a sobrecarga de gerenciamento humano. Wang *et al.* desenvolvem o FLINT (FL *Integration*), que aborda os desafios relacionados à seleção de clientes com limitação de recursos heterogêneos e disponibilidade intermitente, considerando uma única tarefa [Wang et al., 2023]. Moon *et al.* implementam o FedOps para gerenciar o ciclo CI/CD (*Continuous Integration and Delivery*) dos modelos de FL resultantes [Moon et al., 2024].

Quebrar um código monolítico em microsserviços, tipicamente implantados como contêineres, facilita modificações e manutenção. Alguns autores adotaram esse conceito para construir plataformas de FL. O Micro-FL, por exemplo, melhora a escalabilidade e a tolerância a falhas de uma única tarefa de FL usando Docker e Kubernetes [Sabuhi et al., 2024]. Além disso, o FLIP (FL *Interactive Platform*) adiciona camadas sobre o arcabouço Flower para implantar tarefas como contêineres Docker, assumindo que o nó de gerenciamento tenha acesso ininterrupto aos clientes [Galende et al., 2024].

As implantações de FL são altamente dinâmicas, uma vez que: i) novos clientes podem surgir, e ii) as estratégias de treinamento podem ser alteradas pelo especialista, justificando trabalhos sobre a transferência *Over-the-Air* (OTA) do código do cliente. Huedo *et al.* distribuem máquinas virtuais para clientes de FL em infraestruturas de borda geodistribuídas, sem considerar a necessidade de atualizações em tempo real [Huedo et al., 2025]. Chahoud *et al.* implantam novos clientes em contêineres Docker *on-the-fly* [Chahoud et al., 2023a]. Como selecionar os clientes mais adequados pode ser uma tarefa muito difícil, os autores também propuseram o uso de Algoritmos Genéti-

cos e Aprendizado por Reforço [Chahoud et al., 2023b, Chahoud et al., 2025]. O FLScalizer, de Yang *et al.*, usa o GitHub para aplicar modificações nos códigos do cliente e do servidor durante a implantação contínua [Yang et al., 2023].

Thomaz *et al.* apresentam a AGATA, uma arquitetura para automatizar o provisionamento e a modificação de tarefas de FL para clientes apropriados, assumindo um cenário de CI/CD. Isso permite *pipelines* semelhantes aos geralmente encontrados em implantações centralizadas, nos quais os cientistas de dados focam apenas em aprimorar seus modelos, abstraindo a distribuição do aprendizado entre os dispositivos participantes. Além disso, a sobrecarga para executar as operações de gerenciamento na AGATA é muito menor do que a sobrecarga introduzida pela própria execução do FL [Thomaz et al., 2025].

2.4. Desafios do Aprendizado Federado Veicular

O FL tradicional possui desafios inerentes ao sistema distribuído, como a heterogeneidade dos dispositivos e dados. Para um cenário de aprendizado síncrono, a heterogeneidade de dispositivos implica a espera de clientes mais rápidos por clientes retardatários, o que prolonga a duração do treinamento do modelo global. Por outro lado, a presença de distribuições heterogêneas nos clientes faz com que os pesos gerados ao longo do treinamento tendam a divergir. Além dos desafios presentes no FL tradicional, o VFL introduz desafios devido ao alto grau de mobilidade dos clientes e às necessidades da aplicação veicular. Assim, nesta seção são inicialmente discutidos os desafios do FL em geral, devido à heterogeneidade de dispositivos e dados. Após isso, apresentam-se os desafios específicos do VFL, discutindo a heterogeneidade no canal de comunicação dos clientes e desafios específicos de aplicações veiculares.

2.4.1. Heterogeneidade de dispositivos

Um dos principais desafios para os sistemas distribuídos é assegurar a evolução do desempenho ao longo do tempo de forma justa quando os clientes possuem dispositivos heterogêneos. Em um cenário veicular os dispositivos apresentam uma alta variação de capacidade computacional. Enquanto veículos autônomos de nível 2 possuem dispositivos capazes de realizar 2,5 teraoperações por segundo (TOPS), veículos de nível 5 realizam mais de 2.000 TOPS [ElectronicsMaker, 2024].

Um desafio que a heterogeneidade de dispositivo impõe ao aprendizado federado é a seleção do modelo. Dispositivos com menor poder computacional necessitam de modelos mais leves, enquanto dispositivos mais potentes podem executar modelos mais complexos. Entretanto, o algoritmo de aprendizado federado mais popular, o FedAvg, assume que os modelos são idênticos, para realizar a média dos parâmetros dos clientes no servidor.

Utilizar modelos mais complexos no sistema permite aos clientes mais potentes atingirem um alto desempenho, limitando a participação no treinamento dos clientes com poucos recursos computacionais. Por outro lado, utilizar modelos mais simples para todos os clientes subutiliza os recursos computacionais dos clientes mais potentes, além de prejudicar o desempenho do modelo. Assim, é necessária a utilização de alternativas adequadas ao uso de modelos heterogêneos de acordo com as diferentes capacidades computacionais encontradas em ambientes veiculares. Dessa forma, os clientes podem

contribuir mutuamente ao longo do treinamento, porém utilizando modelos adequados às capacidades de seus dispositivos.

Além disso, clientes com maior poder de computação finalizam sua tarefa em um tempo menor do que os clientes com menor poder computacional, ou retardatários (*stragglers*). Dependendo da disparidade entre os dispositivos, clientes mais rápidos podem aguardar dezenas de minutos entre o fim de uma rodada e o início da próxima [Kang et al., 2025].

Assim, uma forma de mitigar este desafio é impor um tempo limite de atualização. Dessa forma, clientes que ultrapassem o tempo de atualização de uma roda têm suas atualizações descartadas e o treinamento continua sem a sua interferência. No entanto, definir o tempo limite é uma tarefa árdua, pois ao reduzir o tempo limite, menos parâmetros de clientes são utilizados para atualizar o modelo global, reduzindo a velocidade de convergência. Ao mesmo tempo, aumentar o tempo limite implica aguardar por mais tempo os clientes retardatários [Ali et al., 2025].

Essa é uma motivação para o surgimento das estratégias de coordenação alternativas apresentadas na Seção 2.2.2. Ao reduzir ou eliminar a necessidade de sincronismo entre os clientes, estratégias de coordenação como o aprendizado federado semissíncrono e o assíncrono se adequam a um ambiente com dispositivos heterogêneos [Xu et al., 2023, Wang et al., 2022, Stripelis et al., 2022, Wu et al., 2021a].

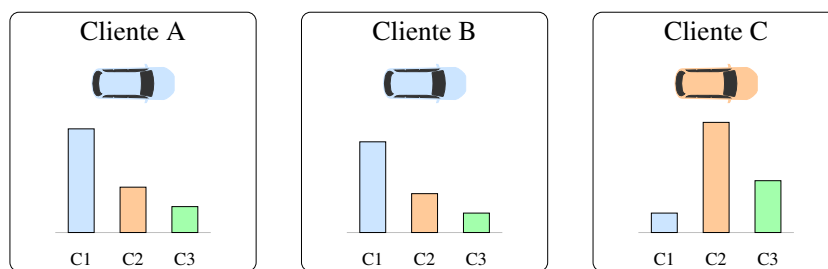
Entretanto, reduzir a necessidade de sincronismo pode penalizar clientes com maior poder computacional, uma vez que estes passam a executar mais atualizações do que clientes mais lentos. Um segundo problema mais grave é o fato de que a capacidade computacional do cliente não implica uma maior qualidade dos dados de treinamento. Assim, ao forçar que clientes com maior poder computacional realizem mais atualizações, não garante que o treinamento seja mais curto ou mesmo que o modelo treinado seja o melhor, principalmente quando os clientes possuem distribuições de dados heterogêneas [Ali et al., 2025].

2.4.2. Distribuições de dados heterogêneas

Trabalhos clássicos no contexto de FL consideram um cenário ideal, em que características dos clientes participantes no treinamento federado são distribuídas de forma homogênea. Embora avaliações nessas condições tenham demonstrado a capacidade das técnicas de FL em obter bom desempenho, esse cenário pode não representar bem casos de aplicações realistas de larga escala [Ye e ohters, 2023]. No contexto do VFL, por exemplo, a coleta dos dados não é feita de maneira uniforme. Cada veículo está inserido em um ambiente particular, com características distintas que serão representadas nos dados coletados. Um motorista que dirige no período noturno tenderá a coletar mais imagens com baixa luminosidade do que outro que apenas realiza coleta durante o dia. Se os conjuntos de dados locais dos clientes são inconsistentes e não representam bem a distribuição geral dos dados, diz-se que esses são não independentes e identicamente distribuídos (*Independent and Identically Distributed – IID*), ou não-IID [Li et al., 2022a]. Nesse contexto, Ye *et al.* propõem uma categorização das diferentes formas de heterogeneidade presentes em ambientes federados, destacando casos específicos de fontes de inconsistências entre os dados dos clientes [Ye e ohters, 2023]. A seguir, são apresentadas

essas categorias.

- **Viés de rotulação (*label skew*):** Ocorre quando as distribuições de classes dos clientes participantes são inconsistentes. Pode se manifestar como um viés de distribuição de rótulos (*label distribution skew*) [Zhang et al., 2022b], em que o processo de coleta condiciona o conjunto de dados a ter uma representação desbalanceada das classes, ou um viés de preferência de rótulos (*label preference skew*) [Kairouz et al., 2021], em que, mesmo para dados com características consistentes entre os clientes, rótulos diferentes podem ser atribuídos em cada caso. A Figura 2.4 ilustra os dois tipos de vieses de rotulação.



(a) Diagrama ilustrativo do fenômeno de viés de distribuição de rótulos.



(b) Diagrama ilustrativo do fenômeno de viés de preferência de rótulos.

Figura 2.4. Diagramas exemplificando cenários de viés de rotulação.

- **Viés de características (*feature skew*):** Ocorre quando as distribuições das características dos dados são inconsistentes entre os clientes. No contexto veicular, um possível exemplo de viés de características seria a diferença entre imagens de carros capturadas durante um inverno intenso com geadas e durante o verão. Nessas condições, seriam produzidas imagens com características distintas, mas que representam a mesma classe de objetos, caracterizando uma modalidade de viés de características, denominada viés de distribuição de características (*feature distribution skew*). Outra possibilidade corresponde ao viés de condição de características (*feature condition skew*), que ocorre quando, mesmo considerando uma mesma classe de interesse, diferentes clientes observam subconjuntos distintos. Enquanto um cliente que habita um cenário urbano pode observar predominantemente veículos leves como carros, clientes de ambientes rurais podem coletar imagens de caminhões. Mesmo que se atribua a mesma classe “Veículo” para ambos os casos, ocorrerá uma inconsistência entre as características dos mesmos.

- **Viés de qualidade (*quality skew*):** Representa inconsistência entre a qualidade da instrumentação de captura de dados e do procedimento de rotulação entre os clientes. Pode se manifestar como ruído nas amostras, como no caso de imagens ruidosas, ou como ruído nos rótulos, como no caso de rótulos de baixa qualidade produzidos por usuários sem o conhecimento necessário para produzir as anotações.
- **Viés de quantidade (*quantity skew*):** Decorre da possibilidade de ocorrer desbalanceamento das quantidades de dados entre os clientes. No contexto veicular, um usuário que dirige diariamente teria mais oportunidades de capturar dados do que um usuário que dirige uma vez por semana.

A consequência do viés introduzido pelo caráter não-IID dos dados utilizados no treinamento é uma degradação do desempenho do modelo treinado. Devido ao problema de heterogeneidade estatística descrito nesta subseção, não há garantia de alinhamento entre os gradientes dos modelos treinados localmente, o que pode gerar inconsistências no processo de agregação desses modelos. Isso ocorre porque em abordagens tradicionais de agregação, como o caso do FedAvg, os dispositivos clientes tipicamente realizam múltiplas etapas de treinamento, ajustando os parâmetros dos seus modelos na direção dos seus pontos ótimos locais, que são condicionados pelos seus dados não-IID. Se cada cliente direciona o ajuste do seu modelo para ótimos locais divergentes, o modelo agregado passa a se distanciar do ponto de ótimo global. Nesse caso, diz-se que ocorreu o fenômeno de desvio dos clientes (*client shift*), resultando em convergência mais lenta, instabilidade e desempenho possivelmente inferior ao dos modelos locais não agregados [Karimireddy et al., 2020, Ye e ohters, 2023].

2.4.3. Características da comunicação

As técnicas de rádio-frequência (RF) das redes atuais possuem limitações conhecidas em termos de atenuação/alcance e compartilhamento eficiente de recursos no canal, gerando alguns desafios na adoção do FL veicular. Para enfrentar estes desafios, alguns trabalhos de FL propõem modificações substanciais na infraestrutura das redes *6G & Beyond*. O artigo de Han *et al.* utiliza a proposta de redes integradas *ground-air-space* e comunicação satelital D2D (*Direct-to-Device*) para que dispositivos móveis façam o *offloading* do treinamento para satélites de órbita baixa (*Low Earth Orbit – LEO*) [Han et al., 2024]. Para evitar o curto tempo de cobertura, o sistema utiliza enlaces inter-satélites (*Inter-Satellite Links – ISLs*) para repassar os dados e modelo para os satélites vizinhos, que darão continuidade ao processo de FL naquela região.

Um desafio relacionado às técnicas de modulação e múltiplo acesso é a necessidade de suportar um número crescente de usuários em um cenário no qual o espectro é um recurso escasso. Chen *et al.* resolvem um problema de alocação de potência de *uplink* em um sistema de FL veicular na borda utilizando NOMA (*Non-Orthogonal Multiple Access*). Diferente do OFDMA, usado no 4G e 5G, um receptor NOMA cancela a interferência entre usuários que utilizam o mesmo bloco de recurso. Assim, múltiplos clientes enviam os parâmetros de um modelo ao mesmo tempo e na mesma frequência [Chen et al., 2024].

Além do desafio anteriormente citado, envolvendo o projeto de transmissores e re-

ceptores, existem desafios intrínsecos ao canal de comunicação sem fio. Em altas frequências, o esvanecimento (*fading*) se torna mais pronunciado, aumentando a chance de um cliente perder o *deadline* de envio do modelo ou funcionar como um *straggler* que atrasa a finalização da rodada. Zejun *et al.* propõe o uso de superfícies inteligentes reconfiguráveis (*Reconfigurable Intelligent Surfaces – RIS*) para criar um linha de visada (*Line-of-Sight – LoS*) virtual entre um veículo e a ERB [Zejun *et al.*, 2024]. Através da solução de um problema de otimização, é possível configurar dinamicamente o ângulo de reflexão de um feixe incidente em uma RIS, fazendo com que ela atue como um repetidor de sinal passivo muito mais energeticamente eficiente do que alternativas ativas, como o uso de *drones* (*Unmanned Aerial Vehicles – UAVs*). Os autores argumentam que a técnica aumenta a quantidade de dados utilizados no treinamento, o que é essencial para tornar o modelo mais útil e representativo.

Existem alguns desafios em aberto quanto aos protocolos utilizados para FL veicular. A camada de aplicação precisa lidar com clientes e servidores que se desconectam ou demoram muito tempo para enviar suas contribuições. Zhangchen *et al.* propõem um protocolo para FL que modifica toda a FSM de troca de mensagens em um cenário P2P [Xu *et al.*, 2024b]. Utilizando privacidade teórica da informação (*Information-theoretic privacy*), os autores provam a tolerância à falhas bizantinas na presença de f nós maliciosos se o número total de nós for $N = 3f + 2$.

Alguns trabalhos demonstram limitações de desempenho importantes do TCP em condições de redes desafiadoras. O atraso introduzido pelo *three-way handshake* é inadequado quando o contato entre cliente e servidor ocorre em um período de tempo muito breve, o que é comum em locais com cobertura incompleta. Valores baixos de janela de congestionamento em redes sem fio também foram reportados em experimentos da literatura [Thomaz *et al.*, 2024]. Um trabalho recente de Chen *et al.* propôs um protocolo de transporte disruptivo, com controle de congestionamento otimizado especificamente para aprendizado distribuído, com foco em envio de parâmetros 1:N (*in-cast*) [Chen *et al.*, 2023b].

Por fim, as redes IP não proveem garantias como cache, agregação de pacotes, *multicast* nativo, endereçamento móvel e Notificação Explícita de Congestionamento (*Explicit Congestion Notification – ECN*). Amadeo *et al.*, por exemplo, mostram que as Redes Centradas em Informação (*Information Centric Networks – ICNs*) permitem com que a própria rede participe ativamente da descoberta, seleção e inicialização dos clientes do FL [Amadeo *et al.*, 2025]. A arquitetura *Named Data Entity* (NDN) define um esquema próprio de controle de congestionamento na camada de transporte, chamado de PCON (*Practical Congestion Control*), que implementa Gerenciamento Ativo de Fila (*Active Queue Management – AQM*) nas interfaces dos roteadores da ICN NDN para permitir ECN. Um dos diferenciais do artigo é o fato de que os resultados comparativos entre mecanismos de controle de congestionamento (TCP BIC, TCP CUBIC, TCP Westwood e PCON) demonstram que a topologia de rede influencia o desempenho, sugerindo que o projeto adequado da interconexão entre elementos no núcleo também é um desafio.

2.4.4. Características da aplicação

O paradigma do FL difere do CL não somente pela distribuição do treinamento dos modelos, mas também pelo processo de coleta e anotação dos dados utilizados. No CL, a ausência de restrições sobre a transmissão de dados permite que o custo associado à rotulação seja abstraído do usuário final do sistema, uma vez que torna possível o envio desses dados para que terceiros produzam as respectivas anotações. Já no caso do FL, a premissa de privacidade que justifica a abordagem impede a transferência dessa responsabilidade, centralizando no usuário todo custo envolvido no procedimento de rotulação.

O problema de falta de rótulos é particularmente acentuado no contexto veicular. Muitas tarefas de interesse como detecção de veículos, rastreamento de objetos e segmentação semântica dependem de rótulos complexos, envolvendo a definição de caixas delimitadoras para cada objeto na cena capturada ou a produção de máscaras de segmentação, atribuindo uma classe a cada instância em ambos os casos. Esse elevado custo associado à produção de anotações promove um cenário de escassez de rótulos. No entanto, essa escassez não necessariamente se distribui de forma homogênea pelo sistema. É possível, por exemplo, que o servidor tenha acesso a conjuntos de dados públicos que contenham informação relevante ou mesmo dados que tenham sido previamente coletados e rotulados, servindo como uma possível referência para o treinamento. Apesar de os dados coletados pelos clientes serem majoritariamente não rotulados, alguns clientes podem dispor de recursos para realizar a rotulação de uma parcela ou a totalidade dos seus dados.

Trabalhos na literatura mapeiam, exploram e propõem soluções para diferentes cenários de rotulação, sendo os casos de rótulos nos clientes ou rótulos no servidor os mais predominantes [Jeong et al., 2021, Ji et al., 2024]. A Figura 2.5 apresenta uma taxonomia das diferentes combinações de dados rotulados e não rotulados pertencentes aos participantes no treinamento federado, relacionando os cenários com abordagens de aprendizado federado semissupervisionado (*Federated Semi-Supervised Learning* – FSSL).

Outro problema que dificulta o treinamento dos modelos utilizados em contextos veiculares, especialmente quando voltados para tarefas de visão computacional, é que comumente essas aplicações impõem restrições de latência severas, devendo ser capazes de operar em tempo real. Um modelo de detecção de objetos que contribui para a tomada de decisões de um sistema de direção autônoma deve ser capaz de processar o fluxo de imagens recebido da câmera em tempo suficientemente curto para que não atrase o restante do sistema. Para atender a esses requisitos, detectores de objetos de um estágio são frequentemente escolhidos, uma vez que comumente apresentam menor tempo de inferência quando comparados aos detectores de dois estágios.

Entretanto, detectores de um estágio apresentam limitações quando utilizados no contexto de aprendizado semissupervisionado (*Semi-Supervised Learning* – SSL). Técnicas propostas para esse contexto tipicamente dependem da geração de pseudo-rótulos, ou seja, utilizam rótulos gerados pelo próprio modelo em seu treinamento. Em detectores de um estágio, a ausência de uma etapa separada de geração de regiões de interesse e de refinamento dessas predições torna a filtragem de pseudo-rótulos de melhor qualidade mais difícil, potencialmente levando ao uso de anotações incorretas ou desbalanceadas [Zhang et al., 2022d, Kim et al., 2023]. Dessa forma, a construção de estratégias de

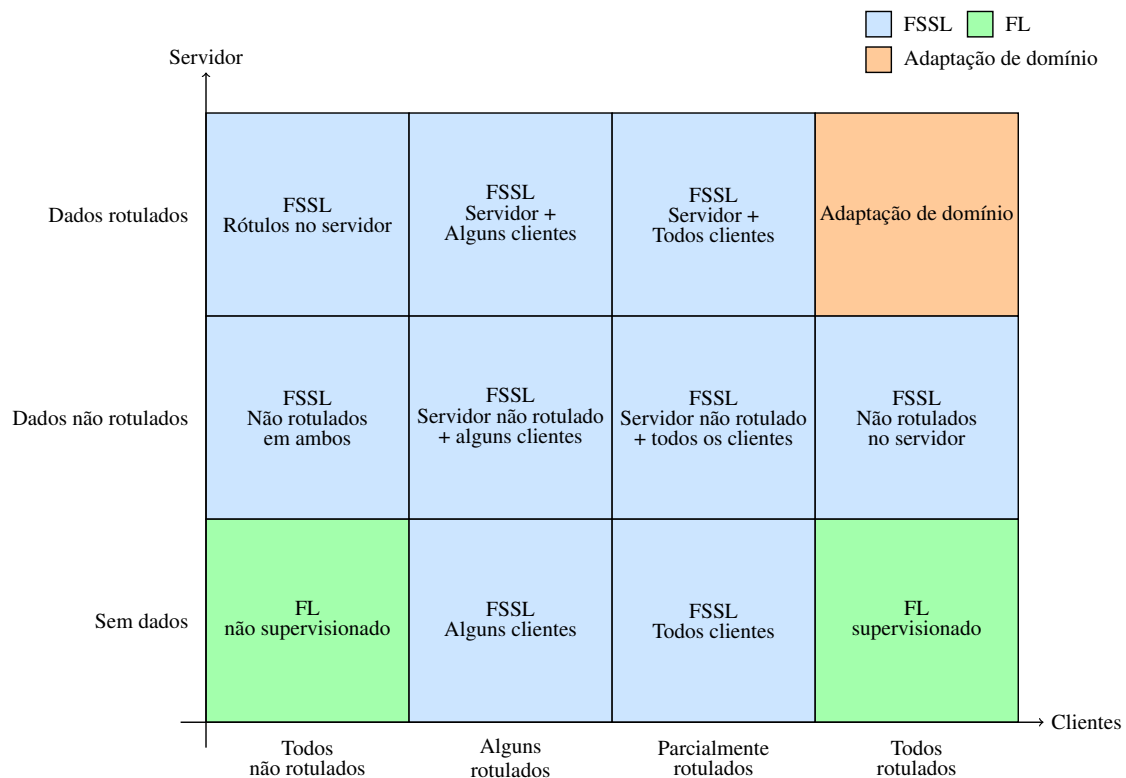


Figura 2.5. Taxonomia de cenários de aprendizado federado em diferentes cenários de disponibilidades de rótulos. Blocos azuis representam casos típicos de aplicações de FSSL. Blocos verdes representam casos de FL clássico. Já o bloco laranja representa um caso específico de aplicação da técnica de adaptação de domínio.

FSSL que suportem arquiteturas capazes de atender aos requisitos de latência das aplicações veiculares e sejam capazes de lidar com a baixa qualidade dos rótulos gerados por essas arquiteturas é um desafio adicional para o desenvolvimento de sistemas de VFL.

2.5. Atividade Prática

Esta seção descreve a atividade prática do minicurso, que foi concebida como uma simulação realista e detalhada do aprendizado federado aplicado especificamente ao contexto veicular, integrando de forma coesa os aspectos teóricos discutidos anteriormente. Primeiramente, apresenta-se o conjunto de dados para a tarefa de aprendizado que será abordada na atividade prática. O treinamento prático inicia-se com a demonstração de como simular a heterogeneidade dos clientes, abordando tanto as diferenças de hardware entre os clientes quanto a diversidade nos volumes e tipos de dados coletados por cada veículo. Esta etapa é crucial para que os participantes compreendam como os modelos de inteligência artificial se comportam diante da variabilidade encontrada em redes veiculares de larga escala no mundo real. Para conferir autenticidade ao experimento, a atividade aborda a geração de traços de mobilidade por meio do *Simulation of Urban MObility* (SUMO)¹ [Lopez et al., 2018], uma ferramenta para simular o movimento de veículos em ambientes urbanos. Em conjunto com a aplicação de modelos de canal de

¹Disponível em: <https://eclipse.dev/sumo/>

comunicação, esses traços serão utilizados para gerar estatísticas precisas sobre os clientes, oferecendo uma base de dados sólida para a análise do sistema. Além disso, o curso utiliza ferramentas de visualização, que permitem a demonstração clara e intuitiva dos fluxos de mobilidade e das dinâmicas de comunicação que ocorrem durante o processo de aprendizado.

A fase final da parte prática foca na implementação técnica utilizando o arcabouço Flower [Beutel et al., 2020], adaptado para incorporar os traços de computação, mobilidade e comunicação gerados nas etapas anteriores. Por meio desta plataforma, os participantes têm a oportunidade de avaliar o impacto real dos diferentes parâmetros configurados no aprendizado federado, observando como cada ajuste influencia o desempenho final do modelo treinado de forma distribuída. Essa abordagem permite uma compreensão profunda das técnicas de otimização necessárias para garantir que os sistemas inteligentes de transporte operem com máxima eficiência e segurança. Esta seção descreve as etapas a serem executadas ao longo da atividade prática. A definição dos pré-requisitos, dados necessários e comandos a serem executados estão disponíveis no repositório desse capítulo².

Ao fim da atividade prática, os participantes compreenderão de forma abrangente e aplicada o funcionamento das ferramentas essenciais apresentadas, estando plenamente capacitados para adaptá-las à implementação de novas e avançadas técnicas de aprendizado federado no contexto veicular.

2.5.1. Definição do cenário federado

O conjunto de dados a ser utilizado na atividade prática será o *Traffic Signs Preprocessed*³ [Stallkamp et al., 2011]. Este conjunto de dados possui próximo a 50.000 imagens de treino e 12.000 imagens de teste e tem como tarefa de aprendizado a classificação de placas de trânsito alemãs, divididas em 43 classes. Assim, o classificador recebe uma imagem de 32 x 32 pixels com 3 canais de cores RGB e retorna a classe mais provável. Para o cenário abordado na prática, utilizam-se 20 clientes, cujas definições de dados e dispositivos são apresentadas a seguir.

2.5.2. Modelagem da heterogeneidade de clientes

No ambiente federado, há duas fontes principais de heterogeneidade: os dados coletados e os dispositivos. Como visto na Seção 2.4, a heterogeneidade dos dados está relacionada ao ambiente em que os dados foram coletados, ao método de coleta, às configurações e ao equipamento utilizado. Por outro lado, a heterogeneidade de dispositivos está relacionada a características físicas, como a frequência de processamento, a quantidade de memória e o número de núcleos de processamento disponíveis.

2.5.2.1. Heterogeneidade de Dados

Ao contrário de cenários centralizados nos quais um servidor armazena o conjunto de dados completo, no aprendizado federado considera-se que os dados estão distribuídos

²Disponível em: <https://github.com/GTA-UFRJ/MinicursoAprendizadoFederadoVeicularSBRC2026>

³Disponível em: <https://www.kaggle.com/datasets/valentynsichkar/traffic-signs-preprocessed>

entre os dispositivos participantes do treinamento. Em tarefas de classificação de imagens, como a abordada na atividade prática, os clientes possuem conjuntos de dados sem interseção, uma vez que a captura dos dados depende de sua posição, condições da câmera e do ambiente no momento da captura.

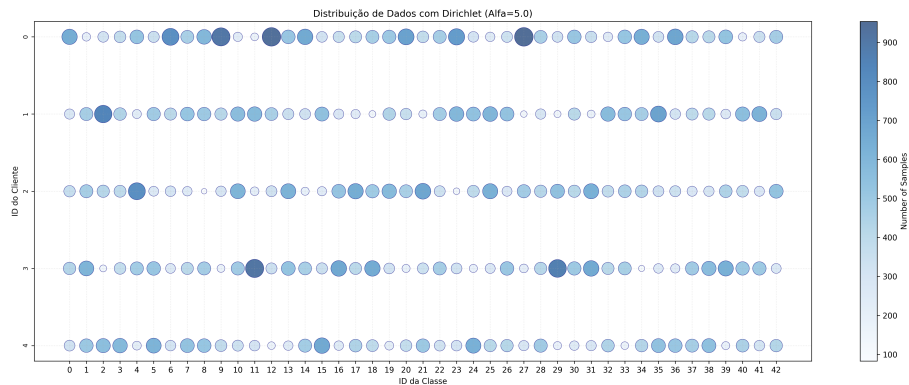


Figura 2.6. Distribuição dos dados de clientes ao utilizar a função de Dirichlet com $\alpha = 5$. O conjunto de dados gerados é mais homogêneo quando o valor do parâmetro α é maior.

Assim, a distribuição de dados afeta o desempenho de propostas de aprendizado federado. Cenários realistas de treinamento consideram a variação da distribuição de dados, que é inexistente em conjunto de dados capturados para realizar o treinamento centralizado. Portanto, para gerar subconjuntos sem interseção e simular a heterogeneidade de dados, a função de Dirichlet [Ferguson, 1973] é amplamente utilizada. A função recebe dois parâmetros, o número de conjuntos de dados esperado e o $0 \leq \alpha < \infty$. O número de conjuntos de dados será igual ao número de clientes presentes na simulação, enquanto o α é o parâmetro responsável por controlar o grau de heterogeneidade dos dados. Um valor alto de α , tipicamente maior do que 5, gera distribuições semelhantes, como exibido na Figura 2.6, enquanto valores menores geram conjuntos de dados mais desbalanceados, como exibido na Figura 2.7.

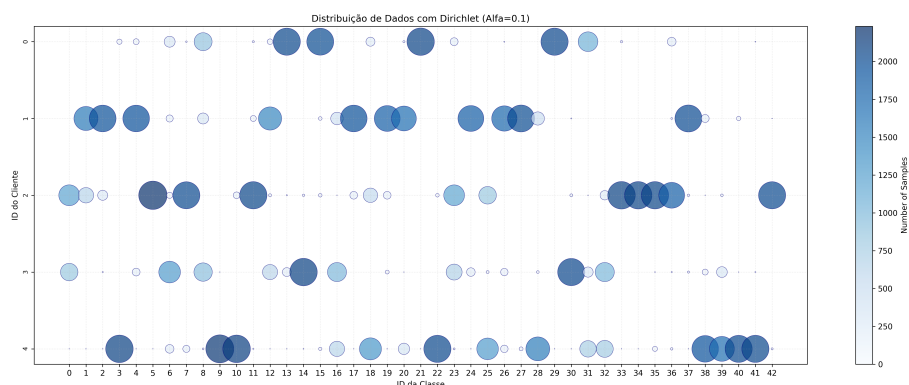


Figura 2.7. Distribuição dos dados de clientes ao utilizar a função de Dirichlet com $\alpha = 0,1$. O conjunto de dados gerados é desbalanceado segundo o número de amostras de cada classe para diferentes clientes.

Dessa forma, pode-se gerar conjuntos de dados heterogêneos, com dados únicos,

para executar o treinamento simulando a existência de diversos clientes. Além disso, a heterogeneidade de dados abrange um escopo maior do que o cenário veicular, aplicada em diversos cenários de aprendizado federado estáticos ou dinâmicos. A seguir, discute-se como simular a heterogeneidade dos dispositivos dos clientes.

2.5.2.2. Heterogeneidade de Dispositivo

Em um cenário veicular, como visto na Seção 2.4.1, os clientes apresentam diferentes dispositivos. Veículos autônomos de níveis baixos realizam algumas unidades de teraoperações por segundo (TOPS), enquanto que veículos completamente autônomos executam milhares de teraoperações. Dessa forma, veículos que possuem maior capacidade computacional, ao treinarem o mesmo modelo de aprendizado profundo, atualizam de forma mais rápida do que veículos com menor capacidade computacional.

Assim, nesta parte da atividade prática modela-se por meio de um modelo matemático o tempo de processamento de clientes no aprendizado federado veicular a partir das especificações do dispositivo do cliente. O modelo matemático permite que o sistema seja executado em uma mesma máquina, o que é necessário para casos em que não há disponibilidade de dispositivos heterogêneos. Caso haja disponibilidade de diversos dispositivos, é possível substituir o modelo matemático pelo tempo de computação do próprio dispositivo. Outra parte essencial para o VFL é a simulação da mobilidade dos clientes, para a qual utiliza-se um simulador de mobilidade discutido a seguir.

2.5.3. Simulação da mobilidade de veículos com o SUMO

Uma vez definidas as características de dispositivo, deve-se simular o movimento dos veículos de forma realista. O simulador SUMO (*Simulation of Urban MObility*)⁴ modela características microscópicas da dinâmica de tráfego urbano. O SUMO calcula o comportamento de cada veículo de forma individual e, com isso, também considera a interação entre eles. Elementos como interseções, sinais, placas de trânsito, pedestres podem ser adicionados para adicionar mais elementos que corroboram uma simulação mais realista. Uma simulação é construída com base na escolha do mapa e na definição do comportamento dos veículos na simulação. A simulação retorna uma lista com a posição de todos os veículos ao longo da simulação, e outras ferramentas utilizam essa lista para determinar a posição dos elementos no instante em que se realiza uma transmissão.

O site do SUMO possui uma documentação bem detalhada sobre os diversos elementos que podem ser adicionados na simulação para tornar os dados mais próximos de reais. Os principais parâmetros necessários para a configuração da simulação são o movimento dos veículos, quantidade de veículos na simulação, velocidade máxima das vias e o mapa a ser utilizado. Além disso, é possível configurar pontos de interesse, como estações rádio base e pontos de parada.

Para a geração dos movimentos dos veículos utiliza-se a ferramenta **random-Trips**. A partir da especificação da quantidade de veículos e informações sobre a velocidade e taxa de entrada, a ferramenta gera voltas aleatórias no mapa especificado. A

⁴<https://eclipse.dev/sumo/>

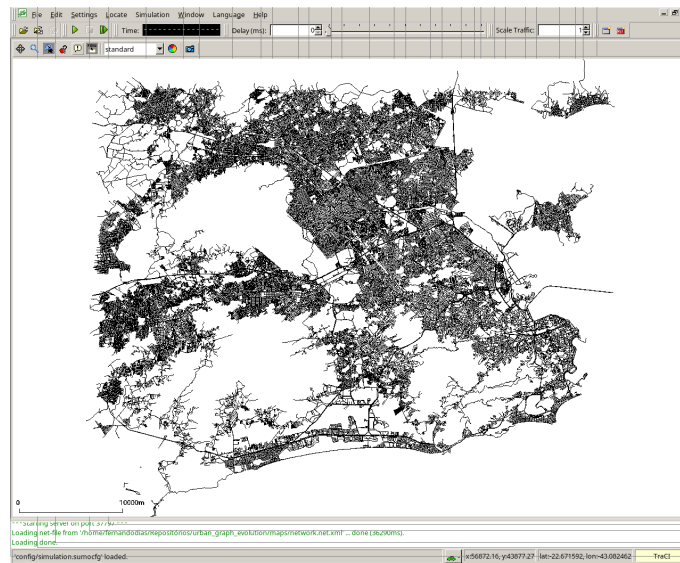


Figura 2.8. Ilustração do uso do mapa da região metropolitana do Rio de Janeiro.

fim de manter todos os clientes ao longo do treinamento, configura-se a ferramenta para que os veículos entrem na simulação ao mesmo tempo. Além disso, para simplificar a execução, todos os veículos viajam com a velocidade máxima da via.

O mapa é um elemento que pode ser obtido de um cenário real ou criado sinteticamente. Por exemplo, o mapa exibido na Figura 2.8 exibe o mapa real da cidade do Rio de Janeiro. Esse mapa é obtido por meio do OSM (*Open Street Map*)⁵, uma ferramenta para exportar mapas realistas com informações precisas sobre velocidade de vias e pontos de interesse. Entretanto, para fins práticos é comum utilizar cenários sintéticos, que simulam a cidade de Manhattan, por exemplo. Assim, nessa atividade prática, utiliza-se um mapa quadrado de 1000 x 1000 m², configurado de forma que as rotas sejam circulares, para evitar a saída dos clientes ao longo da simulação do aprendizado federado, com uma estação rádio base na origem como um ponto de interesse.

A seguir, utilizam-se os dados gerados como entrada do modelo de comunicação para determinar a vazão dos clientes e verificar o impacto no treinamento federado veicular.

2.5.4. Simulação da comunicação

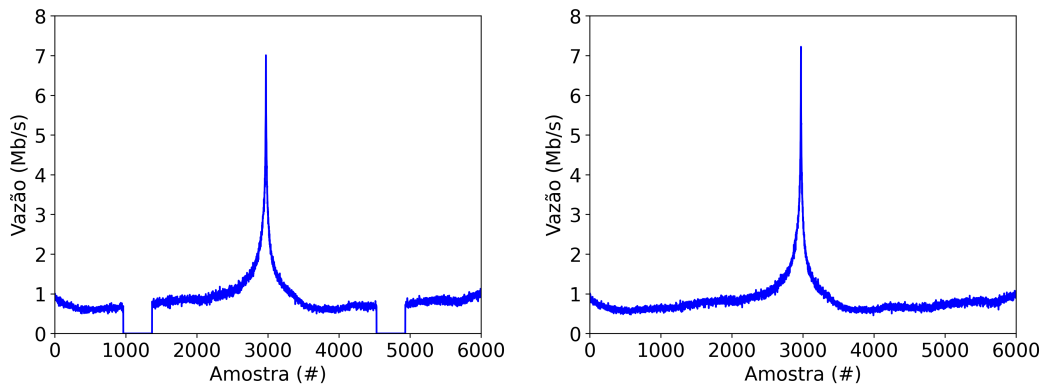
O modelo de comunicação recebe a saída processada do SUMO para determinar os atrasos de comunicação dos clientes. Para o modelo de comunicação, utiliza-se o 3GPP TR 38.901 [Zhu et al., 2021a, Chatzoulis et al., 2023], o modelo de canal mais versátil e amplamente aceito. Ele é capaz de lidar com cenários urbanos, rurais e rodoviários com parâmetros realistas para desvio Doppler, perda de percurso (*path loss*) e propagação multicaminho.

A implementação para modelar o canal 3GPP TR 38.901 para o código de comunicação sem fio foca nos modelos de perda de percurso (*path loss*) e desvanecimento

⁵<https://www.openstreetmap.org>

(*fading*) de Macro célula Urbana (UMa - Urban Macrocell) e Micro célula Urbana (UMi - Urban Microcell). Assim, a implementação inclui o cálculo de perda de percurso (*path loss*) para cenários com visada direta (*Line-of-Sight* – LOS) e sem visada direta (*Non-Line-of-Sight* – NLOS), desvio Doppler (*Doppler shift*) para consideração da mobilidade, desvanecimento de pequena escala (*small-scale fading*) utilizando os modelos de desvanecimento de Rician e Rayleigh e sombreamento (*shadowing*) para desvanecimento de larga escala (*large-scale fading*).

Para a perda de percurso, considera-se o modelo de Macro célula Urbana (UMa - Urban Macrocell) do 3GPP TR 38.901, que inclui condições LOS e NLOS com efeitos de sombreamento. Para o desvio Doppler, precisa-se simular o efeito do movimento relativo entre o usuário e a estação rádio-base. Esse movimento é calculado com base na velocidade do cliente e frequência da portadora. Assume-se o desvanecimento de pequena escala considerando o desvanecimento Rician e o desvanecimento Rayleigh. Para detalhes sobre o modelo matemático utilizado, consulte o repositório de referência⁶.



(a) Vazão em um cenário com desconexão de 11 segundos. (b) Vazão em um cenário com conexão constante.

Figura 2.9. Visualização da vazão média de um cliente executando a ferramenta de simulação com uma estação rádio base.

Portanto, utiliza-se o modelo de mobilidade como a entrada do modelo de comunicação para gerar os atrasos relativos ao sistema, ilustrados na Figura 2.9. Na Figura 2.9(a)

A seguir, discute-se como adaptar o arcabouço Flower para incluir os atrasos de computação e comunicação gerados para simular a heterogeneidade e mobilidade dos clientes.

2.5.5. Adaptação do arcabouço Flower para o treinamento federado veicular

A simulação com o arcabouço se baseia no desenvolvimento de dois arquivos principais, um para o cliente e um segundo para o servidor. No lado do cliente, é necessária a implementação de ao menos dois métodos, o método para o ajuste dos pesos do modelo e o método de avaliação do modelo. O primeiro método é executado no início de cada rodada, sendo invocado pelo servidor por meio de uma chamada gRPC. O segundo método é invocado no fim de cada rodada para testar o desempenho do modelo treinado até

⁶Disponível em <https://github.com/AiramL/TimeOptimizedFederatedLearning>

o momento. Ambas as funções permitem o envio de dicionários de parâmetros, os quais podem ser utilizados para a implementação de estratégias no servidor.

O servidor, por sua vez, utiliza um objeto que deve implementar a estratégia do aprendizado federado. Assim, a estratégia define como realizar a agregação dos modelos de diferentes clientes, selecionar os clientes e processar parâmetros enviados. Para isso, a estratégia possui métodos que implementam a fase de treino e de avaliação de clientes. Em cada fase, existe um método de seleção de clientes que pode variar, por exemplo, na quantidade de clientes selecionados em cada fase.

Portanto, a adaptação do arcabouço para o aprendizado veicular consiste em modificar os métodos relativos ao cliente para incluir parâmetros para simular a mobilidade e a heterogeneidade dos dispositivos. Assim, os clientes devem simular o atraso do recebimento do modelo, a atualização local dos parâmetros e o reenvio ao servidor. Ao variar os parâmetros de mobilidade, comunicação e computação, é possível verificar o impacto na eficiência e tempo de treinamento ao utilizar diferentes estratégias adaptadas ao VFL.

2.6. Tendências de Pesquisa

Esta seção apresenta as principais tendências de pesquisa em aprendizado federado para solucionar os desafios práticos abordados na Seção 2.4. As tendências são divididas em seis linhas de pesquisa e analisadas conforme a viabilidade de implementação em um cenário veicular. Inicialmente, abordam-se as propostas de seleção de clientes que substituem a seleção aleatória. Após isso, discutem-se estratégias que alteram a função de agregação de modelos para consideração da heterogeneidade de dados. Além disso, são discutidos a personalização de modelos e a sua subárea de pesquisa, o agrupamento de clientes para o aumento da acurácia do modelo e redução do tempo de convergência quando os clientes possuem distribuições de dados heterogêneos. Ainda, apresenta-se a alternativa para o treinamento federado com múltiplos modelos, a fim de mitigar problemas ocasionados pela heterogeneidade dos dispositivos ou do canal de comunicação de clientes. Por fim, são apresentados os novos paradigmas de aprendizado que buscam solucionar problemas específicos de aplicações, por exemplo, lidar com rótulos faltantes.

2.6.1. Seleção de clientes

A seleção de clientes no aprendizado federado consiste em definir um subconjunto de clientes para participar de uma determinada rodada de treino. É comum, especialmente em cenários veiculares, a heterogeneidade nas características dos clientes. Elementos como a distribuição de dados locais, a capacidade computacional dos dispositivos de treino de cada cliente, a qualidade da conexão e até mesmo a distribuição geográfica dos clientes são fatores que impactam a qualidade dos gradientes locais, afetando o modelo global e o tempo de treino de uma rodada. Os métodos de seleção de clientes são baseados em: otimização de recursos, importância estatística, aprendizado por reforço e reputação [Soltani et al., 2022, Fu et al., 2023, Li et al., 2024]. Os métodos baseados em reputação estão fora do escopo deste minicurso, pois visam um cenário no qual os clientes podem ser maliciosos.

2.6.1.1. Métodos baseados em otimização de recursos

Os métodos baseados em otimização de recursos formulam uma função objetivo a ser maximizada ou minimizada a fim de selecionar os melhores clientes. Esses métodos podem considerar diversas métricas na formulação do problema de otimização, como a capacidade computacional, largura de banda disponível ou localização geográfica dos clientes. Nem sempre a solução ótima dos problemas propostos pode ser calculada em tempo hábil, sendo comuns aproximações da solução por algoritmos gulosos.

Nishio e Yonetani propõem um protocolo para seleção de clientes no aprendizado federado [Nishio e Yonetani, 2019] no qual os clientes com maiores capacidades de processamento e menor latência de comunicação são priorizados no esquema de seleção apresentado. [Buyukates e Ulukus, 2021] consideram um cenário de aprendizado federado no qual o servidor de agregação realiza uma seleção aleatória e descarta parte dos modelos treinados pelos clientes. O objetivo é aumentar a velocidade de treinamento do aprendizado federado selecionando os primeiros m clientes que enviam o modelo ao servidor de agregação. A vantagem da proposta é utilizar uma heurística simples que dispensa a formulação de um problema de otimização para a seleção de clientes. No entanto, a proposta introduz um desperdício computacional para os clientes retardatários que foram selecionados.

[Su et al., 2024] propõem um algoritmo de seleção de clientes para o *Online Federated Learning* (OFL), chamado LCCS (*Low-Cost Client Selection*). Os autores formulam um problema de otimização para maximizar a utilidade do modelo e minimizar o custo de comunicação. A proposta é relevante para cenários veiculares, pois as condições de trânsito, como o engarrafamento, geram áreas com mais clientes. Logo é interessante otimizar a largura de banda para reduzir o custo de comunicação.

O TOFL (*Time Optimized Federated Learning*) [de Souza et al., 2025] é uma proposta para seleção de clientes no aprendizado federado veicular que considera a mobilidade dos clientes. Como a posição dos clientes é dinâmica, suas condições de conexão variam. Isso aumenta o tempo de treinamento do modelo devido ao atraso de comunicação com o servidor. Assim, os autores formulam um problema de otimização que considera os atrasos de cada cliente para selecionar os melhores em uma rodada a fim de reduzir o tempo total de treinamento em cada rodada. No entanto, o atraso é uma variável desconhecida, pois é obtido apenas após a seleção. Portanto, o TOFL, a partir de informações sobre as vazões anteriores dos clientes, utiliza um modelo LSTM para prever a vazão futura dos clientes e determinar os próximos atrasos. A seleção, apesar de reduzir o tempo de treinamento em determinadas condições, possui limitações, como a qualidade do estimador em determinar as condições dos clientes. Uma alternativa para evitar o descarte de atualizações de forma mais eficiente é executar o estimador de atrasos no cliente, como feito em CAIROS [de Souza et al., 2026]. Assim, os clientes podem enviar antecipadamente o modelo quando detectam que as condições de comunicação estão se degradando. Entretanto, como a proposta aumenta a complexidade do cliente, CAIROS é inadequado para o caso em que os clientes possuem recursos computacionais limitados.

Apesar dos métodos reduzirem o tempo de execução de uma rodada, a formulação proposta desconsidera o número de rodadas totais. Isso ocorre pois os métodos otimizam

a seleção de cliente considerando a heterogeneidade de sistema, isto é, a heterogeneidade dos dispositivos participantes. Por outro lado, o número total de rodadas pode ser otimizado considerando a relevância dos dados dos clientes para a atualização do modelo global. Dessa forma, há um compromisso entre o número de rodadas e o tempo de cada rodada, resolvido pelos métodos baseados em importância estatística.

2.6.1.2. Métodos baseados em importância estatística

A qualidade dos dados utilizados para o treinamento dos modelos impacta o desempenho do modelo treinado, além de reduzir a quantidade de rodadas totais para o treinamento. Assim, outros métodos de seleção de cliente buscam clientes com maior influência positiva na atualização do modelo global. O objetivo de métodos baseados em importância é acelerar a convergência do modelo global através da seleção dos clientes que possuem dados mais relevantes. Em alguns casos, esses métodos podem também considerar a otimização de recursos computacionais, como em Oort [Lai et al., 2021], que considera a heterogeneidade do sistema na formulação do problema de otimização.

[Luo et al., 2022] e [Lai et al., 2021] propõem esquemas de seleção de clientes que otimizam a velocidade de convergência do modelo em ambientes de aprendizado federado. Selecionar clientes baseando-se unicamente na representatividade dos dados diminui o número total de épocas para a convergência do modelo. No entanto, clientes que possuem dados mais relevantes para o problema podem ter um tempo de treinamento mais longo a cada época, de acordo com a quantidade de dados e as características do dispositivo utilizado. Por um lado, o aumento do tempo entre as épocas implica um atraso geral maior, uma vez que os ambientes de FL são geralmente síncronos e aguardam todos os clientes enviarem seus modelos locais ou atingirem o tempo limite. Por outro lado, selecionar clientes com maior poder computacional para reduzir o tempo entre as rodadas pode resultar em mais rodadas para a convergência, caso os clientes selecionados possuam dados estatisticamente insignificantes para a tarefa de aprendizado. Existe um compromisso entre o número de épocas para o treinamento do modelo e o tempo total de cada rodada. Assim, os autores consideram simultaneamente as características dos dispositivos e a relevância da distribuição de seus dados para reduzir o tempo de convergência do modelo global. Enquanto Luo *et al.* propõem uma função não convexa em relação ao número esperado de épocas, baseada nas atualizações anteriores dos clientes, [Lai et al., 2021] evitam a programação linear, criando uma função de seleção que avalia as perdas de treinamento dos clientes.

FedSA [Neto et al., 2022] utiliza uma meta heurística baseada no arrefecimento simulado para o ajuste dos hiperparâmetros e a seleção eficiente de clientes. A partir do estado atual do modelo, a proposta seleciona um grupo de clientes para realizar o treinamento na rodada atual. Já [Rai et al., 2022] propõem a métrica de amostragem por irrelevância para seleção de clientes, visando melhorar a acurácia final de modelos de aprendizado federado em cenários IID e não-IID. O objetivo é selecionar clientes considerando a qualidade e a quantidade de suas amostras. Cada cliente calcula sua métrica de amostragem por irrelevância e a envia ao servidor. O servidor, então, agrupa os clientes de acordo com o valor informado em três clusters: positivo, negativo e zero. Finalmente,

a metodologia proposta seleciona aleatoriamente os clientes de cada cluster para alcançar uma convergência mais rápida do modelo.

Porém, os métodos baseados em importância estatística são muito complexos devido à estocasticidade e à não causalidade do problema, uma vez que o servidor obtém a acurácia de teste do modelo após a seleção. Assim, uma forma de reduzir a complexidade do problema da seleção de clientes é delegar a tarefa a um modelo de aprendizado por reforço.

2.6.1.3. Métodos baseados em aprendizado por reforço

Métodos baseados em aprendizado por reforço profundo (*Deep Reinforcement Learning – DRL*) utilizam redes neurais profundas para abstrair políticas de seleção de clientes. Esses métodos formulam um espaço de estados e de ações onde o modelo de DLR é treinado, aprendendo padrões intrínsecos do cenário em uma representação de alta dimensionalidade a partir da interação com o ambiente.

FAVOR [Wang et al., 2020a] utiliza DRL na seleção de clientes em um cenário onde os clientes possuem uma distribuição de dados locais não IID. A abordagem consiste em treinar uma rede (*Deep Q-Network – DQN*) em um cenário onde o espaço de estados é definido como o conjunto de pesos do modelo global e o conjunto de pesos do modelo local de cada cliente. Devido à alta dimensionalidade resultante, aplica-se uma técnica de redução de dimensionalidade sobre esses pesos antes que o agente selecione o subconjunto ideal para o treinamento. Já os autores em [Zhang et al., 2021] utilizam o algoritmo *Deep Deterministic Policy Gradient* para realizar DRL para a seleção de clientes em um cenário de (*Internet of Things – IoT*) considerando o custo de treinamento, de comunicação e a qualidade dos modelos locais.

Um cenário veicular é abordado em [Moon e Lim, 2024], onde clientes realizando computação veicular (*Vehicular Edge Computing – VEC*) treinam um modelo em aprendizado federado que será enviado a um servidor para agregação. A função de recompensa considera a acurácia do modelo global no *dataset* de teste, enquanto a observação considera o tempo de processamento total e a sobrecarga de comunicação em uma rodada para modelar um cenário de aprendizado por reforço. O algoritmo DQN foi utilizado para treinar um modelo de DRL nesse cenário. Ademais, o trabalho apresenta outro modelo de seleção baseado em lógica *fuzzy*.

A seleção de clientes é desafiadora devido à sua complexidade, principalmente em ambientes veiculares, onde o atraso por rodada é dependente da mobilidade dos clientes selecionados. Além disso, reduzir a complexidade da aplicação do cliente é necessário em casos de computação limitada. Assim, uma forma de reduzir o tempo de convergência do modelo, mantendo a aplicação do cliente simples, é modificar no servidor a função utilizada para a agregação, considerando a heterogeneidade dos clientes.

2.6.2. Estratégias de modificação da função de agregação

O servidor de agregação no FedAvg realiza uma média ponderada dos pesos da rede neural para a agregação de diferentes modelos. A contribuição de cada cliente na

média ponderada é dada pela quantidade de dados que o cliente possui sobre a quantidade de dados totais utilizados para a agregação da rodada atual. Entretanto, essa estratégia não é ótima quando os clientes possuem dados ou dispositivos heterogêneos. Assim, novas estratégias buscam otimizar a contribuição dos clientes na agregação a fim de reduzir o tempo de convergência e aumentar o desempenho do modelo.

Uma das primeiras propostas de agregação para substituir o FedAvg foi o FedProx [Li et al., 2020]. A proposta modifica a função de agregação para lidar com a heterogeneidade de dispositivos e de dados. Para evitar que as atualizações dos clientes retardatários sejam descartadas, os autores propõem o uso de épocas locais variáveis, ajustadas à capacidade computacional do cliente. Além disso, o desafio da heterogeneidade de dados é resolvido por meio da aplicação de um termo proximal para regularizar as atualizações. Esse termo é otimizado e aplicado localmente pelo cliente para considerar o impacto da heterogeneidade de dados e da quantidade distinta de épocas locais executadas na atualização. O objetivo do termo proximal, que regulariza o modelo, é produzir atualizações que sejam próximas do modelo global da rodada, garantindo maior consistência.

A proposta do SCAFFOLD (*Stochastic Controlled Averaging algorithm*) [Karimireddy et al., 2020] também modifica como os modelos são agregados ao aplicar uma variável de regularização para reduzir a variância das atualizações. A cada passo local, a variável de controle é aplicada para evitar que o modelo local se distancie do modelo global. A variável de controle é armazenada por cada cliente e pelo servidor, a fim de compartilhar o estado entre rodadas diferentes, sendo atualizada ao final de cada rodada. Dessa forma, a variável de controle possui um efeito similar ao do termo proximal do FedProx. No entanto, como a variável de controle atua no espaço de gradientes, esta permite uma maior exploração do espaço de soluções. Por outro lado, o termo proximal atua no espaço do modelo e reduz o tamanho dos passos, aumentando o tempo de convergência em comparação com o SCAFFOLD.

FedNova (*Federated Normalized averaging*) [Wang et al., 2020b] é uma proposta que assume que os clientes podem executar um número diferente de épocas locais de acordo com as condições do dispositivo. Assim, para resolver a inconsistência causada pela diferença na quantidade de atualizações locais, os autores propõem uma normalização da atualização, considerando a quantidade de computação realizada por cada cliente. Essa estratégia é executada diretamente no servidor, diferentemente do FedProx e do SCAFFOLD que modificam a arquitetura do cliente diretamente.

MOON (*MOdel-cOntrastive federated learNing*) [Li et al., 2021c] modifica a função de perda dos clientes ao utilizar três modelos para treinar: o modelo global, o modelo local e o modelo local da rodada anterior. Assim, os clientes aplicam o aprendizado construtivo, no qual a função de perda busca aproximar as representações geradas pelo modelo local atual às representações do modelo global, além de afastá-las das representações do modelo local da rodada anterior. Por fim, a perda construtiva é somada à perda supervisionada para atualizar os pesos do modelo.

No FedFTG (*Federated Fine-tuning Global model*) [Zhang et al., 2022c], antes de enviar o modelo agregado aos clientes, o algoritmo aplica a destilação de conhecimento usando os modelos dos clientes como professores do modelo global. O objetivo é au-

mentar o desempenho do modelo agregado, principalmente quando os clientes possuem distribuições de dados heterogêneas.

Entretanto, as propostas de novas estratégias de agregação assumem o aprendizado de um único modelo global, o que pode ser prejudicial para o sistema quando os clientes possuem dados e dispositivos altamente heterogêneos. Assim, a personalização de modelos é mais adequada, pois permite mitigar os problemas de heterogeneidade no modelo global ao mesmo tempo que produz um modelo local mais adequado aos dados e dispositivos dos clientes.

2.6.3. Personalização de modelos de inteligência artificial

O aprendizado federado tradicional, baseado no FedAvg [McMahan et al., 2017], otimiza um modelo global único que maximize o desempenho médio entre todos os participantes. No entanto, em redes veiculares, a premissa de que um único modelo é capaz de atender a todos os clientes é frequentemente invalidada pela heterogeneidade dos dados [Li et al., 2021a]. Veículos operando em diferentes regiões geográficas, sob condições climáticas variadas ou com perfis de condução distintos, geram distribuições estatísticas distintas. Nesse contexto, o Aprendizado Federado Personalizado (Personalized Federated Learning, PFL) surge como um paradigma que visa encontrar um equilíbrio ótimo entre o conhecimento global compartilhado e a especialização local necessária para cada veículo.

O FedDLD (*Federated Dual-Level Distillation*) [Xiao et al., 2025] é uma proposta que combina o aprendizado federado veicular com a destilação de conhecimento para reduzir os problemas causados pela heterogeneidade dos dados. Na proposta, os clientes recebem o modelo global e realizam a destilação localmente para personalizar o modelo de acordo com os dados locais. Além disso, os clientes são agrupados hierarquicamente de acordo com a similaridade dos modelos gerados no treinamento local. Essa etapa tem como objetivo realizar uma destilação mútua e adaptativa entre os clientes que possuem distribuições de dados similares. O servidor, por sua vez, os modelos locais e realiza a agregação. Para garantir a segurança do sistema, o FedDLD considera a reputação dos clientes durante a agregação e armazena as versões dos modelos globais em uma corrente de blocos em forma de grafo direcionado acíclico.

Embora promissores, métodos convencionais de personalização restringem-se a ajustes finos locais ou à aplicação de termos de regularização que tentam equilibrar a divergência entre os pesos locais e o modelo global. No entanto, no cenário veicular, tais métodos enfrentam o desafio crítico da divergência de pesos (*weight drift*), onde a otimização para distribuições locais heterogêneas pode prejudicar a estabilidade da agregação entre os nós. Nesse contexto, linhas de pesquisa recentes avançam em direção a promover adaptações arquiteturais e semânticas profundas, capazes de lidar de forma mais robusta com a complexidade dos dados e as restrições de rede.

2.6.3.1. Técnicas de agrupamento de clientes

O aprendizado federado com o agrupamento de clientes é uma subárea da personalização de modelos, cujo objetivo é mitigar o problema da heterogeneidade de dados ou

capacidades computacionais. Para isso, os clientes são separados em diferentes grupos de treinamento. Esses grupos podem ser independentes entre si ou compartilhar informações dos modelos treinados de acordo com cada proposta.

Estratégias baseadas em agrupamento (*clustering*) procuram reunir dispositivos com características semelhantes para otimizar a tomada de decisão. As métricas utilizadas para formar esses grupos geralmente envolvem os pesos do modelo local, o tempo de treinamento por época, a distribuição dos dados e a localização geográfica [Mayhoub e M. Shami, 2024].

O TiFL [Chai et al., 2023] é uma estratégia de seleção que consiste em traçar um perfil dos clientes de acordo com métricas de latência e classificá-los em *tiers*. Em seguida, clientes são selecionados aleatoriamente dentro de cada *tier*.

Os autores em [Bao et al., 2021] consideram um cenário onde veículos se comunicam entre si e com unidades de beira de estrada (*roadside units* – RSUs). Os autores propõem um método para identificar clientes de borda em potencial para realizar aprendizado federado. Esses clientes coletam dados de sensores de outros veículos não selecionados ao seu entorno e treinam uma época não apenas com os dados gerados pelo próprio veículo, mas também com os dados coletados de clientes menos capazes em seu entorno. Os clientes aptos para realizar o treinamento são selecionados a partir de um cálculo de competência utilizando lógica *fuzzy*, considerando a estabilidade de sua conexão, a topologia da rede e um fator de conectividade. Dessa forma, a proposta busca diminuir a sobrecarga de comunicação, selecionando os clientes mais aptos.

[de Souza et al., 2024a] propõem um sistema de aprendizado federado no qual os clientes são agrupados baseando-se na similaridade dos dados. Para manter a privacidade dos dados, os pesos dos modelos treinados a partir do mesmo estado inicial são utilizados como um parâmetro de similaridade. Cada grupo treina um único modelo por meio do FedAVG. Essa abordagem aumenta a acurácia do modelo e reduz o tempo de convergência. No entanto, a proposta é limitada a tarefas específicas, como classificar dados semelhantes aos dados de treinamento do grupo, dificultando a generalização para amostras de outras distribuições fora do grupo. Assim, o ATHENA-FL [de Souza et al., 2024b] aprimora o sistema, fornecendo uma maneira de combinar os modelos gerados em diferentes grupos por meio do modelo Um-Contra-Todos (*One-versus-All* – OvA). Este modelo é formado por um conjunto de modelos baseados em detectores binários para cada classe, permitindo adição de novos detectores uma vez que haja novas classes.

Entretanto, as divisões dos clientes em grupos podem ser complexas em cenários reais, nos quais os cliente não possuem apenas um subconjunto das classes. Assim, a alocação de um cliente em um grupo incorreto pode ser altamente prejudicial tanto para o grupo quanto para o cliente. Dessa forma, a personalização de camadas da rede resolve a sensibilidade da personalização ao algoritmo de agrupamento ao permitir a pernalização de camadas ao invés do modelo completo.

2.6.3.2. Personalização por divisão arquitetural

Uma das estratégias de personalização mais utilizadas consiste em dividir a arquitetura do modelo em componentes globais e locais. O FedPer [Arivazhagan et al., 2019] é um dos trabalhos pioneiros nessa linha, propondo que as camadas iniciais do modelo, responsáveis pela extração de características gerais, sejam treinadas de forma federada, enquanto as camadas finais, voltadas à personalização, permaneçam exclusivamente locais.

Expandindo essa ideia, o FedRep [Collins et al., 2021] propõe um esquema de treinamento alternado que desacopla o aprendizado da representação global da tarefa de classificação local. Nesse procedimento, o cliente inicialmente congela o extrator de características recebido do servidor e treina apenas o classificador local, de modo a alinhar as camadas finais às particularidades de suas predições. Em seguida, o classificador é congelado, e o treinamento local prossegue atualizando apenas os pesos do extrator de características. Ao final das épocas locais, somente os pesos atualizados do extrator são enviados para a agregação federada, enquanto os classificadores permanecem nos clientes. Esse processo em duas etapas favorece o aprendizado de um extrator de características mais genérico e robusto, ao mesmo tempo em que preserva a especialização dos classificadores para as características e classes de cada nó.

O FedBABU [Oh et al., 2022] simplifica essa dinâmica ao mostrar que a atualização contínua do classificador durante o treinamento federado pode, em alguns casos, prejudicar a qualidade das representações globais aprendidas. Diferentemente do FedRep, essa abordagem mantém a cabeça de classificação fixa ao longo de todas as rodadas de comunicação, concentrando a atualização e a agregação exclusivamente no extrator de características. A personalização ocorre apenas após a convergência do modelo global, por meio de um ajuste fino final do classificador com os dados locais de cada veículo, o que pode resultar em maior precisão em cenários com elevada heterogeneidade de rótulos.

Apesar da personalização de camadas aportar benefícios para o aprendizado federado, a definição das camadas que devem ser personalizadas é uma tarefa complexa. Assim, uma forma de contornar a escolha das camadas é realizar a personalização no objetivo local de otimização.

2.6.3.3. Personalização por modificação do objetivo de otimização

Outra linha de personalização em aprendizado federado baseia-se na modificação do objetivo de otimização, de forma a favorecer a adaptação local ou a equilibrar explicitamente os objetivos globais e locais.

O pFedMe [T Dinh et al., 2020], por sua vez, também desacopla a atualização do modelo global da otimização do modelo personalizado, mas o faz por meio de uma formulação de otimização bi-nível, na qual a especialização local e a evolução global são tratadas de forma interdependente. Nessa abordagem, o problema é organizado em dois níveis. No nível interno, cada cliente busca os parâmetros mais adequados para seu modelo personalizado por meio de múltiplas iterações locais, utilizando um termo de proximidade para evitar desvios excessivos em relação à referência global. No nível externo, o

servidor atualiza o modelo global não a partir de gradientes brutos dos dados locais, mas com base nos pesos já otimizados dos modelos personalizados. Essa formulação permite integrar de forma mais consistente a personalização local ao processo de evolução global do sistema.

O Per-FedAvg [Fallah et al., 2020], por exemplo, reformula o objetivo do aprendizado federado ao tratá-lo como um problema de preparação para adaptação. Em vez de buscar um modelo que seja apenas com um desempenho aceitável para todos os clientes simultaneamente, o método procura aprender um modelo global que funcione como um ponto de partida ideal para rápida adaptação local. Tecnicamente, isso é realizado em duas etapas em cada nó. Primeiro, executa-se uma atualização provisória para avaliar como o modelo se comportaria após um breve ajuste com os dados locais. Em seguida, os parâmetros originais são otimizados de modo que essa versão adaptada apresente o melhor desempenho possível. Assim, o objetivo deixa de ser a simples minimização da perda do modelo atual e passa a considerar a perda do modelo após um passo de gradiente voltado à adaptação. Como resultado, obtém-se um modelo global mais flexível, capaz de se transformar em uma solução especializada com baixo custo computacional, exigindo poucas iterações de ajuste fino para se adequar ao domínio de cada cliente.

Dentro desse mesmo paradigma, o Ditto [Li et al., 2021d] propõe decompor o problema de otimização em duas frentes complementares: uma global e outra local. A otimização global consolida a contribuição de todos os clientes para o treinamento de um modelo comum, seguindo o fluxo tradicional do FedAvg. Em paralelo, a otimização local ajusta o modelo individual de cada cliente com base em seus próprios dados, incorporando à função de perda um termo de regularização que penaliza a distância euclidiana entre o modelo personalizado do cliente e o modelo global mais recente. Esse mecanismo permite que o modelo local aprenda padrões específicos do cliente sem se afastar excessivamente do conhecimento global, reduzindo o risco de sobreajuste e mantendo os benefícios das representações compartilhadas entre os participantes.

Entretanto, a personalização local mitiga apenas o problema de desempenho dos modelos causados pela diferença estatística dos clientes. Assim, estratégias recentes de personalização propõem a combinação de técnicas como destilação de conhecimento e modelos generativos para permitir o treinamento de modelos heterogêneos nos clientes que compartilham o conhecimento global.

2.6.3.4. Estratégias recentes de personalização

Uma tendência recente para reduzir drasticamente a sobrecarga de comunicação em redes intermitentes é a personalização baseada em *prompt learning*. O Fed-Prompt [Zhao et al., 2023] introduz esse conceito ao manter o modelo base (*backbone*) completamente congelado durante o treinamento federado. Em vez de transmitir gradientes associados a milhões de parâmetros, os clientes otimizam e compartilham apenas pequenos vetores de contexto, denominados *prompts*, que adaptam o modelo global às tarefas locais. Expandindo essa ideia, o FedMGP (*Federated Learning with Multi-Group Text-Visual Prompts*) [Bo et al., 2025] propõe o uso de *prompts* multigrupo para lidar com a variação semântica em dados visuais. O principal diferencial dessa abordagem é sua ca-

pacidade de personalizar o reconhecimento de padrões de acordo com o contexto, como condições climáticas ou sinalizações regionais específicas, sem a necessidade de reentrenar camadas mais pesadas da rede neural, o que torna a adaptação mais leve e eficiente para unidades com recursos limitados.

O FedTP (*Federated Learning by Transformer Personalization*) [Li et al., 2023] busca lidar com a heterogeneidade dos clientes por meio da personalização de modelos baseados em *Transformers*. O objetivo central da proposta é aprender uma base de transformação capaz de alinhar as representações dos dados locais de diferentes clientes, permitindo que cada nó personalize camadas específicas do modelo, como os mecanismos de atenção, para mitigar problemas de convergência. Tecnicamente, a abordagem emprega parâmetros de projeção aprendidos localmente para transformar o espaço de características, com o intuito de tornar mais semelhantes os dados provenientes de clientes distintos antes da agregação. No entanto, essa estratégia impõe uma sobrecarga de comunicação significativa, além de exigir tempo elevado para o ajuste dos parâmetros de projeção e para a convergência do modelo, o que limita sua aplicabilidade em cenários veiculares de alta mobilidade, nos quais as janelas de conexão tendem a ser curtas.

Em um nível mais estrutural de adaptação, a Mistura de Especialistas Personalizada (*Personalized Federated Learning with Mixture of Experts – PFL-MoE*) surge como uma alternativa promissora para lidar simultaneamente com a heterogeneidade de hardware e com a diversidade dos domínios de condução. O FedMoEKD (*Mixture of Specialized Experts for Model-Heterogeneous Personalized Federated Learning*) [Liang et al., 2025] propõe uma arquitetura esparsa na qual apenas uma fração dos parâmetros é ativada para cada entrada. Por meio de um mecanismo de *gating* aprendido localmente, cada cliente identifica quais submodelos são mais adequados ao seu cenário, processando apenas os componentes necessários. De forma complementar, o pFedMoE [Yi et al., 2026] introduz uma distinção entre especialistas globais e privados. Enquanto os especialistas globais consolidam o conhecimento compartilhado pela frota, os especialistas privados permanecem no cliente para aprender comportamentos específicos e padrões sensoriais mais sensíveis, promovendo uma personalização mais profunda sem comprometer a privacidade dos dados locais.

Em conjunto, essas abordagens mostram que a personalização em aprendizado federado pode ser realizada em diferentes níveis, desde ajustes na arquitetura e no objetivo de otimização até mecanismos mais recentes, como *prompts*, *Transformers* personalizados e mistura de especialistas (MoE). Em redes veiculares, essa pluralidade de soluções é essencial para acomodar a diversidade de contextos de operação da frota, incluindo diferenças de ambiente, comportamento de condução, condições climáticas, disponibilidade de conectividade e capacidade computacional dos veículos. Dessa forma, a personalização deixa de ser apenas uma alternativa para melhorar a acurácia local e passa a representar um componente central para viabilizar sistemas federados mais robustos, adaptativos e eficientes nesse domínio.

O FedCG (*Federated learning method that leverages Conditional Generative adversarial networks*) [Wu et al., 2021b] decompõe os modelos dos clientes em um extrator de características e um classificador, ao mesmo tempo que introduz uma CGAN (*Conditional Generative Adversarial Network*) dividida em um gerador e um discriminador. A

proposta compartilha apenas o classificador e o gerador com o servidor, que são agregados por meio de KD em um modelo global, compartilhado com os clientes. O objetivo é manter o extrator de características, constituído pelas camadas iniciais da rede neural do cliente, privado para evitar vazamento de dados durante o treinamento do modelo global e ao mesmo tempo reduzir o efeito de dados negativo da heterogeneidade de dados.

O CompFL (**C**ooperative multiple **m**odel training framework for **p**ersonalized **F**ederated **L**earning) [Xu et al., 2024a] executa aprendizado federado com modelos heterogêneos. Em vez de treinar apenas um único modelo, os clientes com maior poder computacional também têm a probabilidade de selecionar um modelo mais simples para ajudar os clientes com menor poder computacional durante as rodadas de treinamento. Além disso, os clientes executam KD para aprender com diferentes modelos em suas amostras locais. No entanto, a proposta sobrecarrega os clientes com maior poder computacional, o que implica um alto custo computacional para contribuir com o sistema.

2.6.4. Utilização de múltiplos modelos de inteligência artificial

Além das estratégias de personalização, que visam reduzir os efeitos da heterogeneidade de dados, recentemente, existe a tendência de utilizar múltiplos modelos para mitigar a heterogeneidade de dispositivos. A fim de criar um modelo único, técnicas de agregação que utilizam a destilação do conhecimento têm sido aplicadas.

O FedGen (*Federated Distillation via Generative Learning*) [Zhu et al., 2021b] aborda o problema da distribuição heterogênea de dados. Os clientes compartilham a distribuição de rótulos com o servidor para treinar o gerador. Em seguida, o gerador é compartilhado com os clientes, para que possam executar a destilação de conhecimento.

FedAdKD (*Federated Adaptive Knowledge Distillation*) [Song et al., 2024] aplica a destilação de conhecimento para reduzir os efeitos da heterogeneidade das distribuições de dados dos clientes. Inicialmente, os clientes executam uma fase de treinamento federado com o FedAvg de um modelo de difusão (*diffusion model*), utilizado para gerar dados sintéticos. Dessa forma, o servidor pode ter acesso ao gerador que cria amostras sintéticas que seguem as distribuições sem revelar dados locais dos clientes. Após essa etapa, o servidor inicia o treinamento do modelo para a tarefa de aprendizado principal em conjunto com os clientes. A diferença dessa estratégia para o treinamento com o FedAvg é que o servidor, após agregar os modelos locais em um novo modelo global, realiza uma destilação de conhecimento com múltiplos professores (*Multi-Teacher Knowledge Distillation* – MTKD). Nesse caso, o modelo global após a agregação é utilizado como estudante dos modelos locais recebidos, que atuam como professores, usando o conjunto de dados sintético. A proposta mantém o uso de arquiteturas idênticas para realizar a agregação antes da destilação do conhecimento.

FedZIO (*Federated Distillation with Zonal Interaction Optimizer*) [Palazzo et al., 2023] é uma proposta de aprendizado federado descentralizado que utiliza a destilação de conhecimento como uma forma de treinamento dos modelos. Os autores assumem que existem no sistema clientes heterogêneos, que treinam um modelo adequado às suas capacidades computacionais. Por meio de um protocolo distribuído, os clientes recebem a predição (*logits*) de outros clientes para realizar a destilação de conhecimento. Entretanto, a proposta possui um desempenho limitado em

comparação com outras técnicas do estado da arte.

O HeteroFL [Diao et al., 2020] é uma das primeiras estratégias de FL que permite aos clientes treinar com modelos heterogêneos. A proposta divide o modelo global em submodelos e os clientes treinam o submodelo mais adequado à sua capacidade computacional. Como os submodelos estão relacionados ao modelo global, o esquema de agregação dispensa a destilação de conhecimento e o uso de dados públicos.

O FedDKC (*Federated Distillation algorithm based on Distributed Knowledge Congruence*) [Wu et al., 2024b] é um algoritmo que utiliza o KD para mitigar o problema decorrente da heterogeneidade no canal de comunicação e na distribuição de clientes no aprendizado federado. Os autores propõem uma estratégia para refinar o KD e reduzir a discrepância entre os clientes participantes do treinamento federado. Apesar da abordagem dispensar o uso de dados dos clientes para realizar o KD, os clientes devem realizar o KD. Assim, essa abordagem adiciona sobrecarga no lado do cliente.

O Felo [Chan e Ngai, 2025] difere de outras propostas FL-KD por utilizar características de nível intermediário, além dos *logits*, para transferir conhecimento de clientes com modelos heterogêneos. As características de nível intermediário são os resultados das camadas intermediárias do modelo. Uma vez gerados no lado do cliente, os *logits* e as características de nível intermediário são calculados localmente por classe e, em seguida, enviados juntamente com os parâmetros do modelo. O servidor agrega modelos com a mesma arquitetura de rede neural e, em cada classe, calcula a média das informações enviadas pelos clientes. Os autores estendem a primeira proposta com o Velo, que aprimora o processo ao adicionar um gerador para produzir dados sintéticos no lado do servidor, utilizando características de nível intermediário. No entanto, o lado do cliente apresenta uma sobrecarga em comparação com a proposta FedAvg, uma vez que os clientes precisam gerar *logits* e características de nível intermediário ao executar o cálculo da média desses dados.

O FedBKD (*Federated Bidirectional Knowledge Distillation*) [Qi et al., 2022] é um arcabouço de FL para lidar com a heterogeneidade de dados e de modelos em sistemas de borda em dispositivos IoT. Em uma fase de aquecimento, os clientes treinam um Autoencoder Variacional Condicional (*Conditional Variational Autoencoder – CVAE*) para gerar dados sintéticos no servidor. Durante o treinamento, o FedBKD utiliza a destilação de conhecimento em duas fases. Na fase de enlace ascendente (*uplink*), o servidor recebe os modelos dos clientes e executa um MTKD para treinar o modelo global. Além disso, na fase de enlace descendente (*downlink*), o modelo global obtido na fase anterior é usado localmente pelo cliente como professor para atualizar o modelo local. Apesar de a proposta ser projetada para permitir que os clientes treinem com diferentes arquiteturas de modelo, os experimentos testam a eficácia apenas com clientes compartilhando modelos similares, como o mesmo modelo base modificando apenas o número de camadas. Além disso, o FedBKD implica uma sobrecarga para o cliente, uma vez que na fase de enlace descendente a destilação deve ocorrer no dispositivo do cliente.

O FedAgg (*Agglomerative Federated Learning*) [Wu et al., 2024a] é uma proposta que aplica a destilação de conhecimento em uma arquitetura hierárquica fim-borda-nuvem para aprender um modelo conjunto em cada nível. Dispositivos de nuvem e de borda possuem mais poder computacional do que os nós finais. Portanto, aplicar um modelo homo-

gêneo a todos os nós subutiliza os recursos dos nós mais poderosos e reduz a acurácia do modelo global. Assim, os autores propõem o uso de modelos diferentes, de acordo com o poder de processamento, para melhor aproveitar os recursos do sistema. Como apenas os clientes possuem dados, o FedAgg compartilha amostras de ponte (bridge samples), que são amostras transformadas para preservar a privacidade dos usuários, para executar a destilação de conhecimento e treinar modelos nos nós de borda e de nuvem. No entanto, o FedAgg apresenta desvantagens para o cenário veicular, pois não é adaptado a um ambiente dinâmico. Por exemplo, a proposta pressupõe que os nós seguem uma distribuição fixa, de modo que o sistema sempre possui um único nó de nuvem e que os clientes são mais numerosos do que os nós de borda. Além disso, os autores assumem que os clientes são homogêneos e a heterogeneidade computacional do sistema encontra-se nos níveis hierárquicos da arquitetura, os quais não possuem amostras de dados.

O *Data-free One-Shot federated Learning* (DENSE) [Zhang et al., 2022a] treina modelos heterogêneos com apenas uma rodada de comunicação. Na proposta, os clientes treinam seus modelos, enquanto o servidor primeiramente treina um gerador com base nos modelos recebidos, garantindo similaridade, transferibilidade e estabilidade de forma adversarial. Em seguida, o servidor utiliza a destilação de conhecimento com os dados gerados para obter um modelo global. No entanto, o DENSE não considera as mudanças na distribuição de dados (*data shifts*) ao modelar o problema, o que leva o modelo global a esquecer catastróficamente conhecimentos úteis aprendidos em rodadas anteriores e a sofrer com a degradação de desempenho. O DFRD (Data-Free Robust Distillation) [Luo et al., 2023] é um método para Aprendizado Federado (FL) heterogêneo baseado em um gerador condicional treinado de forma adversarial. O sistema recebe os modelos dos clientes e os utiliza para treinar o gerador a fim de produzir amostras, garantindo, ao mesmo tempo, a fidelidade, a transferibilidade e a diversidade dos dados. Para evitar o esquecimento catastrófico, o DFRD utiliza uma cópia de média móvel exponencial (EMA) do gerador para armazenar o conhecimento prévio dos modelos locais. O gerador é então usado para criar dados sintéticos a cada rodada para realizar a destilação de conhecimento e melhorar o desempenho do modelo global. Além disso, o DFRD lida com a heterogeneidade de modelos empregando um método de treinamento parcial (PT) [Yang et al., 2022], no qual os clientes treinam submodelos de larguras diferentes extraídos do modelo global de acordo com seus orçamentos de recursos, o que significa que apenas um subconjunto dos pesos do modelo é atualizado localmente.

As estratégias de personalização permitem um treinamento mais especializado do modelo, aumentando o desempenho local. No entanto, essas estratégias são ortogonais às necessidades de aplicações federada veiculares, como a existência de dados rotulados. Assim, surgem novos paradigmas de aprendizado que buscam resolver problemas mais amplos, relacionados às aplicações.

2.6.5. Paradigma de aprendizado

Um dos principais desafios do aprendizado veicular é a obtenção de dados rotulados em aplicações como a visão computacional supervisionada. Isso ocorre devido ao cenário crítico, onde o usuário deve manter atenção constante ao trajeto e dificilmente pode contribuir com a rotulação ativa de dados quando o veículo está parado. Dessa forma, surgem novos paradigmas de aprendizado a fim de contornar esses desafios.

Os autores em FedMatch [Jeong et al., 2021] propõem um novo método de aprendizado semissupervisionado com troca de informação entre clientes de aprendizado federado. O FedMatch ataca o problema da falta de rótulos nos clientes de aprendizado federado. O aprendizado semissupervisionado em FedMatch utiliza o conceito de regularização de consistência, onde o modelo é treinado para prever o mesmo rótulo de classe para uma amostra original e a mesma amostra com aumentações. No entanto, o FedMatch amplia esse conceito para o cenário federado, introduzindo uma nova função de perda de consistência intercliente, onde modelos auxiliares são enviados pelo servidor baseados na similaridade do modelo local. Ademais, o FedMatch decompõe os parâmetros do modelo em dois grupos, um para o aprendizado semissupervisionado e um para o supervisionado. Durante o treinamento, a otimização desses parâmetros ocorre de forma disjunta, ao aprender com dados rotulados, o modelo atualiza os parâmetros supervisionados enquanto congela os não supervisionados, enquanto ao aprender com dados não rotulados, o modelo atualiza os parâmetros não supervisionados mantendo os supervisionados congelados.

O RSCFed [Liang et al., 2022] considera dois cenários de aprendizado federado semissupervisionado. O primeiro consiste em clientes com dados totalmente rotulados sendo treinados em conjunto com clientes com dados não rotulados. O segundo consiste em clientes com dados parcialmente rotulados sendo treinados em conjunto. Assumindo um cenário não-IID, o RSCFed consiste em considerar os modelos locais como modelos ruidosos e extrair diversos modelos de consenso por meio de amostragem aleatória antes de agregá-los ao modelo global. O aprendizado federado com o RSCFed consiste em (1) amostrar aleatoriamente clientes locais; (2) atribuir o modelo global atual aos clientes selecionados como inicialização e realizar o treino local nesses clientes; (3) realizar agregação ponderada pela distância nos modelos treinados; (4) repetir as etapas anteriores para criar um conjunto de modelos de subconsenso; (5) agregar um novo modelo a partir desse conjunto de subconsensos para se tornar o próximo modelo global.

O FedSTO [Kim et al., 2023] é proposto como uma solução para o caso particularmente desafiador de treinamento federado semissupervisionado em que apenas o servidor apresenta dados rotulados (*labels-at-server*), enquanto os clientes dispõem exclusivamente de dados não rotulados e possivelmente não-IID. O método é estruturado em dois estágios. No primeiro estágio, denominado treinamento seletivo, o modelo é treinado inicialmente com os dados do servidor. Em seguida, o *backbone* do modelo é ajustado aos dados não rotulados dos clientes por meio de uma abordagem de minimização de entropia baseada na comparação entre pseudo-rótulos de uma rede aluna e outra rede professora, que tem seus pesos atualizados por meio de uma média móvel exponencial (*Exponential Moving Average - EMA*) dos pesos da rede aluna. Com o fim de garantir maior robustez ao problema de viés de características, discutido na subseção 2.4.2, os autores incorporam uma técnica de regularização por ortogonalidade que promove maior adaptabilidade do modelo a dados provenientes de domínios distintos.

Os autores em [Li et al., 2022b] atacam o problema de alocação de recursos em redes veiculares onde enlaces V2V devem reutilizar os canais celulares já alocados, decidindo de forma autônoma e descentralizada quais canais reutilizar e qual potência de transmissão adotar, sem interromper as operações celulares e evitando colisões com outros enlaces V2V. Os autores propõem uma solução utilizando aprendizado por reforço

profundo multi-agente federado. Cada par V2V treina um modelo *Dueling Double Deep Q Network* de forma federada para selecionar qual canal celular deve ser utilizado e selecionar o nível de potência de transmissão. A função de recompensa formulada tenta otimizar a taxa de transmissão total de todos os usuários celulares enquanto mantém os requisitos de *Quality of Service* dos usuários celulares e pares V2V. Os clientes recebem um modelo global, coletam observações locais e treinam localmente. Os parâmetros dos modelos locais são enviados ao servidor e um novo modelo local é obtido através de agregação com o FedAvg.

2.7. Considerações Finais

Este minicurso apresentou o aprendizado federado veicular e discutiu as principais características e desafios que o diferenciam do FL tradicional. As arquiteturas abordadas demonstram diversas possibilidades de implementação do VFL segundo as tecnologias de comunicação presentes no veículo. No VFL é comum utilizar estratégias de agregação hierárquicas. Dessa forma, as RSU podem ser utilizadas como pontos de agregação entre o cliente e o servidor de nuvem, reduzindo a quantidade de dados transmitidos no núcleo da rede e oferecendo maior desempenho para os dispositivos próximos. Outra possibilidade é a realização de treinamento par a par, na qual os clientes atuam como cliente e servidor de agregação ao mesmo tempo. Porém, os problemas de generalização do modelo global são agravados em cenários hierárquicos e distribuídos. Assim, a principal estratégia de agregação utilizada é a centralizada quando possível. Como o treinamento do FL é realizado em múltiplos dispositivos, este está sujeito aos problemas enfrentados por sistemas distribuídos. Dessa forma, é necessário definir estratégias de coordenação para determinar o momento no qual o modelo é agregado e em que os clientes são sincronizados com o modelo global. A estratégia síncrona é estrita com relação ao tempo, descartando clientes muito lentos quando ultrapassam o tempo limite de resposta. Ao mesmo tempo, o sincronismo impõe tempo ocioso aos dispositivos mais rápidos. Por outro lado, o assíncrono elimina o descarte de modelos, agregando sob demanda a cada vez que um cliente envia suas atualizações. Entretanto, a atualização frequente impõe maior discordância entre modelos treinados por clientes retardatários e clientes rápidos, chamada de obsolescência. Por fim, a estratégia semissíncrona utiliza elementos síncronos, como a definição de tempo limite e descarte de modelos muito antigos ao mesmo tempo que permite a contribuição de clientes retardatários em rodadas diferentes para reduzir o descarte de modelos. Entretanto, o SGD, o principal algoritmo de atualização dos parâmetros, é sensível ao estado do modelo. Logo, versões diferentes prejudicam a agregação, reduzindo o desempenho de soluções não síncronas.

Limitações de latência, cobertura e vazão na rede de acesso podem acarretar no descarte de modelos e atrasos no término das rodadas, introduzindo requisitos de tolerância a falhas no projeto das máquinas de estado do VFL. As camadas de protocolos devem ser selecionadas conforme as necessidades de sincronismo entre cliente e servidor, sobrecarga de controle e confiabilidade do canal de comunicação. Ferramentas de gerenciamento de contêineres, transferência de arquivos e controle de versão de *software* são combinadas para construir arquiteturas de VFL que incluem não só a troca de parâmetros do modelo, como também a troca de mensagens de controle entre cliente e servidor.

A seção prática demonstrou como modelar as características da arquitetura e in-

fraestrutura apresentada, permitindo aos participantes o entendimento da implementação do VFL. Inicialmente discutiu-se como modelar a heterogeneidade dos dados dos clientes e seus recursos computacionais. Além disso, foi implementado um modelo de mobilidade que permite a análise do ambiente veicular. Por fim, demonstrou-se como agregar esses elementos no arcabouço Flower para a simulação fim a fim do aprendizado federado veicular. Após a atividade prática, foram apresentadas as principais tendências de pesquisas atuais, que utilizam arcabouços, como o Flower, para o desenvolvimento e teste de técnicas inovadoras.

As tendências de pesquisa identificadas indicam que o aprendizado federado em redes veiculares evolui para abordagens cada vez mais adaptadas à heterogeneidade dos dados, dos dispositivos e das condições de comunicação, afastando-se de estratégias genéricas baseadas em seleção aleatória de clientes e em um único modelo global. Nesse contexto, destacam-se como direções principais a seleção inteligente de clientes, as novas estratégias de agregação, a personalização de modelos, o agrupamento de clientes e o uso de múltiplos modelos, com o objetivo de reduzir o tempo de convergência, aumentar a acurácia e tornar o treinamento mais robusto em cenários dinâmicos e não-IID. Além disso, observa-se um avanço de técnicas baseadas em destilação de conhecimento e de novos paradigmas de aprendizado voltados a restrições práticas das aplicações, como assimetria de recursos, mudanças de distribuição e escassez de rótulos. Em conjunto, essas tendências mostram que a viabilidade do aprendizado federado veicular depende, cada vez mais, de soluções que conciliem adaptação ao contexto, eficiência de comunicação e robustez operacional.

Apesar dos avanços recentes, ainda há desafios a serem superados para a adoção efetiva do aprendizado federado em redes veiculares. Sobretudo, devido à dificuldade de lidar simultaneamente com a alta mobilidade do domínio veicular, heterogeneidade estatística dos dados, assimetria de recursos computacionais e instabilidade de comunicação. Além disso, muitas propostas ainda dependem de pressupostos simplificados, como conectividade consistente, custo computacional adicional nos clientes ou distribuições de dados relativamente estáveis, limitando sua aplicabilidade em cenários veiculares reais.

Nesse sentido, surge como direção futura a oportunidade de desenvolver métodos mais contextuais e realistas, capazes de integrar mecanismos de seleção de clientes, estratégias de agregação e personalização mais leves e robustas e técnicas de agrupamento dinâmico sensíveis às mudanças do ambiente e abordagens com múltiplos modelos que preservem a eficiência do sistema. Além disso, pesquisas futuras podem explorar com maior profundidade cenários realistas de implantação, incluindo aprendizado contínuo diante de mudanças de distribuição, escassez de rótulos, bem como mecanismos de segurança, confiança e privacidade compatíveis com a alta dinamicidade das redes veiculares. Assim, o avanço da área dependerá da proposição de soluções que não apenas melhorem o desempenho dos modelos, mas que também sejam efetivamente implementáveis em ambientes veiculares reais, dinâmicos e de larga escala.

Agradecimentos

Este capítulo foi realizado com apoio do CNPq (310234/2025-5, 407304/2025-8, 402531/2025-6, 408255/2023-4, 405940/2022-0 e 309304/2021-0); da Coordena-

ção de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001, 88887.954253/2024-00 e 88887.987121/2024-00; da FAPERJ (E-26/200.380/2023, E-26/204.122/2024 e E-26/210.778/2025); da FAPESP (2023/00673-7 e 2023/00811-0) e da Fundação de Desenvolvimento da Pesquisa - Fundep - Rota 2030 em conjunto dos nossos parceiros Stellantis e Mobway.

Referências

- [Abad et al., 2020] Abad, M. S. H., Ozfatura, E., Gunduz, D. e Ercetin, O. (2020). Hierarchical Federated Learning across Heterogeneous Cellular Networks. Em *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 8866–8870. IEEE.
- [Ali et al., 2025] Ali, A., Jianjun, H. e Jabbar, A. (2025). Recent Advances in Federated Learning for Connected Autonomous Vehicles: Addressing Privacy, Performance, and Scalability Challenges. *IEEE Access*.
- [Amadeo et al., 2025] Amadeo, M., Campolo, C., Ruggeri, G. e Molinaro, A. (2025). Improving Communication Performance of Federated Learning: A Networking Perspective. *Computer Networks*, 267:111353.
- [Arivazhagan et al., 2019] Arivazhagan, M. G., Aggarwal, V., Singh, A. K. e Choudhary, S. (2019). Federated Learning with Personalization Layers. *arXiv:1912.00818*.
- [Azimi-Abarghouyi e Fischione, 2025] Azimi-Abarghouyi, S. M. e Fischione, C. (2025). Multi-Layer Hierarchical Federated Learning with Quantization. *arXiv:2505.08145*.
- [Bao et al., 2021] Bao, W., Wu, C., Guleng, S., Zhang, J., Yau, K.-L. A. e Ji, Y. (2021). Edge Computing-Based Joint Client Selection and Networking Scheme for Federated Learning in Vehicular IoT. *China Communications*, 18(6):39–52.
- [Beltrán et al., 2023] Beltrán, E. T. M. et al. (2023). Decentralized Federated Learning: Fundamentals, State of the Art, Frameworks, Trends, and Challenges. *IEEE Communications Surveys & Tutorials*, 25(4):2983–3013.
- [Beutel et al., 2020] Beutel, D. J. et al. (2020). Flower: A Friendly Federated Learning Research Framework. *arXiv preprint arXiv:2007.14390*.
- [Bo et al., 2025] Bo, W., Sun, Y., Wang, Y., Zhang, X. e Li, Z. (2025). FedMGP: Personalized Federated Learning with Multi-Group Text-Visual Prompts. Em *Conference on Neural Information Processing Systems*.
- [Buyukates e Ulukus, 2021] Buyukates, B. e Ulukus, S. (2021). Timely Communication in Federated Learning. Em *International Conference on Computer Communications Workshops (INFOCOM WKSHPs)*, p. 1–6. IEEE.
- [Chahoud et al., 2023a] Chahoud, M., Otoum, S. e Mourad, A. (2023a). On the Feasibility of Federated Learning Towards On-demand Client Deployment at the Edge. *Information Processing & Management*, 60(1):103150.

- [Chahoud et al., 2023b] Chahoud, M., Sami, H., Mourad, A., Otoum, S., Otrok, H., Bentahar, J. e Guizani, M. (2023b). On-demand-FL: A Dynamic and Efficient Multicriteria Federated Learning Client Deployment Scheme. *IEEE Internet of Things Journal*, 10(18):15822–15834.
- [Chahoud et al., 2025] Chahoud, M., Sami, H., Mourad, A., Otrok, H., Bentahar, J. e Guizani, M. (2025). On-demand model and client deployment in federated learning with deep reinforcement learning. *IEEE Internet of Things Journal*.
- [Chai et al., 2023] Chai, Z. et al. (2023). TiFL: A Tier-based Federated Learning System. Em *International Symposium on High-Performance Parallel and Distributed Computing*, HPDC '20, p. 125–136. Association for Computing Machinery.
- [Chan e Ngai, 2025] Chan, Y.-H. e Ngai, E. C. (2025). Exploiting Features and Logits in Heterogeneous Federated Learning. *Computer Networks*, 264:111271.
- [Chatzoulis et al., 2023] Chatzoulis, D. et al. (2023). 5G V2X Performance Comparison for Different Channel Coding Schemes and Propagation Models. *Sensors*, 23(5):2436.
- [Chellapandi et al., 2023] Chellapandi, V. P., Yuan, L., Brinton, C. G., Žak, S. H. e Wang, Z. (2023). Federated Learning for Connected and Automated Vehicles: A Survey of Existing Approaches and Challenges. *IEEE Transactions on Intelligent Vehicles*, 9(1):119–137.
- [Chen et al., 2023a] Chen, M., Xu, Y., Xu, H. e Huang, L. (2023a). Enhancing Decentralized Federated Learning for Non-IID Data on Heterogeneous Devices. Em *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, p. 2289–2302. IEEE.
- [Chen et al., 2024] Chen, Z., Ni, Z., Guan, P., Wang, L., Cai, L. X., Hashemi, M. e Li, Z. (2024). Optimizing NOMA Transmissions to Advance Federated Learning in Vehicular Networks. Em *Global Communications Conference (GLOBECOM)*, p. 962–967.
- [Chen et al., 2023b] Chen, Z., Shi, L., Liu, X., Ai, X., Liu, S. e Xu, Y. (2023b). Boosting Distributed Machine Learning Training Through Loss-tolerant Transmission Protocol. Em *2023 IEEE/ACM 31st International Symposium on Quality of Service (IWQoS)*, p. 1–10. ISSN: 2766-8568.
- [Cleland et al., 2022] Cleland, G., Wu, D., Ullah, R. e Varghese, B. (2022). FedComm: Understanding Communication Protocols for Edge-based Federated Learning. Em *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*, p. 71–81, Vancouver, WA, USA. IEEE.
- [Collins et al., 2021] Collins, L., Hassani, H., Mokhtari, A. e Shakkottai, S. (2021). Exploiting Shared Representations for Personalized Federated Learning. Em *International Conference on Machine Learning*, p. 2089–2099. PMLR.
- [De Rango et al., 2021] De Rango, F., Guerrieri, A., Raimondo, P. e Spezzano, G. (2021). A Novel Edge-based Multi-Layer Hierarchical Architecture for Federated Learning. Em *DASC/PiCom/CBDCoM/CyberSciTech*, p. 221–225. IEEE.

- [de Souza et al., 2024a] de Souza, L. A. C., Camilo, G. F., Campista, M. E. M. e Costa, L. H. M. (2024a). Increasing the Accuracy of Federated Learning on Non-IID Scenarios using Client Clustering. Em *Latin-American Conference on Communications (LATINCOM)*, p. 1–6. IEEE.
- [de Souza et al., 2024b] de Souza, L. A. C., Camilo, G. F., Rebello, G. A. F., Sammarco, M., Campista, M. E. M. e Costa, L. H. M. K. (2024b). ATHENA-FL: Avoiding Statistical Heterogeneity with One-versus-All in Federated Learning. *Journal of Internet Services and Applications*, 15(1):273–288.
- [de Souza et al., 2026] de Souza, L. A. C. et al. (2026). CAIROS: Controle Adaptativo do Aprendizado Federado em Redes Sem Fio. Em *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBC.
- [de Souza et al., 2025] de Souza, L. A. C., Sammarco, M., Achir, N., Costa, L. H. M. K. e Campista, M. E. M. (2025). TOFL: Time Optimized Federated Learning. Em *SBSeg 2025-Simpósio Brasileiro de Cibersegurança*.
- [Diao et al., 2020] Diao, E., Ding, J. e Tarokh, V. (2020). HeteroFL: Computation and Communication Efficient Federated Learning for Heterogeneous Clients. *arXiv preprint arXiv:2010.01264*.
- [EletronicsMaker, 2024] EletronicsMaker (2024). Driving the Future: The Role of Computing Power in Autonomous Vehicles. <https://electronicsmaker.com/driving-the-future-the-role-of-computing-power-in-autonomous-vehicles>.
- [Fallah et al., 2020] Fallah, A., Mokhtari, A. e Ozdaglar, A. (2020). Personalized Federated Learning: A Meta-Learning Approach. *arXiv preprint arXiv:2002.07948*.
- [Ferguson, 1973] Ferguson, T. S. (1973). A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics*, p. 209–230.
- [Fittipaldi et al., 2025] Fittipaldi, G., Couto, R. S. e Costa, L. H. (2025). Exploring Traffic Pattern Variability in Vehicular Federated Learning. *Computer Communications*, p. 108279.
- [Fu et al., 2023] Fu, L., Zhang, H., Gao, G., Zhang, M. e Liu, X. (2023). Client Selection in Federated Learning: Principles, Challenges, and Opportunities. *IEEE Internet of Things Journal*, 10(24):21811–21819.
- [Galende et al., 2024] Galende, B. A. et al. (2024). FLIP: A New Approach for Easing the Use of Federated Learning. *Applied Sciences*.
- [Ghanem et al., 2022] Ghanem, M., Dawoud, F., Gamal, H., Soliman, E., El-Batt, T. e Sharara, H. (2022). FLoBC: A Decentralized Blockchain-Based Federated Learning Framework. Em *International Conference on Blockchain Computing and Applications (BCCA)*, p. 85–92. IEEE.

- [Han et al., 2024] Han, D.-J., Hosseinalipour, S., Love, D. J., Chiang, M. e Brinton, C. G. (2024). Cooperative Federated Learning Over Ground-to-Satellite Integrated Networks: Joint Local Computation and Data Offloading. *IEEE Journal on Selected Areas in Communications*, 42(5):1080–1096.
- [He et al., 2020] He, C., Li, S., So, J., Zeng, X., Zhang, M., Wang, H., Wang, X., Vepakomma, P., Singh, A., Qiu, H., Zhu, X., Wang, J., Shen, L., Zhao, P., Kang, Y., Liu, Y., Raskar, R., Yang, Q., Annavaram, M. e Avestimehr, S. (2020). FedML: A Research Library and Benchmark for Federated Machine Learning. arXiv:2007.13518 [cs].
- [Heydarishirayeh, 2023] Heydarishirayeh, N. (2023). *FedMingle: Communication in Federated Learning*. PhD thesis, Politecnico di Milano.
- [Huang et al., 2021] Huang, Y. et al. (2021). Evaluating Gradient Inversion Attacks and Defenses in Federated Learning. Em Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P. e Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, p. 7232–7241. Curran Associates, Inc.
- [Huedo et al., 2025] Huedo, E. et al. (2025). On-Demand Deployment of Edge Cloud Infrastructures for Federated Learning. *Software: Practice and Experience*, 55(8):1377–1388.
- [Islam et al., 2025] Islam, M. S., Panta, S., Xu, F., Yuan, X., Chen, L. e Tzeng, N.-F. (2025). SEAF: Enhancing Efficiency in Semi-Asynchronous Federated Learning through Adaptive Aggregation and Selective Training. Em *International Parallel and Distributed Processing Symposium (IPDPS)*, p. 509–519. IEEE.
- [Jeong e Kountouris, 2023] Jeong, E. e Kountouris, M. (2023). Personalized Decentralized Federated Learning with Knowledge Distillation. Em *IEEE International Conference on Communications*, p. 1982–1987.
- [Jeong et al., 2021] Jeong, W., Yoon, J., Yang, E. e Hwang, S. J. (2021). Federated Semi-Supervised Learning with Inter-Client Consistency & Disjoint Learning.
- [Ji et al., 2024] Ji, S., Tan, Y., Saravirta, T., Yang, Z., Liu, Y., Vasankari, L., Pan, S., Long, G. e Walid, A. (2024). Emerging Trends in Federated Learning: from Model Fusion to Federated X Learning. *International Journal of Machine Learning and Cybernetics*, 15(9):3769–3790.
- [Kairouz et al., 2021] Kairouz, P. et al. (2021). Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.*, 14(1–2):1–210.
- [Kalra et al., 2023] Kalra, S., Wen, J., Cresswell, J. C., Volkovs, M. e Tizhoosh, H. R. (2023). Decentralized Federated Learning through Proxy Model Sharing. *Nature communications*, 14(1):2899.
- [Kang et al., 2025] Kang, N., Lim, Y. e Im, J. (2025). Optimizing Federated Learning: Addressing Key Challenges in Real-World Applications. *Internet of Things Journal*.

- [Karimireddy et al., 2020] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S. e Suresh, A. T. (2020). SCAFFOLD: Stochastic Controlled Averaging for Federated Learning. Em *International conference on machine learning*, p. 5132–5143. PMLR.
- [Kim et al., 2023] Kim, T., Lin, E., Lee, J., Lau, C. e Mugunthan, V. (2023). Navigating Data Heterogeneity in Federated Learning: A Semi-Supervised Federated Object Detection. Em Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M. e Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, p. 2074–2096. Curran Associates, Inc.
- [Kornaros et al., 2020] Kornaros, G., Tomoutzoglou, O., Mbakoyiannis, D., Karadimitriou, N., Coppola, M., Montanari, E., Deligiannis, I. e Gherardi, G. (2020). Towards Holistic Secure Networking in Connected Vehicles through Securing CAN-Bus Communication and Firmware-Over-The-Air Updating. *Journal of Systems Architecture*, 109:101761.
- [Lai et al., 2021] Lai, F., Zhu, X., Madhyastha, H. V. e Chowdhury, M. (2021). Oort: Efficient Federated Learning via Guided Participant Selection. Em *USENIX OSDI*, p. 19–35.
- [Li et al., 2022a] Li, B., Jiang, Y., Pei, Q., Li, T., Liu, L. e Lu, R. (2022a). FEEL: Federated End-to-End Learning With Non-IID Data for Vehicular Ad Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):16728–16740.
- [Li et al., 2021a] Li, B., Jiang, Y., Sun, W., Niu, W. e Wang, P. (2021a). FedVANET: Efficient Federated Learning with Non-IID Data for Vehicular Ad Hoc Networks. Em *IEEE Global Communications Conference (GLOBECOM)*, p. 1–6. IEEE.
- [Li et al., 2021b] Li, C., Li, G. e Varshney, P. K. (2021b). Decentralized Federated Learning via Mutual Knowledge Transfer. *IEEE Internet of Things Journal*, 9(2):1136–1147.
- [Li et al., 2023] Li, H., Cai, Z., Wang, J., Tang, J., Ding, W., Lin, C.-T. e Shi, Y. (2023). FedTP: Federated Learning by Transformer Personalization. *IEEE Transactions on Neural Networks and Learning Systems*.
- [Li et al., 2024] Li, J., Chen, T. e Teng, S. (2024). A Comprehensive Survey on Client Selection Strategies in Federated Learning. *Computer Networks*, 251:110663.
- [Li et al., 2021c] Li, Q., He, B. e Song, D. (2021c). Model-Contrastive Federated Learning. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 10713–10722.
- [Li et al., 2021d] Li, T., Hu, S., Beirami, A. e Smith, V. (2021d). Ditto: Fair and Robust Federated Learning through Personalization. Em *International Conference on Machine Learning*, p. 6357–6368. PMLR.
- [Li et al., 2020] Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A. e Smith, V. (2020). Federated Optimization in Heterogeneous Networks. *Machine Learning and Systems*, 2:429–450.

- [Li et al., 2022b] Li, X., Lu, L., Ni, W., Jamalipour, A., Zhang, D. e Du, H. (2022b). Federated Multi-Agent Deep Reinforcement Learning for Resource Allocation of Vehicle-to-Vehicle Communications. *IEEE Transactions on Vehicular Technology*, 71(8):8810–8824.
- [Liang et al., 2025] Liang, T., Hu, M. e Sun, E. (2025). Mixture of specialized experts for model-heterogeneous personalized federated learning. *IEEE Networking Letters*.
- [Liang et al., 2022] Liang, X., Lin, Y., Fu, H., Zhu, L. e Li, X. (2022). RSCFed: Random Sampling Consensus Federated Semi-Supervised Learning. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 10154–10163.
- [Liu et al., 2024a] Liu, J., Jia, J., Che, T., Huo, C., Ren, J., Zhou, Y., Dai, H. e Dou, D. (2024a). FedASMU: Efficient Asynchronous Federated Learning with Dynamic Staleness-aware Model Update. Em *Conference on Artificial Intelligence (AAAI)*, volume 38, p. 13900–13908.
- [Liu et al., 2020] Liu, L., Zhang, J., Song, S. e Letaief, K. B. (2020). Client-Edge-Cloud Hierarchical Federated Learning. Em *IEEE International Conference on Communications (ICC)*, p. 1–6. IEEE.
- [Liu et al., 2024b] Liu, Y., Shi, Y., Li, Q., Wu, B., Wang, X. e Shen, L. (2024b). Decentralized Directed Collaboration for Personalized Federated Learning. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 23168–23178.
- [Lopez et al., 2018] Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P. e Wiessner, E. (2018). Microscopic Traffic Simulation using SUMO. Em *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, p. 2575–2582.
- [Luo et al., 2022] Luo, B. et al. (2022). Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling. Em *IEEE INFOCOM*, p. 1739–1748.
- [Luo et al., 2023] Luo, K., Wang, S., Fu, Y., Li, X., Lan, Y. e Gao, M. (2023). DFRD: Data-Free Robustness Distillation for Heterogeneous Federated Learning. *Advances in Neural Information Processing Systems*, 36:17854–17866.
- [Magdum et al., 2021] Magdum, S. S., Franklin, A. e Tamma, B. R. (2021). A Cooperative Federated Learning Mechanism for Collision Avoidance using Cellular and 802.11p based Radios Opportunistically. Em *2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, p. 284–289.
- [Mayhoub e M. Shami, 2024] Mayhoub, S. e M. Shami, T. (2024). A Review of Client Selection Methods in Federated Learning. *Archives of Computational Methods in Engineering*, 31(2):1129–1152.
- [McMahan et al., 2017] McMahan, B. et al. (2017). Communication-efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, p. 1273–1282.

- [Moon et al., 2024] Moon, J., Yang, S. e Lee, K. (2024). FedOps: A Platform of Federated Learning Operations With Heterogeneity Management. *IEEE Access*, 12:4301–4314.
- [Moon e Lim, 2024] Moon, S. e Lim, Y. (2024). Client Selection for Federated Learning in Vehicular Edge Computing: A Deep Reinforcement Learning Approach. *IEEE Access*, 12:131337–131348.
- [Neto et al., 2022] Neto, H. N. C., Dusparic, I., Mattos, D. M. e Fernande, N. C. (2022). FedSA: Accelerating Intrusion Detection in Collaborative Environments with Federated Simulated Annealing. Em *International Conference on Network Softwarization (NetSoft)*, p. 420–428. IEEE.
- [Nguyen et al., 2022] Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M. e Huba, D. (2022). Federated Learning with Buffered Asynchronous Aggregation. Em *International conference on artificial intelligence and statistics*, p. 3581–3607. PMLR.
- [Nishio e Yonetani, 2019] Nishio, T. e Yonetani, R. (2019). Client Selection for Federated Learning with Heterogeneous Resources in Mobile Edge. Em *International Conference on Communications*, p. 1–7.
- [Oh et al., 2022] Oh, J., Kim, S. e Yun, S. Y. (2022). FedBABU: Toward Enhanced Representation for Federated Image Classification. Em *10th International Conference on Learning Representations, ICLR 2022*.
- [Palazzo et al., 2023] Palazzo, L., Pennisi, M., Bellitto, G. e Kavasidis, I. (2023). FeD-ZIO: Decentralized Federated Knowledge Distillation on Edge Devices. Em *International Conference on Image Analysis and Processing*, p. 201–210. Springer.
- [Posner et al., 2021] Posner, J., Tseng, L., Aloqaily, M. e Jararweh, Y. (2021). Federated learning in vehicular networks: Opportunities and solutions. *IEEE Network*, 35(2):152–159.
- [Qi et al., 2022] Qi, P. et al. (2022). FedBKD: Heterogenous Federated Learning via Bidirectional Knowledge Distillation for Modulation Classification in IoT-Edge System. *IEEE Journal of Selected Topics in Signal Processing*, 17(1):189–204.
- [Rai et al., 2022] Rai, S., Kumari, A. e Prasad, D. K. (2022). Client Selection in Federated Learning under Imperfections in Environment. *AI*, 3(1):124–145.
- [Sabuhi et al., 2024] Sabuhi, M., Musilek, P. e Bezemer, C.-P. (2024). Micro-FL: A Fault-Tolerant Scalable Microservice-based Platform for Federated Learning. *Future Internet*, 16(3):70.
- [Shi et al., 2023] Shi, Y., Shen, L., Wei, K., Sun, Y., Yuan, B., Wang, X. e Tao, D. (2023). Improving the model consistency of decentralized federated learning. Em *International Conference on Machine Learning*, p. 31269–31291. PMLR.

- [Soltani et al., 2022] Soltani, B., Haghghi, V., Mahmood, A., Sheng, Q. Z. e Yao, L. (2022). A Survey on Participant Selection for Federated Learning in Mobile Networks. Em *Workshop on Mobility in the Evolving Internet Architecture*, p. 19–24. ACM.
- [Song et al., 2024] Song, Y., Liu, H., Zhao, S., Jin, H., Yu, J., Liu, Y., Zhai, R. e Wang, L. (2024). FedAdKD: Heterogeneous Federated Learning via Adaptive Knowledge Distillation. *Pattern Analysis and Applications*, 27(4):134.
- [Stallkamp et al., 2011] Stallkamp, J., Schlipsing, M., Salmen, J. e Igel, C. (2011). The German Traffic Sign Recognition Benchmark: A Multi-Class Classification Competition. Em *International Joint Conference on Neural Networks*, p. 1453–1460. IEEE.
- [Stripelis et al., 2022] Stripelis, D., Thompson, P. M. e Ambite, J. L. (2022). Semi-synchronous federated learning for energy-efficient training and accelerated convergence in cross-silo settings. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(5):1–29.
- [Su et al., 2024] Su, D. et al. (2024). Communication Cost-Aware Client Selection in Online Federated Learning: A Lyapunov Approach. *Computer Networks*, p. 110517.
- [Sun et al., 2025] Sun, Y., Shen, L. e Tao, D. (2025). Towards understanding generalization and stability gaps between centralized and decentralized federated learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- [T Dinh et al., 2020] T Dinh, C., Tran, N. e Nguyen, J. (2020). Personalized Federated Learning with Moreau Envelopes. *Advances in Neural Information Processing Systems*, 33:21394–21405.
- [Thomaz et al., 2024] Thomaz, G. A., Ferreira, J. P. M. F., Barry, T., Sammarco, M. e Campista, M. E. M. (2024). UBOTA Protocol - UDP Bursts for Over-the-Air Secure Vehicular Software Updates. Em *2024 IEEE 13th International Conference on Cloud Networking (CloudNet)*, p. 1–9. ISSN: 2771-5663.
- [Thomaz et al., 2025] Thomaz, G. A., Silva, F. D. d. M., Souza, L. A. C. d., Costa, L. H. M. K. e Campista, M. E. M. (2025). AGATA – Arquitetura para Gerenciamento Automático de Tarefas de Aprendizado Federado. Em *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*, p. 121–128. SBC.
- [Wang et al., 2023] Wang, E., Chen, B., Chowdhury, M., Kannan, A. e Liang, F. (2023). FLINT: A Platform for Federated Learning Integration. *Machine Learning and Systems*, 5:21–34.
- [Wang et al., 2020a] Wang, H., Kaplan, Z., Niu, D. e Li, B. (2020a). Optimizing Federated Learning on Non-IID Data with Reinforcement Learning. Em *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, p. 1698–1707.
- [Wang et al., 2020b] Wang, J. et al. (2020b). Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization. *NeurIPS*, 33:7611–7623.

- [Wang et al., 2022] Wang, Z., Zhang, Z., Tian, Y., Yang, Q., Shan, H., Wang, W. e Quek, T. Q. (2022). Asynchronous federated learning over wireless communication networks. *IEEE Transactions on Wireless Communications*, 21(9):6961–6978.
- [Wu et al., 2021a] Wu, W. et al. (2021a). SAFA: A Semi-Asynchronous Protocol for Fast Federated Learning with Low Overhead. *IEEE Transactions on Computers*, 70(5):655–668.
- [Wu et al., 2021b] Wu, Y., Kang, Y., Luo, J., He, Y. e Yang, Q. (2021b). FedCG: Leverage Conditional GAN for Protecting Privacy and Maintaining Competitive Performance in Federated Learning. *arXiv preprint arXiv:2111.08211*.
- [Wu et al., 2024a] Wu, Z., Sun, S., Wang, Y., Liu, M., Gao, B., Pan, Q., He, T. e Jiang, X. (2024a). Agglomerative Federated Learning: Empowering Larger Model Training via End-Edge-Cloud Collaboration. Em *IEEE Conference on Computer Communications (INFOCOM)*, p. 131–140. ISSN: 2641-9874.
- [Wu et al., 2024b] Wu, Z., Sun, S., Wang, Y., Liu, M., Pan, Q., Zhang, J., Li, Z. e Liu, Q. (2024b). Exploring the Distributed Knowledge Congruence in Proxy-data-free Federated Distillation. *ACM Transactions on Intelligent Systems and Technology*, 15(2):1–34.
- [Xiao et al., 2025] Xiao, S., Huang, X., Zhou, M., Liang, C. e Chen, Q. (2025). Fed-DLD: Dual-Level Federated Distillation with Adaptive Knowledge Transfer for DAG-secured IoVs. *IEEE Transactions on Vehicular Technology*.
- [Xie et al., 2019] Xie, C., Koyejo, S. e Gupta, I. (2019). Asynchronous Federated Optimization. *arXiv e-prints*, p. arXiv:1903.03934.
- [Xie et al., 2026] Xie, R., Liang, W., Diao, Z., Meng, X. e Li, K. (2026). Personalized hierarchical federated learning framework for the internet of vehicles based on split meta-learning. *IEEE Internet of Things Journal*.
- [Xu et al., 2023] Xu, C., Qu, Y., Xiang, Y. e Gao, L. (2023). Asynchronous federated learning on heterogeneous devices: A survey. *Computer Science Review*, 50:100595.
- [Xu et al., 2024a] Xu, J., Wan, S., Li, Y., Luo, S., Chen, Z., Shao, Y., Chen, Z., Huang, S.-L. e Song, L. (2024a). Cooperative Multi-Model Training for Personalized Federated Learning Over Heterogeneous Devices. *Journal of Selected Topics in Signal Processing*, 19(1):195–207.
- [Xu et al., 2024b] Xu, Z., Jiang, F., Niu, L., Jia, J. e Poovendran, R. (2024b). Brave: Byzantine-Resilient and Privacy-Preserving Peer-to-Peer Federated Learning. *arXiv:2401.05562 [cs]*.
- [Yang et al., 2023] Yang, S., Moon, J., Kim, J., Lee, K. e Lee, K. (2023). FLScalize: Federated Learning Lifecycle Management Platform. *IEEE Access*, 11:47212–47222.
- [Yang et al., 2022] Yang, T.-J., Guliani, D., Beaufays, F. e Motta, G. (2022). Partial Variable Training for Efficient On-Device Federated Learning. Em *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 4348–4352. IEEE.

- [Ye e others, 2023] Ye, M. e others (2023). Heterogeneous Federated Learning: State-of-the-Art and Research Challenges. *ACM Comput. Surv.*, 56(3).
- [Yi et al., 2026] Yi, L., Yu, H., Wang, G., Liu, X. e Hu, Q. (2026). pFedMoE: Data-Level Personalization With Mixture of Experts in Model-Heterogeneous Personalized Federated Learning. *IEEE Transactions on Knowledge and Data Engineering*.
- [Yuan et al., 2024] Yuan, L., Wang, Z., Sun, L., Yu, P. S. e Brinton, C. G. (2024). Decentralized Federated Learning: A Survey and Perspective. *IEEE Internet of Things Journal*, 11(21):34617–34638.
- [Zejun et al., 2024] Zejun, L. et al. (2024). Mitigating Straggler Effect in Federated Learning based on Reconfigurable Intelligent Surface over Internet of Vehicles. *China Communications*, 21(8):62–78.
- [Zhang et al., 2022a] Zhang, J., Chen, C., Li, B., Lyu, L., Wu, S., Ding, S., Shen, C. e Wu, C. (2022a). DENSE: Data-Free One-Shot Federated Learning. *Advances in Neural Information Processing Systems*, 35:21414–21428.
- [Zhang et al., 2022b] Zhang, J., Li, Z., Li, B., Xu, J., Wu, S., Ding, S. e Wu, C. (2022b). Federated Learning with Label Distribution Skew via Logits Calibration. Em Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G. e Sabato, S., editors, *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, p. 26311–26329. PMLR.
- [Zhang et al., 2022c] Zhang, L. et al. (2022c). Fine-Tuning Global Model via Data-Free Knowledge Distillation for Non-IID Federated Learning. Em *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, p. 10174–10183.
- [Zhang et al., 2021] Zhang, P., Wang, C., Jiang, C. e Han, Z. (2021). Deep Reinforcement Learning Assisted Federated Learning Algorithm for Data Management of IIoT. *IEEE Transactions on Industrial Informatics*, 17(12):8475–8484.
- [Zhang et al., 2023] Zhang, X. et al. (2023). Federated Learning-Assisted Vehicular Edge Computing: Architecture and Research Directions. *IEEE Vehicular Technology Magazine*, 18(4):75–84.
- [Zhang et al., 2022d] Zhang, Y. et al. (2022d). S4OD: Semi-Supervised Learning for Single-Stage Object Detection.
- [Zhao et al., 2023] Zhao, H., Du, W., Li, F., Li, P. e Liu, G. (2023). FedPrompt: Communication-Efficient and Privacy-Preserving Prompt Tuning in Federated Learning. Em *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5. IEEE.
- [Zhu et al., 2021a] Zhu, Q. et al. (2021a). 3GPP TR 38.901 Channel Model. Em *The Wiley 5G Ref: The Essential 5G Reference Online*, p. 1–35. Wiley Press Hoboken.
- [Zhu et al., 2021b] Zhu, Z., Hong, J. e Zhou, J. (2021b). Data-Free Knowledge Distillation for Heterogeneous Federated Learning. Em *International Conference on Machine Learning (PMLR)*, p. 12878–12889. ISSN: 2640-3498.