TOFL: Time Optimized Federated Learning

Lucas Airam C. de Souza[1,2], Matteo Sammarco[3], Nadjib Achir[2], Miguel Elias M. Campista[1], Luís Henrique M. K. Costa[1]

[11]Universidade Federal do Rio de Janeiro (UFRJ)
[2]INRIA Saclay, École Polytechnique, France
[3]Stellantis

{airam,miguel,luish}@gta.ufrj.br
matteo.sammarco@stellantis.com
nadjib.achir@inria.fr

Resumo. *As redes veiculares enfrentam ameaças cibernéticas que podem prejudicar motoristas, passageiros e pedestres. Nesse cenário, uma possível solução para treinar modelos que detectem ameaças, sem violação da privacidade dos usuários, é o aprendizado federado. No entanto, o aprendizado federado é particularmente sensível a atrasos de comunicação, sendo esta uma consequência natural da alta mobilidade em redes veiculares. Tal problema é comumente ignorado pela literatura, que não considera a possibilidade de desconexões na rede. Este trabalho propõe uma estratégia de seleção de clientes projetada para minimizar o tempo de treinamento de um modelo de aprendizado de máquina para detecção de ameaças veiculares, considerando o tempo de comunicação que varia de acordo com a movimentação dos clientes. Os resultados demonstram que o TOFL, utilizando apenas 20% do total de clientes disponíveis, pode reduzir o tempo necessário para atingir alta acurácia em até 50% em comparação com abordagens do estado da arte, ao mesmo tempo que diminui o consumo de recursos dos dispositivos clientes.*

Abstract. *Vehicular networks face cyber threats that can harm drivers, passengers, and pedestrians. In this scenario, federated learning is a possible solution to train models that detect threats without violating user privacy. However, federated learning is particularly sensitive to communication delays, which is a natural consequence of high mobility in vehicular networks. This problem is commonly ignored in the literature, which does not consider the possibility of network disconnections. This work proposes a client selection strategy designed to minimize the training time of a machine learning model for vehicular threat detection, considering the communication time that varies according to the movement of clients. The results demonstrate that TOFL, using only 20% of the total available clients, can reduce the time required to achieve high accuracy by up to 50% compared to state-of-the-art approaches, while reducing the resource consumption of client devices.*

O presente trabalho foi realizado com apoio do CNPq, Processo 405940/2022-0; da CAPES (Código de Financiamento 001); da FAPERJ; e da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Processos 2023/00673-7 e 2023/00811-0. Os autores agradecem também à Fundação de Desenvolvimento da Pesquisa (Fundep), no âmbito do programa Rota 2030, e às empresas Stellantis e Mobway pelo apoio e colaboração que tornaram esta pesquisa possível.

1. Introdução

Os veículos conectados (*Connected Vehicles* - CVs) melhoram a experiência de condução utilizando sensores para compreender as condições de tráfego ao redor e se comunicando com outros veículos por meio de uma Rede Ad Hoc Veicular (*Vehicular Ad Hoc Network* - VANET). Informações externas são trocadas por meio de padrões diferentes: Mensagens Básicas de Segurança (*Basic Safety Messages* - BSMs) [Committee et al. 2016], Mensagens Descentralizadas de Notificação Ambiental (*Decentralized Environmental Notification Messages* - DENM) [ETSI 2014b] ou Mensagens de Reconhecimento Cooperativo (*Cooperative Awareness Messages* - CAMs) [ETSI 2014a], o que permite a seleção otimizada de rotas. Nas VANETs as identidades dos remetentes podem ser verificadas por meio de assinaturas digitais que dependem de chaves distribuídas por uma Infraestrutura de Chaves Públicas (*Public Key Infrastructure*- PKI). No entanto, verificar a identidade do usuário não garante a veracidade das mensagens. Como resultado, usuários mal-intencionados podem enviar informações falsas para causar distúrbios de tráfego.

Atualmente, pesquisadores se concentram no uso de técnicas de aprendizado de máquina para automatizar a detecção de comportamentos maliciosos em redes veiculares [Bousalem et al. 2023, Boualouache et al. 2023, Van Der Heijden et al. 2018, Kamel et al. 2020, Guimaraes et al. 2022]. No entanto, os modelos são dependentes de grandes volumes de dados, que são privados e escassos. Assim, o Aprendizado Federado (*Federated Learning* - FL) surge como uma alternativa ao treinamento distribuído, sem compartilhamento de dados, preservando a privacidade dos usuários [McMahan et al. 2017, de Souza et al. 2024].

O FL já é aplicado à detecção de ameaças em VANETs [Boualouache e Engel 2022, Korba et al. 2023, Vinita e Vetriselvi 2023, Zhong et al. 2023, Yakan et al. 2023, Neto et al. 2024]. Devido à natureza distribuída do FL, muitos trabalhos se concentram em estratégias de seleção otimizada de usuários a fim de reduzir o tempo de treinamento [Luo et al. 2022, Su et al. 2024]. O tempo de comunicação entre os clientes e o servidor de agregação do FL é dinâmico em redes veiculares, pois a mobilidade dos veículos impacta as condições do canal de comunicação [Fittipaldi et al. 2025]. No entanto, muitas propostas consideram o tempo de comunicação durante o treinamento como um parâmetro fixo ou o excluem da formulação. [Buyukates e Ulukus 2021] propõem a estratégia m-*fastest* para selecionar o subconjunto de clientes que respondem mais rápido, considerando todos os atrasos envolvidos no processo de treinamento. Porém, a proposta desperdiça recursos computacionais dos clientes ao descartar os mais lentos do processo de agregação.

Este trabalho propõe o *Time Optimized Federated Learning* (TOFL)[1], uma estratégia de otimização para seleção de clientes no aprendizado federado em redes sem fio para reduzir o tempo de treinamento de cada época global. A proposta reduz o tempo de treinamento a partir da formulação de um problema de otimização que seleciona os melhores usuários para minimizar o atraso total de cada época. O problema de otimização toma como entrada o tempo de comunicação e o tempo de processamento observado pelos usuários, o número de clientes para selecionar e retorna os clientes selecionados. A estimativa dos atrasos de comunicação é realizada a partir do armazenamento dos 10 atrasos

[1]Disponível em `https://github.com/AiramL/TimeOptimizedFederatedLearning`.

anteriores de mensagens *keep alive* no servidor. Essa informação é utilizada como entrada de um modelo *Long Short-Term Memory* (LSTM). Os tempos de comunicação são obtidos executando uma simulação que implementa um padrão de comunicação FL em uma rede 5G com um padrão de mobilidade gerado a partir da Simulação de Mobilidade Urbana (*Simulation of Urban MObility* - SUMO). Considera-se o problema de detecção de ameaças em redes veiculares por meio de CAM como a tarefa de aprendizado. O objetivo dos invasores é interromper as condições de tráfego, enquanto o modelo treinado determina se as mensagens são reais ou contêm informações falsas. Os resultados mostram que, com 20% do número total de clientes disponíveis, o TOFL é capaz de reduzir até 50% do tempo de treinamento em comparação com a abordagem de seleção aleatória e 33% quando comparado com o m-*fastest*. O TOFL apresenta o menor tempo até convergência, reduzindo o tempo total de treinamento com alto desempenho. Além disso, o TOFL é mais eficiente em relação aos recursos computacionais usados para treinar o modelo global quando comparado com o m-*fastest*, pois ele calcula o modelo global usando todas as respostas dos clientes.

As principais contribuições deste trabalho são:

- a proposta do TOFL, uma estratégia de seleção de clientes de aprendizado federado para reduzir o tempo de treinamento. O TOFL minimiza o tempo de treinamento e torna o modelo disponível mais rapidamente aos usuários, selecionando o subconjunto de clientes com as melhores condições computacionais e de comunicação.
- A abordagem dos padrões de mobilidade dos clientes, uma lacuna existente no estado da arte em seleção de clientes usados pelo FL.
- A disponibilização de código que reproduz o padrão de comunicação do aprendizado federado em uma rede 5G. Isso permite a outros pesquisadores avaliar propostas considerando as condições de comunicação no aprendizado federado.

Este trabalho está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 apresenta a formulação matemática do problema e a proposta do TOFL, enquanto a Seção 4 apresenta o modelo de atacante usado no artigo. A metodologia e resultados dos experimentos com o TOFL são apresentados na Seção 5. Finalmente, a Seção 6 conclui este trabalho e fornece perspectivas futuras.

2. Trabalhos Relacionados

Esta seção apresenta os principais trabalhos que visam identificar ameaças em redes veiculares. Além disso, discute-se também os trabalhos que otimizam o aprendizado federado reduzindo o tempo total de treinamento.

2.1. Detecção de Ameaças em Redes Veiculares

[Bousalem et al. 2023] propõem uma estratégia de aprendizado por reforço para mitigar ataques DDoS em redes veiculares. O cenário proposto permite instanciar fatias de rede com recursos de comunicação baixos para isolar invasores ou clientes com comportamento suspeito. No entanto, esses nós isolados devem recuperar recursos após o fim da ameaça. Assim, os autores propõem um algoritmo de aprendizado por reforço para determinar quando reduzir os recursos dos usuários atribuindo-os a uma fatia com recursos limitados ou quando aumentar seus recursos novamente. No entanto, a proposta carece de

mecanismos de preservação de privacidade, como treinamento por meio de aprendizado federado e um conjunto de dados disponível publicamente.

[Korba et al. 2023] propõem uma solução de detecção de ameaças em redes veiculares 5G. Os autores focam as comunicações V2X e usam o aprendizado federado para treinar os modelos para detectar ataques. A proposta usa um modelo de *autoencoder* para executar treinamento não supervisionado, pois o objetivo é não depender de dados rotulados e detectar ataques de dia zero. A suposição é que o tráfego benigno é o tipo de dado mais comum. Assim, o *autoencoder* aprende a representar essa classe em um espaço latente com um erro de representação menor do que as classes anômalas. Portanto, o sistema gera uma nova representação do fluxo de tráfego e compara a entrada com a saída para determinar o erro de representação que pode identificar ataques. No entanto, os autores desconsideram a mobilidade dos clientes.

[Vinita e Vetriselvi 2023] propõem um sistema para identificar a veracidade de mensagens de emergência transmitidas em redes veiculares. Por um lado, a comunicação V2X traz oportunidades para aumentar a eficiência da direção ao enviar uma atualização sobre as condições de tráfego local, como informar acidentes. Por outro lado, os invasores podem difundir mensagens falsas para degradar as condições de tráfego. Assim, o artigo propõe usar modelos de aprendizado de máquina para detectar esse comportamento malicioso, por exemplo, ataques Sybil. Além disso, a proposta usa o paradigma de aprendizado federado para preservar a privacidade dos usuários.

[Gong et al. 2022] investigam como o desequilíbrio de amostras causa maiores valores de função de perda das abordagens de agrupamento em FL. Clientes com poucas amostras têm mais dificuldade em treinar o modelo, o que incorre em perdas maiores. O oposto ocorre com clientes que têm muitas amostras. Como as técnicas de agrupamento aplicadas em FL geralmente usam como entrada os pesos da rede neural ou as perdas para definir o agrupamento, o desequilíbrio dos conjuntos de dados compromete a definição do agrupamento. No entanto, as perdas locais podem ser reduzidas executando mais épocas de treinamento local. Assim, os autores propõem equalizar as perdas dos clientes executando um número adaptativo de épocas locais dependendo do número de amostras que cada cliente detém. Os resultados mostram que a proposta reduz o número necessário de rodadas de comunicação global e atribui corretamente clientes com dados semelhantes ao mesmo grupo.

[Zhong et al. 2023] propõem um modelo BiGAN baseado em LSTM para detectar ameaças de rede veicular baseadas em mensagens CAM. Os autores usam o conjunto de dados *Vehicular Reference Misbehavior Dataset* (VeReMi) [Van Der Heijden et al. 2018] para treinar e testar seu modelo. Devido ao seu alto desempenho no problema de classificação, o TOFL usa a mesma estratégia de pré-processamento para identificar os ataques no conjunto de dados. [Yakan et al. 2023] também propõe um sistema de detecção de intrusão para identificar ataques em mensagens CAM. Os autores aplicam um modelo LSTM e uma transformação de características no conjunto de dados VeReMi Extension [Kamel et al. 2020] para identificar um conjunto mais amplo de ameaças em redes veiculares. No entanto, ambas as propostas ignoram que a seleção do cliente tem o potencial de minimizar o tempo de treinamento.

2.2. Otimização do Aprendizado Federado

[Luo et al. 2022] propõem um algoritmo de seleção de clientes para aumentar a eficiência do aprendizado federado. A proposta considera que os clientes têm diferentes dispositivos e também heterogeneidade de dados, o que impõe um desafio para selecionar o melhor subconjunto de clientes para treinar o modelo durante uma época global. Assim, os autores formulam um problema de otimização para selecionar os clientes minimizando o tempo total de treinamento, considerando a relevância dos dados e capacidade dos dispositivos. O tempo de comunicação porém é desconsiderado na formulação, sendo importante especialmente em cenários com clientes móveis, como redes veiculares.

[Su et al. 2024] propõem um algoritmo de seleção de clientes para o *Online Federated Learning* (OFL), chamado algoritmo *Low-Cost Client Selection* (LCCS). Os autores formulam um problema de otimização para maximizar a utilidade do modelo e minimizar o custo de comunicação. No entanto, a avaliação considera apenas a largura de banda necessária para determinar o custo de comunicação. Assim, os autores desconsideram o tempo de convergência, que é particularmente sensível para aplicações como detecção de intrusão.

[Buyukates e Ulukus 2021] consideram um cenário de aprendizado federado onde o servidor de agregação realiza uma seleção aleatório 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 de volta ao servidor de agregação. A vantagem da proposta é utilizar uma formulação simples que dispensa a formulação de um problema de otimização para a seleção de clientes. No entanto, a proposta introduz energia desnecessária e desperdício computacional para os clientes com maior tempo de resposta selecionados.

Diferentemente das propostas acima, o TOFL é uma estratégia de seleção de clientes de aprendizado federado que considera as variações de atraso dos clientes causadas pela mobilidade e evitando o desperdício de recursos computacionais. O TOFL é adaptado para redes veiculares onde os clientes estão se movendo e experienciam diferentes condições de rede.

3. Time Optimized Federated Learning (TOFL)

O Time Optimized Federated Learning é uma estratégia de seleção de clientes FL que considera as condições de rede dos usuários para reduzir o tempo de treinamento em ambientes veiculares. Esta seção descreve as principais hipóteses, como modelar para obter os atrasos de comunicação e o problema de otimização.

3.1. Hipóteses Consideradas

Neste trabalho, assume-se que os clientes do aprendizado federado são veículos conectados com recursos computacionais suficientes para participar do treinamento do modelo. Os recursos computacionais e a carga de trabalho dos dispositivos dos clientes são homogêneos. A tarefa de aprendizado é a detecção de informações falsas ou ataques executados por meio de Mensagens de Reconhecimento Cooperativo (*Cooperative Awareness Messages* - CAMs), que são transmitidas na rede veicular. No início do treinamento, assume-se que cada cliente detém um conjunto de dados privado, rotulado, para ajustar o modelo durante o treinamento federado de forma supervisionada. Além disso,

Tabela 1. Notações usadas no artigo.

e	Índice da época global.
k	Índice do cliente.
E	Eficiência do sistema.
$T_G^{e,k}$	Tempo total de época para o k-ésimo cliente na e-ésima época.
$T_D^{e,k}$	Tempo de receber o modelo para o k-ésimo cliente na e-ésima época.
$T_C^{e,k}$	Tempo computacional para o k-ésimo cliente na e-ésima época.
$T_U^{e,k}$	Tempo de enviar o modelo para o k-ésimo cliente na e-ésima época.
$\mathcal{K}$	Subconjunto de clientes disponíveis.
$\mathcal{N}$	Subconjunto de clientes selecionados.
$\mathcal{M}$	Subconjunto de clientes usados para agregação pelo algoritmo M-Fastest.
$\mathcal{A}$	Subconjunto de clientes usados para agregação.
T_{timeout}	Atraso máximo tolerado para encerrar a época global.
d_k	Variável binária que indica se o i-ésimo cliente é selecionado para participar da época global atual.

os veículos enviam mensagens *keep alive* para o servidor de agregação para utilizar como entrada de uma rede neural LSTM e estimar as condições de rede futura para seleção de clientes. Devido à mobilidade dos veículos, as condições de rede variam com o tempo. Assume-se ainda que a distribuição de dados dos veículos é Independente e Distribuída Identicamente (IID)

3.2. Problema de Otimização

O diagrama de execução de uma época global do cenário é ilustrado na Figura 1 e a notação matemática utilizada é apresentada na Tabela 1. O cliente K_i no diagrama inicialmente se conecta com o servidor de aprendizado federado S. Uma vez conectado, o cliente pode ser selecionado para treinar o modelo na t-ésima época global. O servidor transmite o modelo para o cliente selecionado, o que leva $T_D^{e,k}$ unidades de tempo para concluir a transmissão ("download" do modelo). Então, o cliente leva $T_C^{e,k}$ unidades de tempo para treinar e atualizar os parâmetros do modelo local. Após concluir, o cliente transfere o modelo para o servidor em $T_U^{e,k}$ unidades de tempo ("upload" do modelo). Esse processo se repete até o fim do treinamento do modelo no aprendizado federado sempre que o cliente for selecionado.

Cada veículo tem três atrasos associados em cada época global: receber, atualizar e enviar o modelo. O tempo para executar uma época global no tempo t para um cliente i é definido como a soma dos três atrasos $T_{G_{t,i}} = T_D^{e,k} + T_C^{e,k} + T_U^{e,k}$, onde $T_D^{e,k}$ é o tempo para um cliente i receber o modelo do servidor, $T_C^{e,k}$ é o tempo necessário para executar as épocas locais e $T_U^{e,k}$ representa o tempo necessário para enviar o modelo local atualizado para o servidor. $T_D^{e,k}$ e $T_C^{e,k}$ são fáceis de estimar dada uma época, no entanto, $T_U^{e,k}$ é difícil de determinar devido à variabilidade das condições de rede e mobilidade dos clientes ao longo do tempo, principalmente quando o tempo de computação é longo. No entanto, os usuários geralmente têm padrões de mobilidade específicos, tornando possível prever melhor a taxa de transferência [do Couto Teixeira et al. 2021, Gonzalez et al. 2008] e, consequentemente, estimar um valor para $T_U^{e,k}$ próximo ao real. Dessa forma, o tempo de enviar o modelo pode ser aproximado pelo tempo estimado de receber o modelo.

A proposta estima o tempo necessário para cada cliente receber o modelo e utiliza a estimativa para selecionar os clientes, uma vez que os atrasos de receber e enviar o

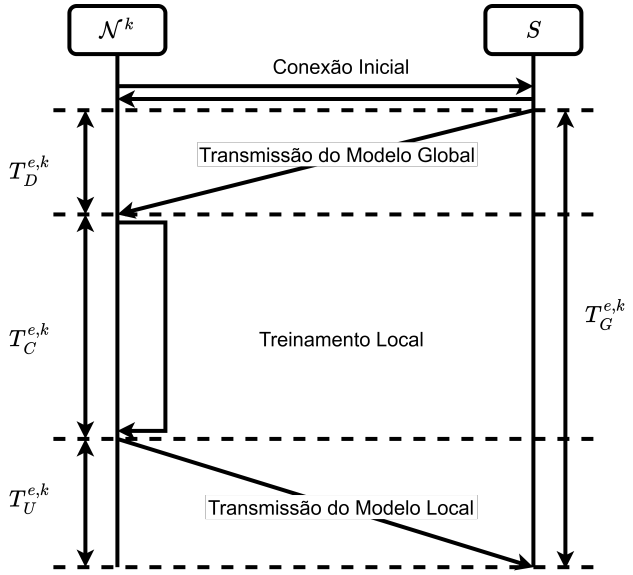

Figura 1. Diagrama de comunicação entre um cliente $\mathcal{N}^k$ selecionado para participar do treinamento na e-ésima época e o servidor de aprendizado federado.

modelo são correlacionados. Para estimar o atraso, o servidor armazena um vetor com as taxas de transmissão previamente experienciadas por cada cliente. O vetor é atualizado a cada 100 ms, extraindo as informações das mensagens *keep alive* definidas no TOFL. Quando o servidor precisa selecionar clientes, o vetor é utilizado como entrada de um modelo LSTM para estimar a condição de rede futura de cada cliente. Após esta etapa, o servidor é capaz de selecionar os clientes que têm os menores atrasos estimados. A seleção de clientes é modelada como um problema de *Facility Location* [Owen e Daskin 1998], que usa a função min-max como um critério de otimização, definido pela Equação 1.

$$\begin{aligned} & \min \max_{k \in \mathcal{K}} (T_D^{e,k} + T_C^{e,k} + T_U^{e,k}) \cdot d_k, \\ \text{sujeito a} : & \sum_{k \in \mathcal{K}} d_k = |\mathcal{N}|, \\ & (T_D^{e,k} + T_C^{e,k} + T_U^{e,k}) \cdot d_k \leq T_{\text{timeout}}, \forall k \in \mathcal{K}, \\ & e \in \mathbb{N}^+, T_D^{e,k}, T_C^{e,k}, T_U^{e,k} \in \mathbb{R}^+, d_k \in \{0,1\}. \end{aligned} \tag{1}$$

O problema possui duas restrições, a primeira indica que o número de clientes selecionados deve ser exatamente igual a $|\mathcal{N}|$. A segunda restrição indica que o atraso total estimado para um cliente, caso este seja selecionado, deve ser inferior ao tempo máximo permitido T_{timeout} para a execução da época global, para evitar que o cliente desperdice recursos computacionais.

4. Modelo de Atacante

Assume-se que os atacantes têm acesso a um veículo conectado já autenticado na rede. Assim, um nó atacante é capaz de enviar mensagens CAM na rede, que podem ou não conter informações falsas. Atacantes também têm a capacidade de armazenar e enviar mensagens anteriores, ou inundar a rede com mensagens repetidas.

O objetivo é identificar se um usuário é malicioso ou não, tendo acesso a mensagens passadas difundidas na rede. A detecção de ameaças é realizada usando um modelo de aprendizado profundo treinado em conjunto com clientes na rede. Ataques ao treinamento do modelo são ortogonais ao nosso trabalho atual e podem ser solucionados usando técnicas de agregação robusta [De Souza et al. 2024, Qi et al. 2024].

Tabela 2. Divisão de amostras de acordo com classes no conjunto de dados VeReMi.

Classe	ID	Quantidade de Amostras (#)	Percentual
Normal	0	1470668	69.14
Posição constante	1	136959	6.44
Deslocamento de posição constante	2	136959	6.44
Posição aleatória	4	150894	7.09
Deslocamento de posição aleatória	8	107888	5.07
Parada Eventual	16	123608	5.81

Para simular o modelo de atacante, são utilizados dois conjuntos de dados: Vehicular Reference Misbehavior Dataset (VeReMi) e VeReMi Extension. [Van Der Heijden et al. 2018] apresenta o VeReMi disponível publicamente e avalia mecanismos para avaliação da plausibilidade das informações contidas nas CAMs enviadas pelos veículos. Todos os ataques estão relacionados à posição do carro, enviando de alguma forma valores incorretos em relação ao tipo de ataque. O conjunto de dados consiste em logs de mensagens para cada veículo na simulação e um arquivo que especifica o tipo de comportamento. O VeReMi contém cinco ataques: constante, deslocamento constante, posição aleatória, deslocamento aleatório e parada eventual. Além disso, [Kamel et al. 2020] estende o VeReMi para incluir novos dados e padrões de ataque. Assim, o VeReMi Extension contém padrões como mensagens atrasadas, DoS, repetição de dados, difusão de mensagens de veículos falsos e mau funcionamento de velocidade, além dos ataques de mau funcionamento de posição na primeira versão do conjunto de dados. As distribuições de ataques do VeReMi e VeReMi Extension são apresentadas nas Tabelas 2 e 3 respectivamente.

5. Experimentos e Resultados

Esta seção apresenta os experimentos executados para avaliar o TOFL. Primeiramente, são descritos o ambiente e os parâmetros usados para executar os experimentos. A seguir apresenta-se a metodologia, experimentos executados e a discussão dos resultados obtidos.

5.1. Ambiente de Execução

Este trabalho utiliza a arquitetura do modelo LSTM-BiGAN [Zhong et al. 2023], descrita na Tabela 4, porque ela captura a dependência temporal das mensagens transmitidas. No entanto, o objetivo é validar a proposta e compará-la com outras técnicas. O TOFL é suficientemente genérico para ser usado como critério de seleção de clientes para o treinamento de outros modelos, inclusive aqueles que oferecem melhor desempenho que o usado na avaliação. Outros modelos podem ser adicionados ao sistema para serem usados com a estratégia proposta.

Tabela 3. Divisão de amostras de acordo com classes no conjunto de dados da extensão VeReMi.

Classe	ID	Quantidade de Amostras (#)	Percentual
Normal	0	1900539	59.49
Posição constante	1	43653	1.37
Deslocamento de posição constante	2	43567	1.36
Posição aleatória	3	43857	1.37
Mudança de posição aleatória	4	42575	1.33
Velocidade constante	5	41925	1.31
Mudança de velocidade constante	6	44359	1.39
Velocidade aleatória	7	42258	1.32
Mudança de velocidade aleatória	8	42583	1.33
Parada Eventual	9	42790	1.34
Disruptivo	10	43264	1.35
Repetição de dados	11	44337	1.39
Mensagens atrasadas	12	43118	1.35
Denial of Service (DoS)	13	131305	4.11
DoS aleatório	14	126724	3.97
DoS disruptivo	15	129270	4.05
Grade Sybil	16	175391	5.49
Repetição Sybil	17	44310	1.39
DoS Sybil aleatório	18	86883	2.72
DoS Sybil disruptivo	19	82100	2.57

O FL está configurado de acordo com os parâmetros da Tabela 5. O modelo de aprendizado profundo é implementado usando o framework TensorFlow v2.15.0. O treinamento é executado no flower v1.7.0[2]. Antes de iniciar o treinamento, há uma fase de pré-processamento onde calcula-se a correlação de Pearson para remover dos conjuntos de dados atributos com características altamente correlacionadas, amostras com valores ausentes e identificadores. Além disso, normaliza-se as características e divide-se aleatoriamente os conjuntos de dados em 80% de treinamento e 20% de teste para todos os clientes em ambos conjuntos de dados. Os clientes possuem os mesmos conjuntos de dados, de forma a evitar que as distribuições sejam não independentes e identicamente distribuídas.

O modelo de mobilidade veicular é implementado usando o SUMO v1.21.0, com o modelo de mobilidade de Manhattan [Patanè et al. 2024]. O tráfego de veículos é gerado pela ferramenta **randomTrips** em um comprimento de grade de 600 x 600 m^2, a uma

Tabela 4. Arquitetura do modelo LSTM usada nos experimentos para detectar ameaças à rede veicular.

Camada	**Formato da saída**	**Quantidade de parâmetros (#)**
LSTM	(None, None, 100)	43,600
LSTM	(None, 49)	29,400
Dense	(None, 6)	300

[2]o flower é um arcabouço de código aberto para aprendizado federado que permite treinar modelos de aprendizado de máquina de forma distribuída sem compartilhar os dados dos usuários.

Tabela 5. Parâmetros do aprendizado federado usados nos experimentos.

Parâmetro	**Valor**
Modelo	LSTM-BiGAN [Zhong et al. 2023]
Tamanho do modelo (kB)	573
Número de épocas globais (#)	40
Número de épocas locais (#)	2

velocidade constante de 30 km/h e em uma única faixa. A comunicação 5G dos clientes usa um modelo de canal, que aloca igualmente a largura de banda de 10 Gbps e 20 Gbps de *upload* e *download*, respectivamente, para os usuários e considera a distância dos usuários da estação base [Zhu et al. 2021, Chatzoulis et al. 2023]. Uma única estação base é usada para todos os usuários, que se conectam a um servidor de agregação remoto.

Os clientes do aprendizado federado são simulados por meio de múltiplos processos em um servidor. O servidor utilizado nos experimentos consiste em uma CPU AMD EPYC 7452 com 64 núcleos e 32 GB de RAM e equipada com duas GPUs NVIDIA Tesla V100S de 8 GB. Executa-se 10 vezes os mesmos experimentos para exibir os resultados dentro de um intervalo de confiança de 95%. Os resultados comparam cinco estratégias de seleção de clientes: aleatória [McMahan et al. 2017], M-Fastest [Buyukates e Ulukus 2021], TOFL oráculo, TOFL estimador e TOFL com M-Fastest. A seleção aleatória retorna exatamente $|\mathcal{N}|$ clientes diferentes amostrados de $\mathcal{K}$. A M-Fastest executa uma seleção aleatória de $|\mathcal{N}|$ clientes, no entanto, ele usa apenas os $|\mathcal{M}|$ primeiros clientes que enviam o modelo de volta ao servidor para agregar o modelo. O TOFL oráculo é a estratégia proposta com o conhecimento de todos os atrasos para estabelecer o melhor caso, enquanto o estimador é uma abordagem realista, onde o problema de otimização tem acesso aos atrasos estimados por meio da rede neural LSTM. Por fim, o estimador TOFL com M-Fastest, estima os atrasos dos clientes e seleciona exatamente $|\mathcal{M}|$ clientes.

O estimador de atraso, que possui a arquitetura 6, é treinado com um conjunto de dados com a vazão nominal de um cliente. É utilizada uma janela deslizante (de tamanho 10) para estimar os atrasos futuros. Uma vez que o modelo é treinado, os parâmetros são congelados para executar apenas previsões durante a operação TOFL. O primeiro experimento analisa o tempo necessário para treinar modelos de aprendizado profundo usando aprendizado federado. O segundo experimento compara a acurácia dos modelos e, por fim, discute-se o uso de recursos computacionais para cada estratégia.

Tabela 6. Arquitetura do modelo LSTM usada para estimar atrasos de clientes.

Camada	Formato de saída	Quantidade de parâmetros (#)
LSTM	(10 50)	10,600
Linear	(10, 1)	51

5.2. Tempo de Treinamento dos Modelos

Conforme dito anteriormente, os clientes têm três tempos diferentes para treinar um modelo usando aprendizado federado: receber, atualizar e enviar o modelo. O tempo de receber e enviar o modelo estão relacionados com a comunicação e os valores são obtidos executando simulações para cada cliente. Assume-se o tempo de processamento igual a zero para todos os clientes e a variação de tempo está relacionada apenas com as condições de comunicação.

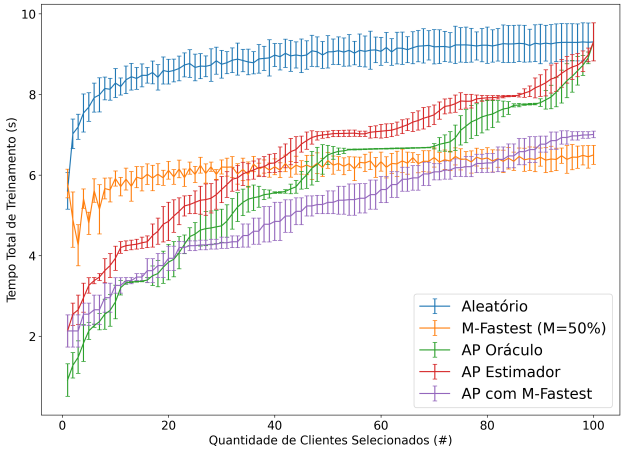

Figura 2. Tempo necessário para treinar o modelo de aprendizado federado de acordo com o número de clientes selecionados durante 40 épocas globais.

No primeiro experimento, avalia-se o atraso total para treinar um modelo de aprendizado profundo usando os parâmetros da Tabela 5. A Figura 2 mostra o resultado deste experimento. Para este experimento, varia-se a quantidade de clientes que participam do treinamento FL no intervalo de [1, 100]. Inicialmente, é possível observar que com $|\mathcal{N}| <= 35$, o TOFL é melhor do que todas as abordagens em termos de tempo necessário para treinar o modelo, mais de 200% e 100% menos tempo do que a abordagem aleatória e M-Fastest, respectivamente, ao estimar e selecionar 5 clientes, conforme mostrado pela linha vermelha na Figura 2. A Figura 3 mostra o tempo médio de cada época global ao selecionar 16 clientes. No entanto, para $55 <= |\mathcal{N}| <= 75$, apenas o estimador TOFL com a mesma quantidade de clientes que o M-Fastest apresenta um tempo menor que as outras propostas. Isso ocorre porque nessa abordagem são selecionados menos clientes para participar do treinamento, enquanto as outras abordagens, até mesmo o TOFL Oráculo, são forçadas a selecionar um número maior de clientes.

Além disso, com mais de 95 clientes, o M-Fastest apresenta o menor tempo para treinar o modelo, como mostrado no tempo médio de época global na Figura 4. De fato, quando aumenta-se o número de clientes selecionados na primeira seleção aleatória do M-Fastest, aumenta-se a probabilidade de selecionar os clientes mais rápidos. Portanto, se os melhores clientes forem escolhidos na seleção aleatória, o M-Fastest atuará como um oráculo. O mesmo não ocorre com o TOFL porque o estimador tem um erro de precisão inevitável, como é possível observar comparando o oráculo TOFL e o estimador TOFL, as linhas verde e vermelha na Figura 2. O estimador TOFL tem um tempo médio sempre maior que o oráculo, mostrando que a estimativa não está exatamente selecionando os melhores clientes.

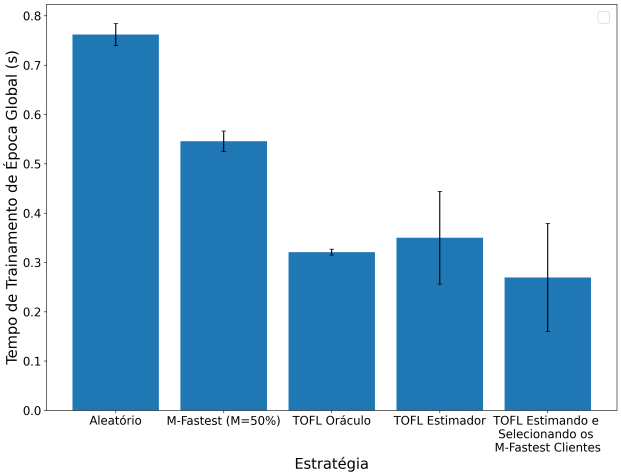

Figura 3. Duração média de cada época usando 16 clientes.

O resultado mostra que, para um número razoavelmente pequeno de clientes, por exemplo, 20% do número total de clientes disponíveis, o estimador TOFL é capaz de reduzir em até 50% do tempo de treinamento em comparação com a abordagem de seleção aleatória e 33% quando comparado com o M-Fastest.

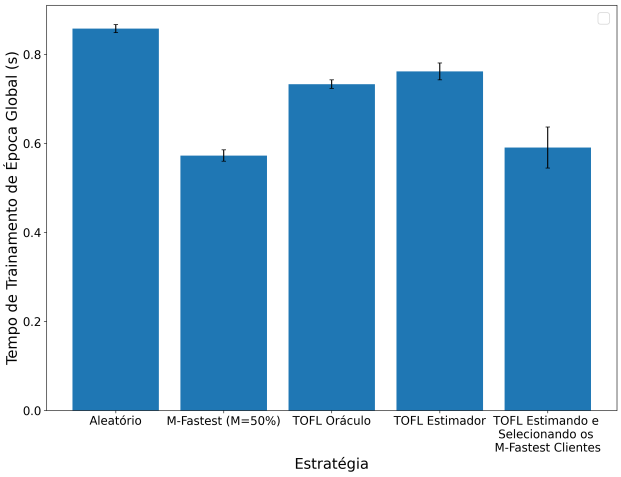

Figura 4. Duração média de cada época usando 95 clientes.

No entanto, uma comparação importante a ser feita é o tempo até convergência das diferentes estratégias. Portanto, na seção a seguir, é discutido quanto tempo leva para o modelo global convergir.

5.3. Tempo até a Convergência

Neste experimento, avalia-se o tempo até a convergência dentro de um número máximo de épocas, utilizando diferentes estratégias de seleção de clientes para os dois conjuntos de dados, com 16 e 95 clientes selecionados para participar do treinamento. Os números de clientes selecionados utilizados são baseados no experimento anterior.

As Figuras 5 e 6 mostram que, para ambos os conjuntos de dados, o TOFL apresenta o menor tempo para uma alta acurácia em comparação com outras propostas ao usar

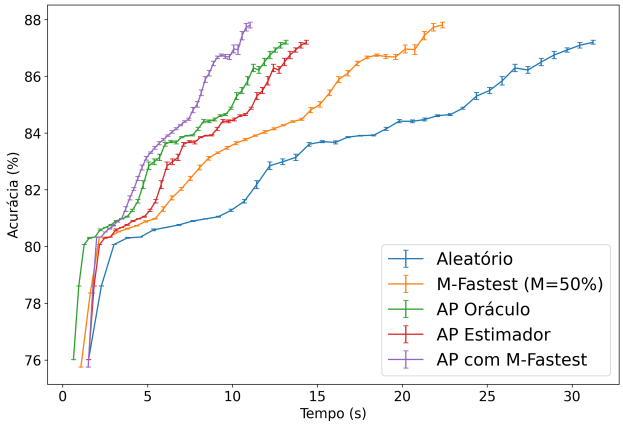

Figura 5. Tempo até a convergência da acurácia dentro de 35 épocas no conjunto de dados VeReMi com 16 clientes selecionados.

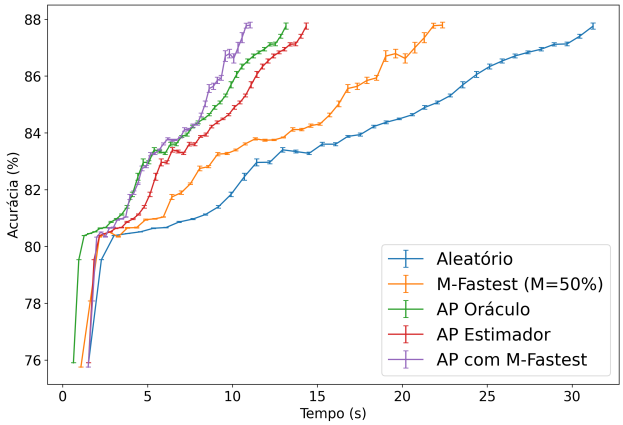

Figura 6. Tempo até a convergência da acurácia dentro de 30 épocas no conjunto de dados VeReMi Extension com 16 clientes selecionados.

16 clientes para participar do treinamento. Além disso, usar menos clientes, a linha roxa M-Fastest, é suficiente para atingir um bom nível de acurácia em um curto período de tempo, pois as estratégias baseadas no M-Fastest utilizam metade do conjunto de clientes selecionados.

Por outro lado, as Figuras 7 e 8 mostram que ao aumentar o número de clientes selecionados, o M-Fastest apresenta o menor tempo para acurácia em comparação com outras propostas e muito próximo ao tempo do TOFL com M-Fastest. No entanto, o melhor desempenho do M-Fastest é acompanhado de maior consumo de recursos computacionais.

5.4. Eficiência de Recursos Utilizados

Neste experimento, compara-se os recursos computacionais usados por cada abordagem. O objetivo é avaliar as estratégias em relação ao recursos utilizados e não aproveitados durante o treinamento. O desperdício de recursos abordado nesse experimento compreende o conjunto de recursos necessários para o treinamento federado, como CPU, memória, disco, energia e comunicação. Isso é especialmente interessante para comparar a proposta atual com a estratégia M-Fastest, pois esta proposta força alguns clientes a uti-

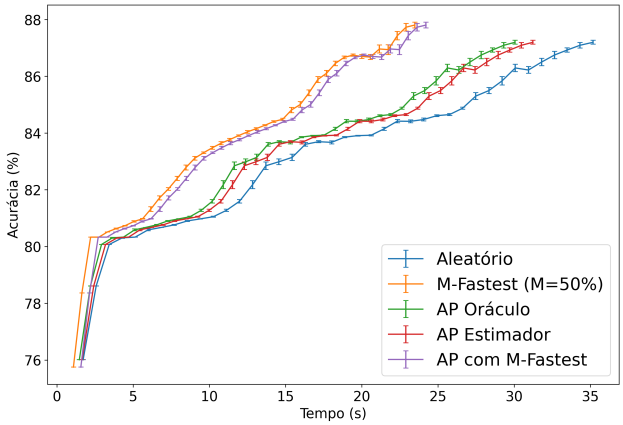

Figura 7. Tempo até a convergência da acurácia dentro de 35 épocas no conjunto de dados VeReMi ao selecionar 95 clientes.

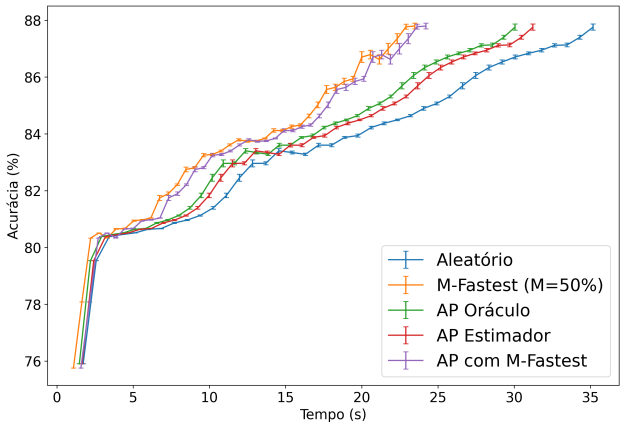

Figura 8. Tempo até a convergência da acurácia dentro de 30 épocas no conjunto de dados VeReMi Extension ao selecionar 95 clientes.

lizar desnecessariamente os recursos computacionais sem usar seus resultados no modelo global. Além disso, quando um cliente falha sem enviar seus resultados para agregação, também seus recursos computacionais são desperdiçados. Dessa forma, a Equação 2 descreve a eficiência E do sistema ao utilizar uma para selecionar os clientes. Por sua vez, cada estratégia possui uma quantidade $|\mathcal{A}|$ de recursos que são realmente usados para treinar o modelo global sobre os recursos totais $|\mathcal{N}|$ usados no sistema. Além disso, R representa um fator de escala que pode mudar em relação ao modelo de aprendizado de máquina usado no treinamento, ao tamanho dos conjuntos de dados dos clientes e à tecnologia de comunicação.

$$E = \frac{|\mathcal{A}|}{|\mathcal{N}|}.R. \tag{2}$$

A Figura 9 exibe os resultados do experimento, fixando o valor de $R = 1$ e variando o percentual de clientes que apresentam falhas em cada rodada. As falhas dos clientes são catastróficas dentro de uma rodada e simuladas de forma aleatória, sem correlação com a rede dos clientes. Para a abordagem aleatória e abordagens TOFL, $|\mathcal{A}| = |\mathcal{N}|$, o

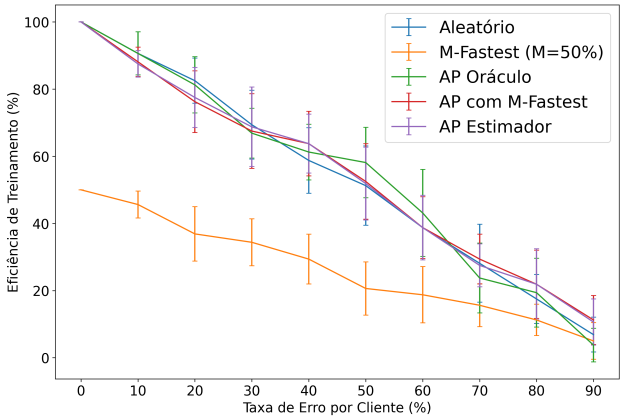

Figura 9. Eficiência do treinamento das 5 propostas ao variar a taxa de erro dos clientes.

que significa que não há perdas se os clientes selecionados enviarem o modelo de volta ao servidor de agregação antes de $T_{timeout}$ segundos. Isso explica a eficiência igual a 100% quando o cenário não apresenta falhas. Por outro lado, M-Fastest tem $|\mathcal{A}| < |\mathcal{N}|$, pois usa apenas 50% dos clientes selecionados para a agregação, o que incorre em desperdício de recursos dos clientes. Portanto, mesmo para um cenário sem falhas, M-Fastest possui um eficiência de apenas 50%. Mesmo quando aumenta-se o número de falhas, a eficiência das demais abordagens é superior ao M-Fastest. Assim, abordagens aleatórias e TOFL são mais eficientes em relação aos recursos computacionais usados para treinar o modelo global. O M-Fastest atinge a mesma eficiência apenas em cenários onde o número de falhas é superior a 70%. Além disso, o TOFL com M-Fastest pode ser usado para estimar os clientes mais rápidos, conforme mostrado nos resultados do tempo para treinar o modelo, e evitar o desperdício imposto por M-Fastest.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou o TOFL, uma estratégia de seleção de clientes para minimizar o tempo total de treinamento no cenário de aprendizado federado. Diferentemente de outras estratégias de seleção de clientes, o TOFL considera os efeitos da mobilidade dos clientes na comunicação, o que tem impacto significativo no cenário veicular. Para 20% do número total de clientes disponíveis, o estimador TOFL foi capaz de reduzir até 50% do tempo de treinamento em comparação com a abordagem de seleção aleatória e 33% quando comparado com o M-Fastest, uma estratégia de seleção do estado da arte. Mostrou-se que o TOFL é mais eficaz quando o sistema tem mais clientes e o número de clientes selecionados é menor do que o total de clientes disponíveis, o que é uma configuração comum em todos os cenários de aprendizado federado. Além disso, o TOFL é mais eficiente em relação aos recursos computacionais usados para treinar o modelo global quando comparado com o M-Fastest, pois calcula o modelo global usando todas as respostas dos clientes. Em trabalhos futuros, planeja-se criar um cenário mais realista usando a previsão de mobilidade dos usuários para estimar os atrasos de comunicação e investigar os efeitos de dados não-IID.

Referências

Boualouache, A. e Engel, T. (2022). Federated Learning-based Scheme for Detecting Passive Mobile Attackers in 5G Vehicular Edge Computing. *Annals of Telecommunications*, páginas 1–20.

Boualouache, A. et al. (2023). 5g vehicle-to-everything at the cross-borders: Security challenges and opportunities. *Internet of Things Magazine*, 6(1):114–119.

Bousalem, B., Sakka, M. A., Silva, V. F., Jaafar, W., Letaifa, A. B. e Langar, R. (2023). DDoS Attacks Mitigation in 5G-V2X Networks: A Reinforcement Learning-Based Approach. Em *International Conference on Network and Service Management (CNSM)*, páginas 1–5. IEEE.

Buyukates, B. e Ulukus, S. (2021). Timely Communication in Federated Learning. Em *International Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, páginas 1–6. IEEE.

Chatzoulis, D. et al. (2023). 5G V2X Performance Comparison for Different Channel Coding Schemes and Propagation Models. *Sensors*, 23(5):2436.

Committee, S. S. J. S. I. D. et al. (2016). Dedicated Short Range Communications (DSRC) Message set Dictionary. *SAE International*.

De Souza, L. A. C., Camilo, G. F., Rebello, G. A. F., Guimaraes, L. C., Campista, M. E. M. e Costa, L. H. M. K. (2024). Blockchain-based Approaches for Secure Federated Learning. Em *2024 6th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*, páginas 1–4. IEEE.

de Souza, L. A. C., Camilo, G. F., Rebello, G. A. F., Sammarco, M., Campista, M. E. M. e Costa, L. H. M. (2024). ATHENA-FL: Avoiding Statistical Heterogeneity with One-versus-All in Federated Learning. *Journal of Internet Services and Applications*, 15(1):273–288.

do Couto Teixeira, D., Almeida, J. M. e Viana, A. C. (2021). On Estimating the Predictability of Human Mobility: The Role of Routine. *EPJ Data Science*, 10(1):49.

ETSI (2014a). Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. *ETSI*.

ETSI (2014b). Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service . *ETSI*.

Fittipaldi, G., Couto, R. S. e Costa, L. H. (2025). Exploring Traffic Pattern Variability in Vehicular Federated Learning. *Computer Communications*, página 108279.

Gong, B., Xing, T., Liu, Z., Xi, W. e Chen, X. (2022). Adaptive Client Clustering for Efficient Federated Learning over Non-IID and Imbalanced Data. *Transactions on Big Data*.

Gonzalez, M. C., Hidalgo, C. A. e Barabasi, A.-L. (2008). Understanding Individual Human Mobility Patterns. *Nature*, 453(7196):779–782.

Guimaraes, L. C., Rebello, G. A. F., Camilo, G. F., de Souza, L. A. C. e Duarte, O. C. M. (2022). A Threat Monitoring System for Intelligent Data Analytics of Network Traffic. *Annals of Telecommunications*, 77(7):539–554.

Kamel, J., Wolf, M., Van Der Hei, R. W., Kaiser, A., Urien, P. e Kargl, F. (2020). VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. Em *International Conference on Communications (ICC)*, páginas 1–6. IEEE.

Korba, A. A. et al. (2023). Federated Learning for Zero-Day Attack Detection in 5G and Beyond V2X Networks. Em *International Conference on Communications (ICC)*. IEEE.

Luo, B., Xiao, W., Wang, S., Huang, J. e Tassiulas, L. (2022). Tackling System and Statistical Heterogeneity for Federated Learning with Adaptive Client Sampling. Em *Conference on Computer Communications (INFOCOM)*, páginas 1739–1748. IEEE.

McMahan, B. et al. (2017). Communication-efficient Learning of Deep Networks from Decentralized Data. *Artificial Intelligence and Statistics*, páginas 1273–1282.

Neto, H. N. C., Hribar, J., Dusparic, I., Fernandes, N. C. e Mattos, D. M. (2024). FedSBS: Federated-Learning Participant-Selection Method for Intrusion Detection Systems. *Computer Networks*, 244:110351.

Owen, S. H. e Daskin, M. S. (1998). Strategic Facility Location: A Review. *European Journal of Operational Research*, 111(3):423–447.

Patanè, R., Achir, N., Araldo, A. e Boukhatem, L. (2024). Can Vehicular Cloud Replace Edge Computing? Em *Wireless Communications and Networking Conference (WCNC)*. IEEE.

Qi, P., Chiaro, D., Guzzo, A., Ianni, M., Fortino, G. e Piccialli, F. (2024). Model Aggregation Techniques in Federated Learning: A Comprehensive Survey. *Future Generation Computer Systems*, 150:272–293.

Su, D. et al. (2024). Communication Cost-Aware Client Selection in Online Federated Learning: A Lyapunov Approach. *Computer Networks*, página 110517.

Van Der Heijden, R. W., Lukaseder, T. e Kargl, F. (2018). VeReMi: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. Em *Security and Privacy in Communication Networks (SecureComm)*, páginas 318–337. Springer.

Vinita, L. J. e Vetriselvi, V. (2023). Federated Learning-based Misbehaviour Detection on an Emergency Message Dissemination Scenario for the 6G-enabled Internet of Vehicles. *Ad Hoc Networks*, 144:103153.

Yakan, H., Fajjari, I., Aitsaadi, N. e Adjih, C. (2023). Federated Learning for V2X Misbehavior Detection System in 5G Edge Networks. Em *Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, páginas 155–163. ACM.

Zhong, Y. et al. (2023). Sybil Attack Detection in VANETs: An LSTM-Based BiGAN Approach. Em *Data Security and Privacy Protection (DSPP)*, páginas 113–120. IEEE.

Zhu, Q. et al. (2021). 3GPP TR 38.901 Channel Model. Em *the wiley 5G Ref: the essential 5G reference online*, páginas 1–35. Wiley Press Hoboken, NJ, USA.