

Universidade de Brasília

Instituto de Ciências Exatas  
Departamento de Ciência da Computação

**RADAR: um mecanismo de predição de fluxo de  
tráfego de veículos em vias urbanas com base em  
Aprendizado Federado**

Oswaldo Henrique Alves Mendonça

Monografia apresentada como requisito parcial  
para conclusão do Curso de Engenharia da Computação

Orientador

Prof. Dr. Geraldo P. Rocha Filho

Brasília  
2022



# Dedicatória

*Ao meu pai, Geraldo Mendonça dos Reis, por sempre ter me apoiado educacionalmente.  
À minha mãe, Luciana Alves Mendonça, por todo carinho dado em momentos difíceis.  
À minha irmã, Letícia Alves Mendonça, por sempre ter me aturado.*

- Oswaldo Henrique Alves Mendonça

# Agradecimentos

Agradeço ao Prof. Dr. Geraldo P. Rocha Filho por me recomendar o tema dessa pesquisa e me direcionar na realização da mesma, estando sempre presente para me auxiliar caso algo seja necessário.

Agradeço aos meus familiares, que me apoiaram e tem me apoiado ao longo de toda essa jornada.

Gostaria também de agradecer a todos amigos e parentes que de uma forma ou de outra ajudaram no desenvolvimento dessa pesquisa.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

# Resumo

A elaboração de mecanismos de predição de tráfego veicular é extremamente importante para o desenvolvimento de malhas urbanas inteligentes. No entanto, devido a grande quantidade de fatores que podem afetar as condições de uma via, essa continua sendo uma atividade complicada. De modo geral, a literatura em torno do tema costuma empregar modelos centralizados, nos quais, para a construção do modelo de aprendizado, são reunidos todos os dados disponíveis em um único local. Porém, cidades normalmente possuem múltiplos dispositivos capazes de coletar dados de tráfego e que poderiam ser utilizados nessa tarefa. Nesse contexto, esta pesquisa apresenta o RADAR: um mecanismo de predição de fluxo de tráfego de veículos em vias urbanas com base em Aprendizado Federado. O RADAR é composto por uma rede de sensores eletrônicos que captam seus próprios dados e treinam um modelo de predição local. Esses modelos locais são agregados por um servidor central, que coordena cada sensor na realização do processo. Dessa agregação é formado um modelo global, sem que os sensores tenham que compartilhar seus dados locais uns com os outros ou com o servidor central. De modo a avaliar o RADAR, foram realizados dois grupos de experimentos: (i) experimentos envolvendo a variação do número de participantes na rede; e (ii) experimentos envolvendo a variação do tipo de dia da semana, i.e. final de semana ou meio de semana. Como base de comparação, também foram avaliados modelos centralizados utilizando os mesmos sensores. Os resultados obtidos mostraram que o RADAR consegue ser competitivo com modelos tradicionais e realizar uma predição satisfatória do fluxo de veículos.

**Palavras-chave:** predição de fluxo de tráfego, aprendizado federado, séries temporais, long short-term memory

# Abstract

*The development of mechanisms for vehicular traffic flow prediction is critical to the advancement of intelligent urban infrastructure. However, this continues to be a complicated task due to the many factors that can affect the circumstances in a roadway. Generally, literature on the subject tends to focus on centralized models, in which the predictive model is built at a single server where all of the available data is gathered. Yet, cities typically have multiple devices able to collect traffic flow data that could be used on these tasks. Hence, this work presents RADAR: a vehicular traffic flow prediction mechanism in urban settings based on Federated Learning. RADAR is composed of a network of sensors that collect their own data and build a local prediction model. These local models are then aggregated by a central server, which coordinates each sensor through the process. From this aggregation a global model is built, without any local traffic data being exchanged between the sensors or with the central server. To validate the RADAR model, two groups of experiments were used: (i) experiments involving the variation of the number of participants in the network; and (ii) experiments involving the variation of the type of day the model is used on, i.e. weekend or weekdays. As a basis of comparison, centralized models with the same sensors were also evaluated. The results showed that the RADAR model is able to compete with the centralized models and perform an adequate prediction of the vehicular flow.*

**Keywords:** *traffic flow prediction, federated learning, time series, long short-term memory*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	3
1.2	Contribuição . . . . .	3
1.3	Estrutura do Trabalho . . . . .	4
<b>2</b>	<b>Fundamentação Teórica</b>	<b>5</b>
2.1	Fluxo de Tráfego . . . . .	5
2.2	Aprendizado de Máquina . . . . .	6
2.2.1	Aprendizado Profundo . . . . .	7
2.2.2	Redes Neurais Recorrentes . . . . .	8
2.2.3	Long Short-Term Memory . . . . .	8
2.3	Federated Learning . . . . .	9
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>10</b>
3.1	Predição de Fluxo de Tráfego com Modelos Centralizados . . . . .	10
3.2	Aplicação de Técnicas de FL em Redes Veiculares Urbanas . . . . .	12
3.3	Considerações Finais . . . . .	13
<b>4</b>	<b>Desenvolvimento</b>	<b>15</b>
4.1	Cenário . . . . .	15
4.2	<i>Federated Traffic Flow Prediction Mechanism</i> (RADAR) . . . . .	16
4.3	Base de Dados . . . . .	20
4.3.1	Pré-processamento dos dados . . . . .	22
4.4	Considerações Finais . . . . .	25
<b>5</b>	<b>Resultados Experimentais</b>	<b>26</b>
5.1	Configuração dos Experimentos . . . . .	26
5.1.1	Definição dos Cenários . . . . .	28
5.1.2	Escolha de parâmetros e hiper-parâmetros . . . . .	29
5.2	Avaliação da capacidade do RADAR em estimar o fluxo de tráfego . . . . .	30

5.3	Análise do desempenho do RADAR em diferentes cenários de densidade . .	35
5.4	Considerações Finais . . . . .	38
<b>6</b>	<b>Conclusão</b>	<b>39</b>
6.1	Trabalhos Futuros . . . . .	40

# Lista de Figuras

4.1	Cenário de funcionamento do RADAR. . . . .	17
4.2	Funcionamento do processo do FL no RADAR com um servidor e $N$ clientes. . . . .	18
4.3	Processo de treinamento do RADAR . . . . .	19
4.4	Localização dos sensores. . . . .	22
4.5	Fluxo de veículos ao longo da semana para o sensor <i>RSI132</i> . . . . .	24
5.1	Utilização de <i>Walk-Forward Validation</i> nos experimentos do RADAR . . . . .	28
5.2	Variação dos melhores resultados obtidos em relação a quantidade de cli- entes utilizados . . . . .	32
5.3	Loss médio ao longo das rodadas de treinamento . . . . .	34
5.4	Fluxo observado vs. predição em um dos experimentos . . . . .	35
5.5	Variação dos melhores resultados obtidos em relação a quantidade de cli- entes utilizados . . . . .	36
5.6	Variação dos melhores resultados obtidos em relação ao tipo de dia da semana . . . . .	37

# Lista de Tabelas

3.1	Comparação dos trabalhos relacionados em relação ao RADAR. . . . .	14
4.1	Amostra de dados coletados pelos sistemas do Detran-DF. . . . .	21
4.2	Descrição do conjunto de dados . . . . .	23
4.3	Número de registros por sensor . . . . .	23
5.1	Conjunto de parâmetros do processo de hiper-parametrização e Min. Loss encontrado . . . . .	30

# Lista de Abreviaturas e Siglas

**AM** Aprendizado de Máquina.

**AP** Aprendizado Profundo.

**CNN** *Convolutional Neural Network.*

**FL** *Federated Learning.*

**GCN** *Graph Convolutional Network.*

**GNN** *Graph Neural Network.*

**GRU** *Gated Recurrent Unit.*

**IA** Inteligência Artificial.

**KNN** *k-nearest neighbor.*

**LSTM** *Long Short-Term Memory.*

**MAE** *Mean Absolute Error.*

**MSE** *Mean Squared Error.*

**RADAR** *Federated Traffic Flow Prediction Mechanism.*

**RMSE** *Root Mean Squared Error.*

**RNAs** Redes Neurais Artificiais.

**RNN** *Recurrent Neural Network.*

**RNs** Redes Neurais.

**RSUs** *Roadside Units.*

**TCN** *Temporal Convolutional Network.*

# Capítulo 1

## Introdução

Com o avanço da urbanização no mundo moderno, o desenvolvimento de mecanismos inteligentes que atuem em problemas de mobilidade urbana se torna cada vez mais importante [1, 2, 3, 4, 5, 6]. Dentre os problemas de mobilidade urbana, destacam-se o monitoramento e a predição do fluxo de veículos em vias urbanas, uma vez que são essenciais para promover uma urbanização mais eficiente e sustentável [1, 5, 7]. Para tanto, essas predições podem ser utilizadas tanto por agentes que se empenham em gerenciar o trânsito quanto por usuários da via que buscam descobrir rotas mais eficientes, seja de modo a evitar congestionamentos [8], chegar ao seu destino mais rapidamente [9] ou reduzir a emissão de poluentes [10].

Diante deste contexto, métodos inteligentes, tais como aprendizado de máquina e meta-heurísticas, estão sendo cada vez mais explorados na predição do fluxo de tráfego veicular [5, 7, 11, 12, 13]. Dentre esses métodos, técnicas de aprendizado profundo, como Redes Neurais (RNs), têm ganhado espaço devido a sua capacidade de modelar a natureza não-linear e estocástica das informações de tráfego [5, 11]. Contudo, dados do fluxo de veículos costumam ser bastante particulares aos padrões de uma via, o que implica em uma falta de generalidade para os modelos treinados [11]. Além disso, essas técnicas costumam ser empregadas de modo centralizado, com todos os dados disponíveis reunidos em um único local para a construção do modelo de aprendizado. Essa centralização muitas vezes cria problemas de escalabilidade e eleva o tempo de tomada de decisão, fatores que podem tornar o sistema inviável na prática [11].

Por outro lado, o recente advento e evolução do conceito de *Federated Learning* (FL) [14, 15] apresenta um caminho promissor que pode ser explorado nos problemas de predição de fluxo de veículos em vias urbanas. Em modelos FLs, múltiplos dispositivos clientes utilizam os seus dados locais para construir um modelo global de forma descentralizada, sem que os dados locais sejam compartilhados. Para isso, cada cliente treina o seu próprio modelo local e compartilha apenas atualizações dos seus parâmetros, de

modo que eles sejam agregados no modelo global [14]. Assim, como cidades normalmente possuem múltiplos dispositivos capazes de coletar dados de tráfego, temos a hipótese que a aplicação de um modelo FL no problema de predição de fluxo em vias urbanas seria viável e poderia aliviar os problemas encontrados nos modelos centralizados tradicionais. Entretanto, para alcançar a sua descentralização, modelos FLs também possuem alguns *trade-offs* com relação aos modelos centralizados [15, 16, 17]. Primeiramente, o desempenho de um modelo FL não costuma ser tão bom quanto o de modelos centralizados, uma vez que não é possível otimizar agressivamente a predição para uma via específica. Além disso, a elaboração de um modelo FL em si apresenta os seus próprios desafios, principalmente pela falta de um corpo mais maduro de técnicas federadas, já que se trata de um conceito relativamente recente [14].

A literatura em torno do problema da predição do fluxo de veículos costuma explorar modelos centralizados. Lv, Hong, Chen, Chen, Zhu e Ji [18], Ali, Zhu e Zakarya [19] e Zheng, Lin, Feng e Chen [20] propõem modelos híbridos, com uma rede convolucional para modelar dependências espaciais e uma rede recorrente para modelar dependências temporais. Todavia, esses modelos híbridos possuem limitações com relação a sua compreensibilidade, generalidade e escalabilidade. Para sanar os problemas dos modelos híbridos, Chen, Li, Teo, Zou, Li e Zeng [21], Luo, Li, Yang e Zhang [22], Zhao, Gao, Ji, Wan, Ye e Bai [23] e Essien, Petrounias, Sampaio e Sampaio [24] atacam diferentes pontos. De modo a resolver o problema da compreensibilidade, Chen, Li, Teo, Zou, Li e Zeng [21] sugerem um modelo baseado em uma rede convolucional para modelar tanto a dependência espacial quanto a temporal enquanto Luo, Li, Yang e Zhang [22] investigam o agrupamento de estações similares para o tratamento da dependência espacial. Porém os dois trabalhos ainda deixam em aberto a questão da escalabilidade. Zhao, Gao, Ji, Wan, Ye e Bai [23], por sua vez, examinam o uso de uma rede convolucional para modelar a dependência temporal, tornando a rede menos complicada e aumentando a escalabilidade do sistema. Entretanto, esse resultado vem ao custo da generalidade do modelo, que apresenta dificuldades em condições atípicas de tráfego. Por outro lado, Essien, Petrounias, Sampaio e Sampaio [24] exploram o uso de informações mineradas de redes sociais para complementar os dados utilizados na predição, o que melhora o desempenho do modelo em condições atípicas de tráfego porém dificulta sua escalabilidade e generalidade para áreas geograficamente grandes.

Por ser um campo mais recente, há poucos trabalhos que exploram FL em redes veiculares. Não obstante, trabalhos como os de Liu, Yu, Kang, Niyato e Zhang [25], Qi, Hossain, Nie e Li [26] e Silva, Bittencourt e Rivera [27] já buscam aplicar técnicas de FL ao problema de predição de fluxo de tráfego. Liu, Yu, Kang, Niyato e Zhang [25] dão um enfoque maior com relação aos aspectos de privacidade do modelo, enquanto Qi, Hossain,

Nie e Li [26] tratam, além da privacidade, também sobre aspectos da segurança do modelo federado contra ataques de entidades maliciosas, com ambos utilizando uma GRU como modelo local. Por outro lado, Silva, Bittencourt e Rivera [27] focam nos aspectos de rede do modelo FL em um contexto de estimativa de tráfego de veículos utilizando computação de borda. Porém, não é dada muita atenção ao *multilayer perceptron* local utilizado e nem é realizada uma comparação extensiva com modelos centralizados. Finalmente, Zhang, Zhang, Yu e Yu [28] investigam a proteção da informação topológica da rede de transporte no contexto de predição de velocidade de tráfego utilizando uma GNN, novamente dando um enfoque com relação aos aspectos de privacidade que o uso de FL pode oferecer. Portanto, diferentemente desta pesquisa, a maioria dos trabalhos que exploram FLs em redes veiculares focam em campos específicos como segurança, privacidade e aspectos de rede.

## 1.1 Objetivos

Nesta pesquisa é apresentado o *Federated Traffic Flow Prediction Mechanism* (RADAR), um mecanismo de predição federado de fluxo de tráfego de veículos em vias urbanas com base em *Federated Learning*. O RADAR é composto por uma rede de sensores eletrônicos que captam seus próprios dados e treinam um modelo de predição local. Esses modelos locais são agregados por um servidor central, que coordena cada sensor na realização do processo. Dessa agregação é formado um modelo global, sem que os sensores tenham que compartilhar seus dados uns com os outros ou com o servidor central, que recebe apenas os modelos treinados. Desse modo espera-se desenvolver e validar um modelo que possa ser utilizado como uma alternativa aos modelos centralizados na predição de fluxo de tráfego.

## 1.2 Contribuição

As contribuições desta pesquisa são listadas a seguir:

1. Investiga o uso de uma LSTM no contexto de predição de fluxo de tráfego de veículos em vias urbanas com base em *Federated Learning*.
2. Apresenta uma predição satisfatória utilizando poucos recursos computacionais e com menos rodadas de treinamento do que o usual em modelos *Federated Learning*.
3. Avalia o modelo em diferentes contextos de densidade de tráfego com relação ao tipo de dia da semana, *i.e.* em dias de Final de Semana ou Meio de Semana.

## 1.3 Estrutura do Trabalho

A construção da pesquisa está estruturada da seguinte maneira. O Capítulo 2 fundamenta um corpo teórico para o entendimento dos temas e conceitos relacionados ao RADAR. Em seguida, no Capítulo 3 são examinados trabalhos relacionados aos campos de predição de fluxo de tráfego e FL, expondo as vantagens e limitações dos mesmos. Posteriormente, o Capítulo 4 desenvolve o RADAR como um mecanismo de predição federado de fluxo de tráfego em vias urbanas e apresenta a modelagem dos seus cenários. Com isso, no Capítulo 5 o RADAR é validado, apresentando a metodologia dos experimentos e os resultados obtidos. Finalmente, o Capítulo 6 discorre sobre algumas considerações finais e possíveis avanços que poderiam ser investigados em trabalhos futuros.

# Capítulo 2

## Fundamentação Teórica

Este capítulo apresenta conceitos teóricos relacionados ao RADAR, um mecanismo de predição federado de fluxo de tráfego de veículos em vias urbanas com base em *Federated Learning*. Primeiramente, são descritos conceitos básicos de fluxo de tráfego. Em seguida, apresenta-se conceitos de aprendizado de máquina. Por fim, o conceito de *Federated Learning* é introduzido.

### 2.1 Fluxo de Tráfego

O crescimento das cidades ao longo dos anos trouxe vários desafios em questões de mobilidade urbana [1, 5]. Para que as cidades possam responder de forma inteligente a esses desafios, é crucial que o monitoramento e a predição de informações do fluxo de tráfego sejam realizados de modo amplo e satisfatório [11, 5]. Sabendo o fluxo de tráfego das vias, é possível atuar na prevenção de situações indesejadas no trânsito, como congestionamentos e acidentes [1, 5].

De modo geral, o fluxo de uma via é definido pela quantidade de veículos que ali trafegam ao longo do tempo [29]. Assim, há uma relação clara entre o fluxo de uma via, a sua densidade, *i.e.* quantidade de veículos ao longo do espaço, e a velocidade média dos veículos que ali circulam. Sendo  $n$  a quantidade de veículos que trafegam por um ponto de referência em um intervalo de  $t$  segundos com uma velocidade média  $v$ , o fluxo de veículos  $q$  e a densidade de veículos  $k$  podem ser dadas por:

$$q = \frac{n}{t}, \quad k = \frac{q}{v}$$

Portanto, o problema de predição do fluxo de tráfego é um problema de regressão sobre uma série temporal de valores de  $q$  [11]. A previsibilidade dessa série temporal está ligada à periodicidade diária e semanal do fluxo de veículos [5]. Padrões típicos de picos

no tráfego durante os dias da semana costumam estar relacionados com o início e o fim das horas úteis do dia, enquanto o período da madrugada costuma registrar os menores valores. Além disso, são observados também diferentes padrões em dias de semana em comparação aos dias de final de semana. Por outro lado, diversos fatores externos à via podem aumentar a imprevisibilidade do fluxo, como fatores climáticos, eventos e acidentes [5]. Ainda, o horizonte de tempo no qual se tenta fazer a previsão influencia a precisão dos resultados, com o fluxo se tornando mais imprevisível ao se tentar prever valores mais distantes no futuro [5].

Desse modo, o fluxo de tráfego é influenciado por três componentes. Primeiramente, temos um componente espacial, relacionado ao lugar onde a via se encontra. Isso é dizer que, por exemplo, uma avenida no centro da cidade irá possuir um fluxo de veículos diferente de uma rua residencial em uma região mais distante, ou seja, vias diferentes apresentam padrões de fluxo diferentes em virtude das suas localizações. Em seguida, temos um componente temporal, relacionado ao horário no qual o fluxo é medido, conforme os padrões já apresentados. Por fim, há também influências de um componente externo, como eventos, feriados, condições climáticas, acidentes e realização de obras na via.

## 2.2 Aprendizado de Máquina

A evolução na complexidade dos problemas que a computação tenta resolver no mundo moderno tem tornado inviável a construção de soluções algorítmicas em diversas situações [30]. Ainda assim, como a construção de sistemas de decisão para resolver esses problemas é imprescindível, o campo de Aprendizado de Máquina (AM) apresenta, em contraste aos modelos algorítmicos, uma alternativa que vem ganhando espaço nos últimos anos [30, 31]. AM é uma sub-área do campo de Inteligência Artificial (IA) focada no desenvolvimento de sistemas que sejam capazes de aprender com suas próprias experiências [31, 32]. Portanto, um modelo AM é um modelo que não é especificamente programado para agir de dada maneira em dada situação, mas que aprende por meio de dados de treinamento qual a maneira correta de agir naquela situação [33].

Nesse contexto, modelos AMs são modelos que se adaptam ao comportamento dos dados de treinamento. Portanto, é dito que há um aprendizado à medida que o modelo consegue absorver padrões dos dados de treinamento e utilizá-los para fazer inferências sobre outras situações [34]. Assim, modelos AMs apresentam como vantagem o fato de serem capazes de resolver um problema sem que o programador saiba explicitamente como resolvê-lo [30, 33]. Esse fato tem feito o paradigma de AM ganhar relevância na construção de modelos para problemas complexos do mundo real nos quais é possível encontrar uma quantidade de dados razoáveis disponíveis para treinamento [31].

Como modelos AMs são caracterizados por aprenderem com dados de treinamento, eles podem ser categorizados em diferentes tipos com relação às características dos dados de treinamento e a maneira com que o modelo aprende, sendo eles [30, 31]:

- Aprendizado supervisionado, no qual existem dados de treinamento previamente rotulados, *i.e.*, sabe-se o valor correto de saída tendo esses dados como entrada para o modelo.
- Aprendizado não supervisionado, no qual se quer extrair alguma informação dos dados, mas não se sabe de antemão qual é a resposta correta para os dados de treinamento.
- Aprendizado semi-supervisionado, no qual há tanto dados rotulados quanto dados não rotulados e portanto utiliza-se uma hibridização de técnicas supervisionadas e não supervisionadas.
- Aprendizado por reforço, no qual o sistema é reforçado ou penalizado de acordo com as ações tomadas no ambiente.

Nesta pesquisa, a predição do fluxo de tráfego se trata de um problema de aprendizado supervisionado. Portanto, temos um conjunto de dados do fluxo de tráfego da via. Esses valores são usados como dados de treinamento para o modelo AM. Ou seja, utiliza-se o fluxo de tráfego do passado para tentar prever qual será o seu valor no futuro. Com isso, queremos construir um estimador do fluxo de tráfego que permita a sua predição.

### 2.2.1 Aprendizado Profundo

A proliferação de tecnologias de coleta de dados ao longo da década passada criou um ambiente propício para o desenvolvimento de modelos AMs [30, 35]. Desse modo, o conceito de Aprendizado Profundo (AP) surge dentro da área de AM com a intenção de promover um melhor reconhecimento de padrões em grandes conjuntos de dados. [36]. Para tanto, um modelo AP utiliza-se de múltiplas camadas conectadas de modo a se obter uma representação desejada do conhecimento [36, 37, 38].

Portanto, um modelo AP é dito profundo devido ao seu uso de múltiplos níveis para abstrair uma representação mais complexa [36]. O uso de múltiplos níveis permite que o modelo AP, mesmo sendo alimentado com dados crus, descubra uma melhor representação dos dados para a tarefa de aprendizado [36, 37]. Evita-se assim um processo de engenharia de características que seria inviável de ser realizada manualmente em tarefas mais complexas.

De modo geral, um modelo AP possui uma rede que passa sequencialmente por três tipos de camadas: (i) uma camada de entrada, responsável por receber os dados de entrada;

(ii) um conjunto de camadas escondidas, responsáveis pelo processo de aprendizagem; e (iii) uma camada de saída, responsável por produzir o resultado final [37]. Assim, as camadas escondidas fazem um mapeamento entre os valores de entrada e saída [37]. Ou seja, tem-se que as camadas escondidas são responsáveis por construir uma função não linear que faça com que os valores de entrada levem aos valores de saída corretos [36].

As diferentes camadas do modelo AP são conectadas por meio de funções de ativação [36]. A função de ativação permite o estabelecimento de relações não-lineares entre a entrada e a saída [38]. Com esse intuito, a função de ativação define se um dado nó será utilizado ou não na próxima camada [37]. Ou seja, a função de ativação está intrinsecamente ligada ao processo de descoberta da melhor representação dos dados. Ao desligar certos nós, altera-se a dimensionalidade de certas características no problema, o que pode levar a melhores resultados [38].

### 2.2.2 Redes Neurais Recorrentes

Dentro do contexto de AP, o uso de Redes Neurais Artificiais (RNAs) é muito comum devido à sua estrutura flexível [37]. Em particular quando se trata de dados sequenciais, uma classe de RNAs chamada *Recurrent Neural Network* (RNN) costuma ser a opção preferencial [36]. RNNs possuem um componente recorrente que permite a assimilação de padrões sequenciais e a retenção de conhecimento ao longo da sequência de dados [39].

Com esse objetivo, RNNs mantêm em suas camadas escondidas um vetor de estado com um histórico de informações sobre os dados já processados da sequência [36]. Isso é feito por meio de um processo de retropropagação da saída da camada, formando um ciclo que permite que a RNN tenha uma memória própria [35]. Além disso, esse desenho de rede permite que RNNs trabalhem com sequências de entradas e sequências de saídas, algo bastante útil ao lidar com séries temporais.

Porém, junto com a retropropagação do estado da RNN ao longo das camadas há também uma retropropagação do gradiente [36]. A cada etapa, os gradientes retro-propagados decaem ou crescem, o que cria um problema durante o treinamento [36]. Em sequências de dados relativamente grandes isso leva a uma queda do desempenho devido à explosão do gradiente ou ao seu desaparecimento.

### 2.2.3 Long Short-Term Memory

De modo a lidar com os problemas encontrados em RNNs tradicionais, modelos do tipo *Long Short-Term Memory* (LSTM) foram criados [40]. LSTMs possuem um mecanismo de portões responsáveis por regular o fluxo de informação ao longo das camadas [41, 42]. Desse modo, evita-se o decaimento ou crescimento do gradiente, já que o sistema

de portões garante um fluxo de erros constante ao longo da propagação [40]. Assim, a rede LSTM é capaz de trabalhar com sequências de dados relativamente longas e explorar intercorrelações de longo prazo entre os dados [41]. Esses fatos ampliam a aplicabilidade do LSTM, pois permitem o tratamento de casos mais complexos [35, 41].

Via de regra, uma unidade da camada LSTM costuma ser composta por: (i) uma célula; (ii) um portão de entrada; (iii) um portão de saída; e (iv) um portão de esquecimento [35, 41]. A célula é responsável por guardar o estado da unidade, enquanto os portões controlam o que é escrito e lido na célula [35]. Portanto, o portão de entrada controla e dimensiona o sinal de entrada da rede para a célula, enquanto o portão de saída controla e protege o acesso aos conteúdos da memória da célula [42]. Por outro lado, o portão de esquecimento atua de modo a determinar quais informações devem ser removidas de estados anteriores da célula [41].

## 2.3 Federated Learning

O mundo moderno apresenta uma crescente difusão da titularidade de dados em múltiplos agentes que possuem múltiplos dispositivos [30, 43]. Muitos desses dados poderiam ser utilizados na construção de um modelo de aprendizado, porém problemas com relação a transferência desses dados para um servidor central impedem esse uso [43, 44]. Somado a isso, por parte de usuários finais há uma demanda por sistemas inteligentes que possam se adaptar a padrões cada vez mais particulares [30, 44]. Dentro desse contexto, o conceito de *Federated Learning* (FL) surge como uma alternativa para se construir um modelo de aprendizado de forma federada [14, 15].

Desse modo, um modelo FL é treinado de forma descentralizada, com múltiplos dispositivos treinando um modelo próprio com os seus dados locais [14]. Para isso, um servidor central coordena o treinamento e a agregação dos resultados em múltiplas rodadas de treinamento federado. No geral, uma rodada é composta por três etapas: (i) o servidor seleciona clientes para participar do treinamento e os envia o modelo atual; (ii) os clientes treinam o modelo recebido com seus dados locais, conforme determinado pelo servidor; e (iii) os clientes enviam os resultados dos treinamentos locais para o servidor, que agrega os resultados em um novo modelo atual [45]. Portanto, no paradigma FL múltiplos clientes treinam conjuntamente um modelo global, sem que o *dataset* inteiro dos clientes tenha que ser compartilhado com o servidor central.

# Capítulo 3

## Trabalhos Relacionados

Este capítulo apresenta trabalhos relacionados ao tema desta pesquisa, i.e. predição de fluxo de tráfego de veículos em vias urbanas com base em *Federated Learning*. A área de predição de fluxo de tráfego possui uma longa história [46]. A literatura mais atual usualmente costuma ter um enfoque em modelos tradicionais centralizados [23, 19, 20, 24, 22, 18, 21]. Por outro lado, o desenvolvimento do campo de FL é mais recente e por isso suas aplicações em redes veiculares são menos desenvolvidas. Ainda assim, já é possível encontrar trabalhos nesse sentido [47, 25, 26, 27, 28]. Portanto, esse capítulo será dividido em duas partes. Na primeira parte serão apresentados trabalhos sobre predição de fluxo de tráfego envolvendo modelos centralizados. Já na segunda parte serão apresentados trabalhos que buscam aplicar técnicas de FL em redes veiculares urbanas.

### 3.1 Predição de Fluxo de Tráfego com Modelos Centralizados

Entre os trabalhos de predição de fluxo de tráfego envolvendo modelos centralizados, Zhao, Gao, Ji, Wan, Ye e Bai [23] propõem a utilização de um mecanismo de aprendizagem profunda baseada em um modelo *Temporal Convolutional Network* (TCN) otimizado com o método de Taguchi. Para isso são coletados dados do fluxo de veículos de uma interseção em ciclos de 15 minutos, com 5 ciclos sequenciais sendo usados como entrada para a rede e tendo como valor de saída a predição do ciclo posterior aos ciclos de entrada. A principal vantagem do modelo proposto é a redução do custo computacional de treinamento, mantendo a capacidade de alcançar resultados compatíveis com o estado da arte. Contudo, é mostrado também que o modelo apresenta uma dificuldade maior de se adaptar em situações atípicas, como condições climáticas que alterem o fluxo de via, resultando em uma queda de desempenho nesses casos.

Já Ali, Zhu e Zakarya [19] apresentam uma rede neural baseada em características espaço-temporais de uma região de uma cidade que visa prever o fluxo de entrada e saída daquela região. Para tanto, a cidade é dividida em um *grid* de  $N \times M$  regiões e o fluxo de entrada e saída é representado por tensores. Além disso, são considerados também fatores temporais recentes, diários e semanais, bem como fatores externos como feriados, eventos e condições climáticas. Com isso, é montada uma arquitetura baseada no uso de uma *Graph Convolutional Network* (GCN) para capturar dependências espaciais seguida de uma LSTM para capturar dependências temporais. O modelo possui como vantagem o fato de levar em conta tanto características espaciais quanto características temporais quanto fatores externos na predição dos fluxos, além de conseguir fazer predições para todas as regiões simultaneamente. Porém, possui como desvantagem o alto custo de treinamento da rede, considerando ainda que a divisão da cidade em um *grid*  $N \times M$  pode tornar o modelo mais complicado ao se tentar escolher valores maiores de  $N$  ou  $M$ .

Zheng, Lin, Feng e Chen [20], por sua vez, procuram chegar em uma abordagem capaz de capturar a não-linearidade inerente ao problema de predição de fluxo de tráfego. Com esse objetivo, os autores introduzem um modelo híbrido combinando uma LSTM convolucional que utiliza um mecanismo de atenção com uma LSTM bidirecional. É demonstrado que o modelo consegue lidar com as dependências espaço-temporal do problema e se comportar bem em diferentes cenários. No entanto, possui como limitação o fato de não se adaptar bem a sistemas de larga escala.

Por outro lado, Essien, Petrounias, Sampaio e Sampaio [24] buscam integrar informações mineradas de redes sociais para complementar dados de tráfego e dados climáticos utilizados em problemas de predição de fluxo de tráfego. Para esse fim, os autores utilizam uma LSTM bidirecional com *Stacked Autoencoders*. Como entrada para a rede LSTM bidirecional são utilizados dados de tráfego, dados climáticos e mensagens mineradas do *Twitter*. Desse modo, o modelo possui como vantagem ser mais robusto em situações atípicas. Porém, possui como desvantagem não ser viável para áreas geograficamente grandes.

Luo, Li, Yang e Zhang [22] propõem um modelo híbrido combinando um *k-nearest neighbor* (KNN) com uma LSTM. Para isso, o KNN é usado para escolher estações similares à estação escolhida para se prever o fluxo de tráfego. A LSTM é então usada para fazer uma predição em todas as estações escolhidas e os resultados são agregados por meio de uma média ponderada com a atribuição de pesos feita via um método *rank-exponent* de modo a dar um peso maior às estações mais relevantes. Assim, o modelo tem como vantagem o fato de propor uma arquitetura relativamente simples, facilitando a compreensão dos resultados. Contudo, possui como desvantagem necessitar de dados de estações que tenham alguma similaridade em seus fluxos, *i.e.* estejam próximas.

Lv, Hong, Chen, Chen, Zhu e Ji [18] investigam o uso de um modelo GCN utilizando múltiplos grafos. Com essa finalidade, são construídos múltiplos grafos com correlações identificadas entre o tráfego de diferentes vias. Esses grafos são utilizados como entrada para GCN de modo a modelar as relações espaço-temporais do problema. Em seguida, é utilizada uma GRU para capturar dependências temporais e uma camada de saída para prever as condições de tráfego. O modelo possui como principal contribuição o fato de considerar correlações semânticas entre as diferentes vias, inclusive possivelmente entre vias distantes. No entanto, o sistema é relativamente complicado e possui a limitação de não modelar bem eventos atípicos nas vias e condições anormais de tráfego.

Finalmente, Chen, Li, Teo, Zou, Li e Zeng [21] apresentam uma rede visando modelar as relações espaciais e temporais do fluxo de tráfego simultaneamente. Para tanto, baseado nas relações espaço-temporais do tráfego de veículos, é proposta uma CNN que possui um mecanismo de portões inspirado nos encontrados em redes LSTM. Com isso, esses portões controlam os atributos espaço-temporais propagados pela hierarquia das camadas da CNN. A cidade é então particionada em um *grid* de  $N \times M$  regiões e um *framework* é montado em cima do modelo proposto para prever o fluxo de entrada e saída das regiões. O modelo possui como contribuição o fato de utilizar apenas a CNN proposta para modelar as dependências espaço-temporais, evitando parte da impraticidade de esquemas comumente encontrados que utilizam uma CNN para modelar dependências espaciais e uma RNN para modelar dependências temporais. Além disso, o fato de explorar dependências espaciais e temporais conjuntamente permite uma melhor modelagem das correlações entre essas duas dependências. Porém, o sistema ainda possui limitações quanto a sua escalabilidade em regiões geograficamente grandes por dividir a cidade em um *grid*  $N \times M$ .

## 3.2 Aplicação de Técnicas de FL em Redes Veiculares Urbanas

Entre os trabalhos que buscam aplicar técnicas de FL em redes veiculares urbanas, Liu, Yu, Kang, Niyato e Zhang [25] apresentam um modelo de predição de fluxo de tráfego baseado em uma GRU federada. Um modelo global é distribuído aos participantes e treinado por eles com seus dados locais. Cada participante então envia os seus resultados para um servidor que agrega os modelos locais em um novo modelo global. O modelo possui como principal vantagem o fato de manter a privacidade dos participantes. Por outro lado, possui como limitação o *overhead* de comunicação presente no modelo.

Qi, Hossain, Nie e Li [26] sugerem um modelo federado de predição de fluxo de tráfego com base em uma rede *blockchain*. Ao invés de utilizar um servidor central para coor-

nar os clientes e agregar o modelo, os clientes são coordenados de maneira descentralizada e tem seus resultados verificados por mineradores na *blockchain* proposta. Desse modo, o modelo possui a vantagem de não possuir um ponto único de falha, como acontece em modelos que utilizam um servidor central. Além disso, o fato de ser totalmente descentralizado impede eventuais abusos por parte de um servidor central malicioso. Porém, apesar de ser relativamente robusta contra clientes maliciosos, a rede possui a limitação de apresentar falhas na existência de uma determinada quantidade de mineradores maliciosos.

Silva, Bittencourt e Rivera [27] investigam uma arquitetura de estimação de tráfego em tempo real utilizando FL e computação de borda. Com esse objetivo, uma arquitetura é montada na qual há uma nuvem atuando como agregador central, um conjunto de *Roadside Units* (RSUs) atuando como agregadores intermediários e veículos atuando na borda coletando e processando seus dados locais. A principal contribuição encontrada é a de demonstrar a viabilidade da utilização de FL no problema de estimação de tráfego em tempo real. Contudo, a pesquisa possui como limitação o fato de não realizar uma comparação significativa com modelos centralizados.

Por fim, Zhang, Zhang, Yu e Yu [28] propõem uma arquitetura capaz de proteger a informação topológica da rede de transporte no contexto de predição de velocidade de tráfego utilizando FL por meio de um modelo baseado em uma *Graph Neural Network* (GNN). Para tanto, cada organização participante da rede federada treina uma GNN utilizando os seus dados do tráfego local e a informação topológica da sua rede. Os participantes então enviam os resultados para um servidor em nuvem que agrega os resultados por meio de uma abordagem de privacidade diferencial. Assim, o sistema apresenta a vantagem de manter a informação topológica da GNN protegida. No entanto, o modelo possui como desvantagem o alto custo de treinamento da rede.

### 3.3 Considerações Finais

Conforme supracitado, é possível encontrar vários desenvolvimentos recentes na área de predição de fluxo de tráfego de veículos utilizando modelos centralizados, o que evidencia a importância desta pesquisa. Por outro lado, devido ao fato de se tratar de um campo relativamente novo, a aplicação de métodos de FL nessa área não é uma prática tão difundida. Ainda que já seja possível encontrar trabalhos que utilizem modelos federados em redes veiculares de modo geral, eles costumam ter um enfoque diferente do enfoque presente nessa pesquisa, seja por não buscar a predição de fluxo de veículos ou por se concentrar em outras características de modelos FL, como por exemplo, a segurança e a privacidade dos dados.

A Tabela 3.1 apresenta uma comparação dos trabalhos analisados com esta pesquisa. No caso de um trabalho não conter um dos tópicos, o respectivo campo estará em branco. Já no caso do trabalho abordar um dos tópicos, o respectivo campo estará marcado. A maioria dos trabalhos aborda predição de fluxo de tráfego, porém não realiza uma avaliação em diferentes dias da semana. Além disso, entre os trabalhos que abordam FL, nenhum utiliza uma LSTM.

Tabela 3.1: Comparação dos trabalhos relacionados em relação ao RADAR.

<b>Trabalho Relacionado</b>	<b>Predição do Fluxo de Tráfego</b>	<b>Avaliação em Diferentes Dias da Semana</b>	<b>FL</b>	<b>LSTM</b>
Zhao <i>et al.</i> [23]	✓			
Ali <i>et al.</i> [19]	✓			✓
Zheng <i>et al.</i> [20]	✓			✓
Essien <i>et al.</i> [24]	✓			✓
Luo <i>et al.</i> [22]	✓			✓
Lv <i>et al.</i> [18]	✓	✓		
Chen <i>et al.</i> [21]	✓			
Liu <i>et al.</i> [25]	✓		✓	
Qi <i>et al.</i> [26]	✓		✓	
Silva <i>et al.</i> [27]	✓		✓	
Zhang <i>et al.</i> [28]			✓	
RADAR	✓	✓	✓	✓

# Capítulo 4

## Desenvolvimento

Este capítulo apresenta o RADAR, um mecanismo de predição federado de fluxo de tráfego de veículos em vias urbanas com base em *Federated Learning*. O RADAR é composto por uma rede de sensores eletrônicos e um servidor central que coordena cada sensor para realizar o processo de predição federado. Para tanto, cada sensor treina com seus próprios dados um modelo de predição, o qual é agregado em um modelo global pelo servidor ao final do processo. O RADAR tem como objetivo principal fornecer uma predição satisfatória em diferentes situações com um único modelo global, sendo simultaneamente capaz de manter os dados dos diferentes clientes participantes em sigilo. Primeiramente, será descrito o cenário modelado para o funcionamento do RADAR. Em seguida, como o RADAR foi modelado será apresentado. Por fim, a base de dados utilizada será caracterizada e detalhada.

### 4.1 Cenário

O cenário de funcionamento do RADAR é apresentado na Figura 4.1. No cenário modelado, há duas interseções formadas por vias de sentido duplo, com uma das vias interligando os dois cruzamentos. Em cada interseção são distribuídos quatro sensores, sendo cada sensor responsável por medir o fluxo de tráfego de um sentido de uma das vias. Para realizar a medição, cada sensor contabiliza a quantidade  $q$  de veículos que trafegam naquele sentido da via em um determinado intervalo de tempo. Além disso, é coletado o dia da semana no qual a observação foi realizada e é contabilizada a velocidade média  $v$  dos veículos. Com isso, deriva-se a densidade  $k$  do tráfego  $k = q/v$ .

O processo de funcionamento do FL no RADAR para realizar a predição federada do fluxo de veículos é apresentado na Figura 4.2. No RADAR, cada sensor funciona como um cliente em uma rede formada pelo conjunto dos sensores, definindo-se assim os participantes das rodadas do FL. De modo a orquestrar a federação, há ainda um

servidor central que é responsável por coordenar as rodadas de aprendizado e agregar os resultados dos clientes. Ao final de cada rodada federada, o servidor devolve para os clientes a atualização do modelo global resultante da agregação realizada. Esse processo requer três etapas:

1. Inicialização da tarefa: Nesta etapa, o servidor central seleciona os  $m$  clientes que participarão do FL, determina os parâmetros do processo de treinamento e distribui aos clientes selecionados o modelo global atual  $w_g^0$ , Etapa 1 Figura 4.2.
2. Treinamento dos modelos locais: Nesta etapa, os clientes  $c$  treinam o modelo global  $w_g^i$  conforme recebido pelo servidor (Etapa 2 Figura 4.2), onde  $i$  é o índice de iteração atual do modelo. Em seguida, cada  $c$  atualiza os parâmetros do modelo local com seus próprios dados locais. Neste caso, o objetivo de cada  $c$  é obter parâmetros ótimos  $w_c^i$  com base no valor mínimo da função de perda  $F(w_c^i)$ . Com os modelos locais atualizados de cada  $c$ , o servidor central recebe de volta tais modelos para realizar a agregação do modelo global, Etapa 3 Figura 4.2.
3. Agregação do modelo global: Nesta etapa, o servidor recebe dos clientes os resultados dos treinamentos locais e agrega os modelos locais dos clientes em um novo modelo global. Em seguida o modelo global atualizado  $w_g^{i+1}$  é repassado novamente para os clientes locais (Etapa 1 Figura 4.2) realizarem a próxima rodada federada para minimizar a função de perda global  $F(w_g^i)$ , isto é  $F(w_g^i) = \frac{1}{m} \sum_{c=1}^m F(w_c^i)$ .

As etapas 2 e 3 são iteradas até que  $F(w_g^i)$  obtenha uma precisão adequada. Na subseção seguinte é apresentado como o RADAR foi modelado para resolver o problema de predição de fluxo de veículos nas vias urbanas.

## 4.2 *Federated Traffic Flow Prediction Mechanism* (RADAR)

O mecanismo proposto para realizar a predição federada de tráfego de veículos nas vias urbanas foi modelado por meio de uma *Long Short Term Memory* (LSTM). A LSTM é uma variação das *Recurrent Neural Network* (RNNs) criada para lidar com o problema de dissipação do gradiente, característica presente nesta pesquisa. Esse problema ocorre pois métodos utilizados para propagar o gradiente durante o treinamento podem fazer com que os erros decaiam à medida que são propagados pelas camadas da rede, tornando-os insignificantes e prejudicando o aprendizado.

Além da motivação supracitada, a LSTM modelada no RADAR justifica-se por ser adaptar às características encontradas em séries temporais, como é o caso do problema

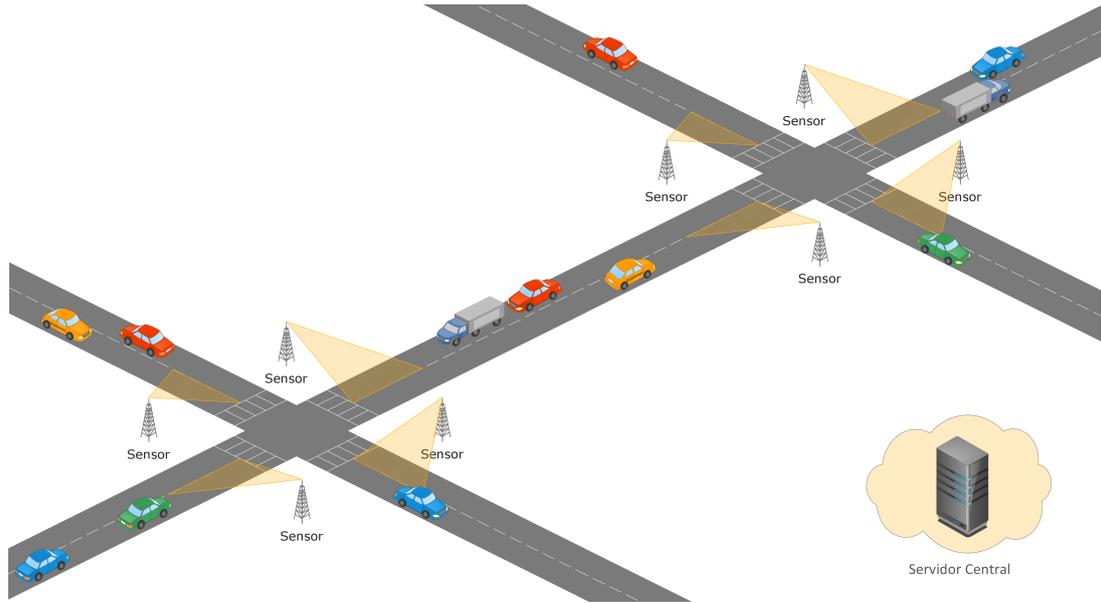


Figura 4.1: Cenário de funcionamento do RADAR.

de predição de fluxo de tráfego. Isso se deve ao fato da LSTM ser capaz de processar sequências temporais com sua memória interna, conseguindo evitar o problema de dependência de longo prazo encontrado em RNNs convencionais [40]. Esse problema ocorre pois, apesar de serem capazes de processar sequências de dados, RNNs sofrem de baixa performance quando precisam se lembrar de informações observadas no passado distante. Ao contrário das RNNs convencionais, a LSTM é capaz de explorar intercorrelações de longo prazo entre os dados (i.e., dependência nos dados), melhorando sua aplicabilidade em tais situações.

No RADAR, cada participante na federação é responsável por treinar um modelo *LSTM* local, utilizando os dados coletados pelo próprio participante. Para tanto, os clientes utilizam as observações anteriores  $R_{old}$  a um dado momento como entrada para o modelo com o intuito de prever o valor do fluxo de veículos da observação posterior  $R_{new}$ . Desse modo, as observações anteriores  $R_{old}$  formam, como entrada para a *LSTM*, uma janela temporal contendo  $n$  observações sequenciais que serão utilizadas na predição de um  $R_{new}$ . Como exemplo, se cada observação representa um intervalo de cinco minutos e é utilizada uma janela temporal de oito observações, isso significa que o modelo recebe como entrada valores referentes aos 40 minutos anteriores ao presente e possui como saída a predição do fluxo para os próximos cinco minutos. Cada cliente faz uso de 10 épocas de treinamento em seu modelo local e treina a rede usando 256 observações como tamanho do lote usado em cada iteração. Esses parâmetros foram escolhidos considerando a relação entre o custo computacional de se aumentar ou diminuir os valores e os benefícios para a aprendizagem que essa alteração proporciona. Deve ser ressaltado ainda que, no contexto

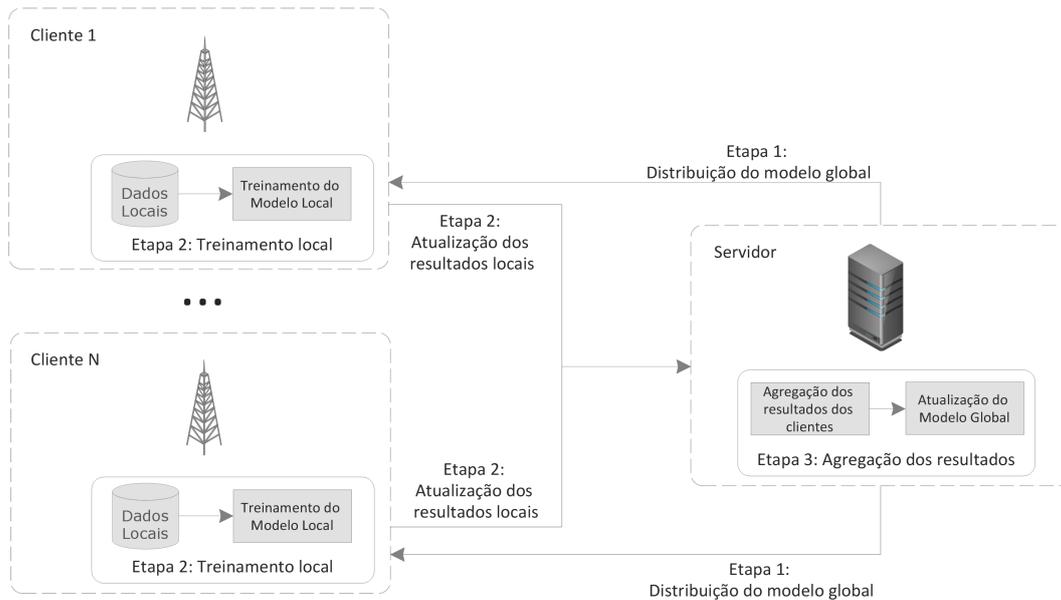


Figura 4.2: Funcionamento do processo do FL no RADAR com um servidor e  $N$  clientes.

do RADAR, os clientes terão recursos computacionais escassos, e portanto um custo computacional muito alto poderia tornar a solução inviável.

Para realizar o treinamento do mecanismo de predição federado, é necessário configurar a rede do FL na qual os clientes participarão. Para isso, é preciso modelar um modelo global de aprendizado a partir da colaboração entre diferentes clientes coordenados por um servidor central, de modo que os clientes não compartilhem seus dados com outros clientes na federação. Por consequência, cada cliente deve ser encarregado de treinar nos seus próprios dados o modelo global atual, que o foi passado pelo servidor central, e comunicar de volta ao servidor a atualização dos seus resultados para que eles possam ser agregados ao modelo global.

De modo que o servidor possa lidar com os participantes da federação, deve-se realizar uma rodada de aprendizado federado. Cada rodada federada consiste de um processo iterativo no qual o servidor: (i) escolhe os clientes que participarão da rodada; (ii) transmite o modelo global atual aos clientes; (iii) espera os clientes computarem seus novos modelos locais; (iv) recebe de volta dos clientes seus resultados locais; e (v) utiliza esses resultados em um processo de agregação para computar um novo modelo global.

A Figura 4.3 apresenta o processo de aprendizado federado do RADAR, que compartilha o modelo global de volta aos clientes. Inicialmente, os clientes coletam os dados locais para treinar o seu modelo local. Como as observações são feitas em um período fixo, os clientes devem inicialmente esperar o fim deste período de captura de dados para começar o treinamento. Uma vez tendo os dados coletados, o cliente pode receber o modelo global atualizado do servidor e realizar a etapa de treino com seus dados locais, devolvendo para

o servidor, ao final do treinamento, o resultado que deve ser agregado ao modelo global. Esse fluxo se repete enquanto o servidor estiver realizando as rodadas de treinamento e, ao final dessas, o cliente adota o modelo final para uso.

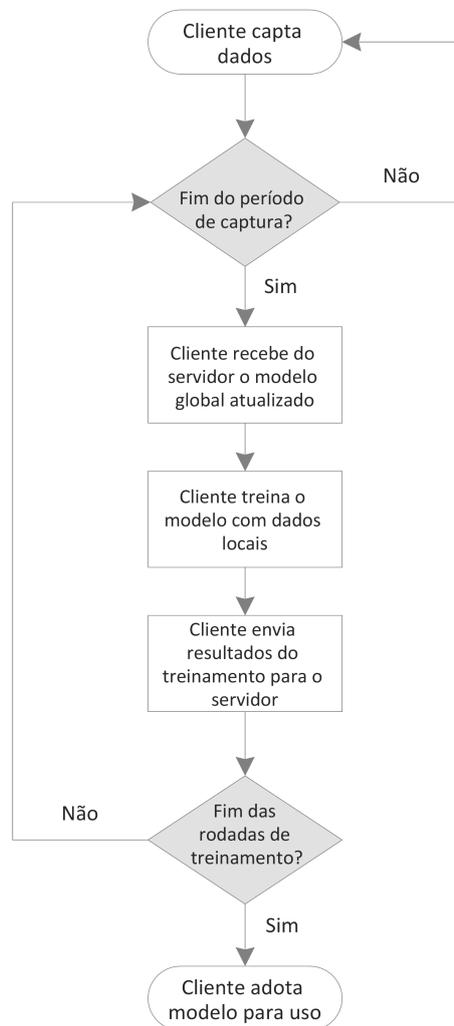


Figura 4.3: Processo de treinamento do RADAR

No Algoritmo 1 é apresentado o processo para gerar o modelo global de aprendizado federado no RADAR. Primeiramente, os pesos são inicializados sem um treinamento anterior usando o método de Glorot e Bengio [48] (Linha 2, Algoritmo 1). Em seguida são realizadas  $T$  rodadas de aprendizado federado (Linha 3, Algoritmo 1). Em cada rodada, o servidor define que o modelo será treinado em uma janela de tempo contendo observações de uma semana (Linha 4, Algoritmo 1), começando em um dia zero, igual para todos os participantes, e avançando um dia por rodada. Com isso definido, o servidor propaga os pesos do modelo global para cada cliente e recebe de volta os valores para realizar a

agregação (Linhas 5-7, Algoritmo 1). Ao final da rodada, o servidor calcula o novo modelo global (Linha 8, Algoritmo 1) com base nos resultados dos clientes.

---

**Algorithm 1** Processo de geração do modelo global no RADAR

---

```

1: procedure SERVEREXECUTION:
2:    $w_g^0 \leftarrow$  (initialize server weights)
3:   for round  $i = 1 \dots T$  do
4:      $S_m \leftarrow$  (set of  $m$  clients data, each starting at day  $i$  and ending at  $i + 6$ )
5:     for client  $c \in S_m$  in parallel do
6:        $w_c^i \leftarrow$  CLIENTUPDATE( $w_g^{i-1}, c$ )
7:     end for
8:      $w_g^i \leftarrow \frac{1}{m} \sum_{c=1}^m w_c^i$ 
9:   end for
10: end procedure
11:
12: function CLIENTUPDATE( $w, c$ )
13:   for local step  $k = 0 \dots K$  do
14:      $w \leftarrow w - \eta \nabla f(w, z)$  for  $z \sim c$ 
15:   end for
16:   return  $w$  to server
17: end function

```

---

No processo de treinamento, o cliente deve receber o modelo global do servidor em cada rodada para realizar o processo de treinamento local (Linha 12, Algoritmo 1). De posse desse modelo, cada cliente executa  $K$  iterações locais para treinar o seu modelo local (Linhas 13-15, Algoritmo 1). Em cada iteração, o cliente, com seus dados locais, otimiza o modelo por meio do método do gradiente descendente (Linha 14, Algoritmo 1). Por fim, o modelo local resultante desse treinamento é retornado ao servidor para ser agregado com os resultados dos outros clientes (Linha 16, Algoritmo 1).

Finalmente, na agregação dos resultados locais, foi utilizada a técnica de *Federated Averaging* (FedAvg) [49] (Linha 8, Algoritmo 1). No FedAvg, os resultados que os clientes retornam ao servidor central são os pesos do modelo local, ao invés dos gradientes, e o servidor faz uma média dos pesos durante a etapa de agregação. Essa técnica se baseia no fato de que se todos os clientes possuem a mesma inicialização, fazer uma média dos gradientes é equivalente a fazer a média dos pesos.

### 4.3 Base de Dados

Nesta seção é apresentada a base de dados utilizada para validar o RADAR. Para isto, foi construído um conjunto de dados de tráfego de veículos capturados por sensores eletrônicos.

A Figura 4.4 apresenta os sensores localizados em dois cruzamentos do Distrito Federal - DF. Os dados foram capturados na avenida Hélio Prates, que cruza as regiões de Taguatinga e Ceilândia, por meio do sistema do Departamento de Trânsito do DF (Detran-DF).

A Tabela 4.1 apresenta uma amostra dos dados coletados pelos sensores. Cada linha do conjunto de dados contém informações sobre o fluxo de veículos na via monitorada, sendo elas: (i) a data e hora da observação, (ii) a velocidade do veículo naquele momento, (iii) a velocidade máxima permitida na via, (iv) em qual das faixas da via ele se encontrava e (v) o tamanho do veículo.

Tabela 4.1: Amostra de dados coletados pelos sistemas do Detran-DF.

<b>Id. do Sensor</b>	<b>Data</b>	<b>Hora</b>	<b>Faixa</b>	<b>Vel.</b>	<b>Vel. Máxima</b>	<b>Tamanho</b>
RSI128	2016/05/01	00:00:09	1	26.0	60.0	0.0
RSI131	2016/05/01	00:00:09	2	20.0	60.0	1.1
RSI132	2016/05/01	00:00:09	1	45.0	60.0	0.0
RSI131	2016/05/01	00:00:10	1	40.0	60.0	0.5
RSI129	2016/05/01	00:00:12	1	35.0	60.0	0.0

Para esta pesquisa, foram utilizados dados coletados por oito sensores entre maio de 2016 até julho do mesmo ano. Cada sensor possui um identificador, *i.e.*, um valor prefixado por *RSI* mais um número de identificação, que permite o diferenciar dos demais. A Figura 4.4 apresenta a localização dos sensores utilizados neste trabalho, identificados pelos valores *RSI017*, *RSI018*, *RSI032*, *RSI033*, *RSI128*, *RSI129*, *RSI131* e *RSI132*. Esses oito sensores modelam o cenário descrito na Seção 4.1, ou seja, cada um desses sensores capta o tráfego de um sentido de uma das vias dos cruzamentos.

A Tabela 4.2 apresenta e descreve os atributos contidos no conjunto de dados utilizado neste trabalho. Podemos notar que a velocidade máxima da via é a mesma em todos os casos, sempre igual a 60 km/h. Além disso, temos valores de difícil interpretação no atributo de tamanho dos veículos. Isso é agravado pelo fato do conjunto possuir veículos de tamanho igual a 0, o que não faz sentido fisicamente. Assim, temos um indicativo de que existem falhas na leitura de tamanho dos veículos.

Salienta-se também que a distribuição dos registros na base de dados entre os sensores não é uniforme. Como apresentado na Tabela 4.3, observa-se que há uma diferença significativa no número de veículos observados entre os quatro sensores com menos registros (*RSI128*, *RSI129*, *RSI132* e *RSI131*) e os quatro sensores com mais registros (*RSI017*, *RSI018*, *RSI032*, *RSI033*). Essa diferença na distribuição de registros entre os sensores aponta para uma característica das vias nas quais esses dados estão sendo coletados. Em outras palavras, o fluxo de tráfego nas vias dos sensores *RSI017*, *RSI018*, *RSI032* e



Figura 4.4: Localização dos sensores.

*RSI033* é maior do que o fluxo nas vias dos sensores *RSI128*, *RSI129*, *RSI131* e *RSI132*. Isso pode ter implicações nos resultados do RADAR, dado que o erro obtido na predição do fluxo será naturalmente maior em vias onde trafegam mais veículos.

### 4.3.1 Pré-processamento dos dados

Nesta subseção é apresentada como a extração de dados foi usada para validar o RADAR. Isso envolve realizar um pré-processamento da base de dados.

A Tabela 4.1 mostra que a base de dados utilizada inicialmente possui os registros dos veículos observados por cada sensor. Entretanto, para o RADAR, é necessário obter o fluxo de veículos em um dado intervalo de tempo. Para esta pesquisa, utilizou-se um intervalo de cinco minutos, conforme o sistema proposto é modelado.

Para tanto, inicialmente descartamos as informações presentes nos dados que não serão úteis para o modelo, tais como velocidade máxima da via, a faixa que o veículo se encontra e o tamanho do veículo. As informações de faixa e tamanho dos veículos são descartadas pois não são úteis quando agregadas por toda a via, conforme será feito no cálculo do fluxo de veículos. No caso do tamanho dos veículos, há ainda o agravante de que os sensores apresentam falhas nessa medição, com alguns registros apontando veículos com tamanho igual a zero, um valor impossível de ocorrer no mundo real. No caso, como o sistema trabalha com fluxo de tráfego, a informação de velocidade máxima da via também pode ser ignorada, visto que ela será constante ao longo do tempo e é a mesma para todas as vias.

Tabela 4.2: Descrição do conjunto de dados

<b>Formato</b>	CSV
<b>Quantidade de sensores</b>	8
<b>Quantidade de instâncias</b>	10.801.781
<b>Velocidade da via (km/h)</b>	Máxima: 60 km/h
<b>Velocidade do veículo (km/h)</b>	[1, 36.55, 112]
<b>Tamanho dos veículos</b>	[0, 0.10, 50]
<b>Faixa</b>	1, 2, ou 3
<b>Início da coleta</b>	2016/05/01 00:00:09
<b>Fim da coleta</b>	2016/07/31 23:59:49
<b>Tarefa</b>	Predição do fluxo de veículos
<b>Predição (Target)</b>	Fluxo de veículos
<b>Features</b>	Velocidade média, densidade, fluxo e tipo de dia
<b>Tipo de dia</b>	Final de semana e meio de semana

Tabela 4.3: Número de registros por sensor

<b>Identificador do Sensor</b>	<b>Número de registros</b>
<b>RSI033</b>	2.382.754
<b>RSI032</b>	2.117.820
<b>RSI018</b>	2.029.559
<b>RSI017</b>	1.686.900
<b>RSI131</b>	816.219
<b>RSI132</b>	652.998
<b>RSI129</b>	578.652
<b>RSI128</b>	536.879

Com essas informações descartadas, precisamos adaptar os dados captados pelos sensores para o RADAR. Para isso, primeiramente podemos agrupar as colunas de data e hora em uma única coluna contendo o momento no tempo em que o registro foi feito, facilitando etapas futuras de pré-processamento que utilizem essa informação. Isso será importante no cálculo do fluxo de veículos, dado que o fluxo será agregado em um intervalo de tempo de reamostragem. Além disso, de modo a facilitar ainda mais o pré-processamento, é possível indexar os registros por essa nova coluna de tempo, garantindo a ordenação temporal dos mesmos. Em seguida, podemos utilizar os identificadores presentes nos registros para separar os dados de modo a se obter oito diferentes subconjuntos, cada um contendo os dados de um sensor específico. Desse modo, cada subconjunto poderá representar um cliente na rede federada do RADAR.

Realizada a separação dos sensores, temos que reamostrar os dados para o fluxo de tráfego no intervalo de tempo definido. Assim, as informações de quantidade de veículos e

velocidade média dos veículos são agregadas no intervalo de reamostragem. Além dessas duas informações, é calculada também a densidade de tráfego por meio da divisão do fluxo de tráfego pela velocidade média.

A Figura 4.5 apresenta o resultado da reamostragem de fluxo de tráfego para o sensor *RSI132*. Como é possível observar, cada dia passa a ser representado por entradas contendo a quantidade de veículos que passaram pelo sensor no intervalo de tempo definido para o cálculo do fluxo de tráfego.

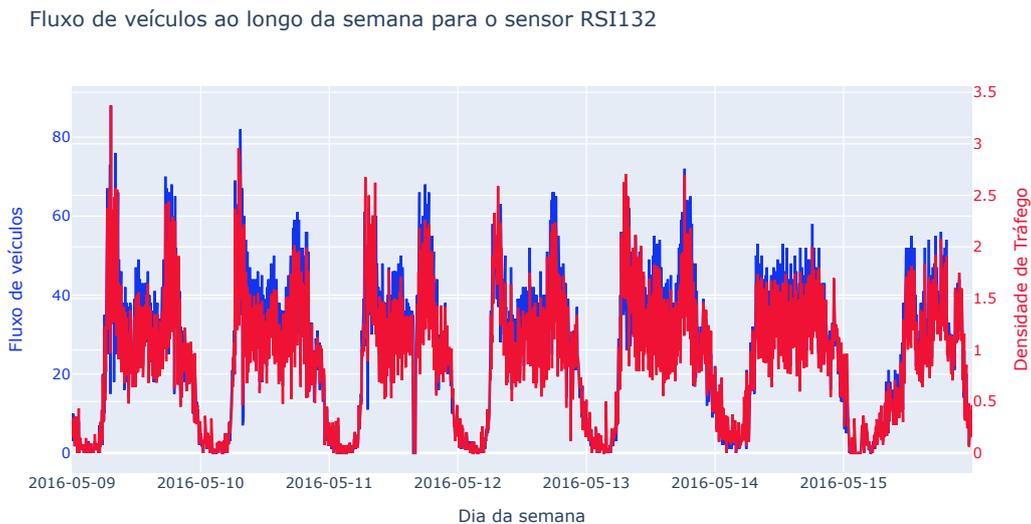


Figura 4.5: Fluxo de veículos ao longo da semana para o sensor *RSI132*.

Contudo, observando os índices temporais de cada conjunto, é possível notar que os registros não partem da mesma origem, dado que o início das observações de alguns sensores começam horas ou até dias após o início das observações de outros sensores. Conforme a modelagem do RADAR, é preciso garantir que os dados dos diferentes conjuntos representem o mesmo período de tempo, de modo que os modelos sejam treinados e as previsões sejam feitas em um referencial temporal comum para todos os clientes. Portanto, os dados dos dias anteriores a 3 de maio de 2016, quando existirem em um conjunto, são descartados, de modo que todos os conjuntos iniciem na mesma hora e no mesmo dia.

Finalmente, um outro fator a se considerar no RADAR é a diferença no fluxo de veículos ao longo da semana. Para tratar esse ponto, são criadas colunas categóricas contendo a informação do tipo de dia em que aquelas observações foram feitas. Para realizar essa categorização, o tipo de dia foi agrupado em: (i) final de semana - sábado e domingo; e (ii) meio de semana, de segunda até sexta. Assim, são introduzidas duas colunas indicando se aquele dia é um dia de final de semana ou um dia de meio de semana.

Essas colunas possuem valor igual a 1 caso seja um dia referente àquele tipo ou valor igual a 0 caso contrário.

## 4.4 Considerações Finais

Neste capítulo, o RADAR, um modelo de *Federated Learning* para predição de fluxo de veículos em vias urbanas, foi exposto e suas características foram detalhadas. Desse modo, um cenário consistindo de oito sensores distribuídos por dois cruzamentos foi elaborado. Nele, cada sensor participa de uma rede de Aprendizado Federado, treinando uma rede neural *LSTM* com os seus dados locais e enviando os resultados desse treinamento para um servidor central, responsável por agregar esses resultados em um modelo global.

Para realizar essa tarefa, uma base de dados composta por registros de sensores eletrônicos no Distrito Federal foi adaptada e pré-processada de modo a se encaixar no cenário proposto. Assim, cada sensor nessa base passa a representar um cliente na rede de Aprendizado Federado, sendo encarregado de treinar seu modelo local com os seus próprios dados e enviar os pesos resultantes do treinamento para o servidor central, que os agrega utilizando a técnica de *Federated Averaging*.

Com o modelo proposto tendo sido apresentado, é preciso ainda examinar o seu desempenho através de experimentos. No capítulo seguinte, serão explicitadas as configurações dos experimentos realizados e as métricas de avaliação utilizadas, com os resultados provenientes da experimentação sendo evidenciados e comparados.

# Capítulo 5

## Resultados Experimentais

Este capítulo apresenta a metodologia dos experimentos e os resultados obtidos na validação do RADAR. Para isso, o RADAR foi validado considerando duas etapas:

1. Cenário de variação do número de participantes na rede, com o objetivo de avaliar os parâmetros do RADAR e escolher a configuração ideal da rede, determinando a capacidade do RADAR em estimar o fluxo de tráfego.
2. Cenário de variação do tipo de dia da semana, *i.e.* final de semana ou meio de semana, com o objetivo de avaliar o RADAR de acordo com a densidade de veículos na via.

Em tais avaliações, o RADAR foi avaliado comparando-o com um modelo tradicional centralizado, sem o uso de *Federated Learning*. O modelo centralizado se baseia na predição de fluxo para um sensor particular, através de uma rede LSTM tradicional, que podemos usar como base de comparação para o desempenho do modelo federado utilizado no RADAR. Desse modo, neste capítulo primeiramente será descrita a configuração dos experimentos usados para avaliar o RADAR. Em seguida, serão apresentados os resultados dos experimentos de variação do número de participantes e de variação do tipo de dia da semana, respectivamente.

### 5.1 Configuração dos Experimentos

Nesta seção será descrita a configuração dos experimentos e a metodologia usada na avaliação do RADAR. Para realizar o processo de avaliação, em um primeiro momento foi realizado o treinamento da rede de FL do RADAR. Para isso, a cada rodada federada de um experimento, são coletadas métricas que permitem analisar a performance do RADAR. Como o objetivo do RADAR é prever o fluxo de tráfego, as métricas abaixo foram

escolhidas. Nas fórmulas abaixo,  $n$  é o número de observações consideradas,  $y_i$  é o valor previsto para a  $i$ -ésima observação e  $x_i$  é o valor real da  $i$ -ésima observação.

- *Mean Absolute Error* (MAE) =  $\frac{1}{n} \sum_{i=1}^n |y_i - x_i|$ , esta métrica tem o objetivo de quantificar o quão próxima a predição está do valor real.
- *Mean Squared Error* (MSE) =  $\frac{1}{n} \sum_{i=1}^n (y_i - x_i)^2$ , esta métrica tem o objetivo de avaliar o impacto de predições que estejam muito distantes do valor real.
- *Root Mean Squared Error* (RMSE) =  $\frac{1}{n} \sum_{i=1}^n \sqrt{(y_i - x_i)^2}$ , esta métrica tem o objetivo de melhorar a interpretabilidade do MSE, uma vez que seus resultados são apresentados na mesma unidade dos valores observados.

Com as métricas definidas, podemos descrever algumas configurações utilizadas nos experimentos. Primeiramente é preciso definir uma função de custo para os experimentos do RADAR. Dentre as métricas consideradas, o MSE foi escolhido por penalizar grandes erros com o seu fator quadrático. Com isso, seu uso busca reduzir o impacto de valores aberrantes na predição. Além da escolha da função de custo, há ainda o fato de que são realizadas diferentes rodadas de treinamento em um experimento, e cada rodada de treinamento terminará com resultados diferentes para as métricas coletadas. Contudo, nas comparações entre experimentos é preciso eleger um resultado representativo para cada experimento. Nesse caso, são utilizados os resultados da rodada em que foi encontrado o menor valor para a função de custo, *i.e.*, a rodada que minimizou a métrica MSE. Finalmente, é considerado também o efeito estocástico presente nos experimentos. Para tanto, o processo de experimentação foi repetido 33 vezes em cada cenário. Desta forma, é possível observar a influência de efeitos aleatórios inerentes a rede RADAR, bem como sua robustez a esses fatores.

A Figura 5.1 apresenta a técnica de *Walk-Forward Validation* [50] utilizada para gerar o modelo do RADAR. Temos distribuídos verticalmente as diferentes rodadas de treinamento, enquanto no eixo horizontal há a distribuição da série temporal ao longo do tempo. Desse modo, em cada rodada, os clientes da rede treinam seus modelos locais utilizando uma janela temporal contendo dados de treinamento, indicados na figura com um verde mais claro. Ao final da rodada, o modelo resultante é validado com dados de teste nos quais ele ainda não foi treinado, indicados na figura com um verde mais escuro. Com isso, a janela temporal da próxima rodada de treinamento é avançada, evitando sobre-ajustes para um período específico. Assim, o uso dessa técnica permite aproveitar o fato de se estar tratando uma série temporal para fazer uma validação cruzada dos dados.

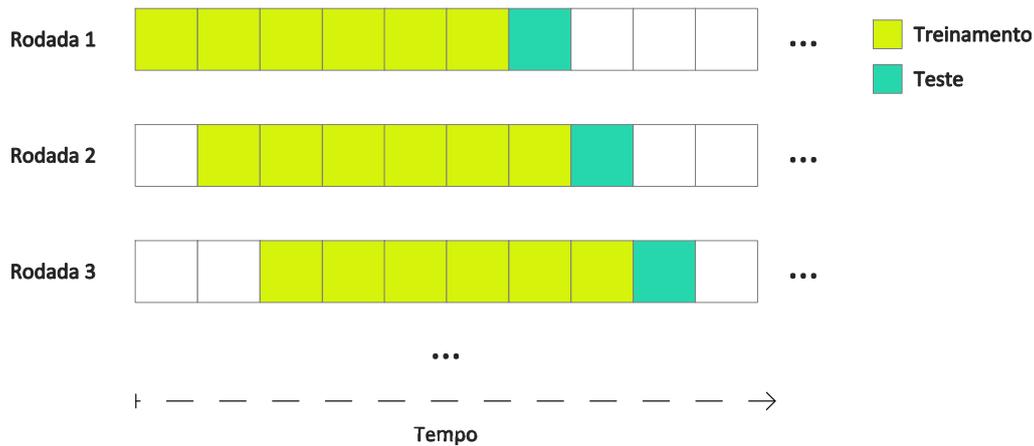


Figura 5.1: Utilização de *Walk-Forward Validation* nos experimentos do RADAR

### 5.1.1 Definição dos Cenários

Os experimentos realizados na avaliação do RADAR se dividiram em duas etapas: (i) cenários envolvendo a variação do número de participantes, com objetivo de encontrar a melhor configuração para a rede; e (ii) cenários envolvendo a variação do tipo de dia da semana, com o objetivo de avaliar o RADAR em diferentes condições quanto a densidade de veículos na via.

A etapa de variação do número de participantes teve como objetivo encontrar a melhor configuração para a rede. Desse modo, os experimentos contemplaram 7 configurações, cada uma envolvendo, respectivamente, a participação de 2, 3, 4, 5, 6, 7 e 8 clientes. A escolha dos clientes participantes em cada experimento se deu de forma aleatória, *i.e.*, se  $n$  é o número de clientes definido para aquela configuração, em cada experimento escolheu-se  $n$  clientes aleatoriamente dentre os 8 disponíveis. Como são feitos 33 experimentos para cada configuração, os participantes dos experimentos dentro de uma configuração podem variar devido essa escolha aleatória. Observando os resultados da primeira etapa, escolheu-se o melhor número de sensores para a realização da segunda etapa de experimentos.

A etapa de variação do tipo de dia da semana, por sua vez, teve como objetivo avaliar a performance do RADAR em diferentes contextos temporais, tendo em vista a maneira com que esse fator afeta a densidade da via. Para isso, foram realizados experimentos considerando duas diferentes configurações com relação ao tipo do dia em que a medição de fluxo foi feita: (i) dias de Final de Semana, nos quais a densidade da via é baixa, ou (ii) dias de Meio de Semana, nos quais a densidade da via é alta. Portanto, em cada configuração, o treinamento e a avaliação do modelo RADAR foi realizado apenas com dias referentes ao tipo de dia daquela configuração. Nesses experimentos, foram utilizados os resultados da etapa anterior para se definir a quantidade de clientes ideal na rede.

Além disso, em ambas etapas de experimentos, foi realizada uma comparação com um modelo de aprendizado centralizado. Para essas comparações, dois diferentes sensores, *RSI032* e *RSI132*, foram escolhidos. A escolha desses dois sensores ocorreu pelo fato de estarem localizados em vias com diferentes níveis de fluxo de veículos. Mais especificamente, conforme a Tabela 4.3, o sensor *RSI032* possui um fluxo de tráfego muito superior ao sensor *RSI132*. Assim, o sensor *RSI032* foi escolhido para representar os sensores de vias que possuem um alto fluxo de veículos, enquanto o *RSI132* foi escolhido para representar os sensores de vias que possuem baixo fluxo de veículos.

### 5.1.2 Escolha de parâmetros e hiper-parâmetros

Antes de implementar o processo de aprendizagem foi realizada uma hiper-parametrização do FL no RADAR. Por se tratar de um modelo de FL, além dos hiper-parâmetros comuns a uma LSTM, no RADAR temos ainda parâmetros específicos do processo federado.

A otimização desses fatores é um passo importante para garantir que o modelo tenha um bom desempenho, porém apresenta alguns desafios. Devido a sua característica descentralizada e por possuir múltiplos clientes, o processo de hiper-parametrização de uma rede FL naturalmente possui um custo computacional elevado. Isso significa que técnicas tradicionais que visam explorar uma grande gama de hiper-parâmetros podem se tornar inviáveis. Esse fato se torna ainda mais importante no contexto do RADAR, tendo em vista os escassos recursos computacionais disponíveis. Portanto, a redução do custo computacional, tanto na hiper-parametrização quanto nos experimentos, também é um fator a ser considerado. Assim, a escolha de hiper-parâmetros busca a otimização da performance do modelo, porém, no contexto do RADAR, alguns parâmetros deverão ser escolhidos levando em conta a redução do custo computacional.

Na Tabela 5.1 temos os valores explorados durante o processo de hiper-parametrização do RADAR. Para realizar a hiper-parametrização do modelo foi realizado um *Grid Search* [51] envolvendo cinco parâmetros: (i) *Units*, (ii) *Activation*, (iii) *Dropout*, (iv) *Recurrent Dropout* e (v) *Learning Rate*. Durante o processo de *Grid Search*, todas as combinações entre os valores considerados para os parâmetros escolhidos são testadas. A hiper-parametrização utiliza todos os 8 clientes e, de modo a reduzir o custo computacional, é realizada apenas com os 20 primeiros dias. Com isso, o melhor resultado é aquele que minimiza a função de custo definida para a LSTM, sendo nesse caso o MSE. O melhor valor para cada hiper-parâmetro é destacado em negrito. Além disso, o *Min. Loss* encontrado para os melhores hiper-parâmetros também é apresentado na tabela.

Outro ponto importante a se considerar é o número de rodadas federadas realizadas. De modo geral, realizar mais rodadas implica em melhores resultados pois o modelo converge iterativamente a cada rodada. No processo de hiper-parametrização, para cada

Tabela 5.1: Conjunto de parâmetros do processo de hiper-parametrização e Min. Loss encontrado

Parâmetro	Valores	Min. Loss
<i>Units</i>	[8, 16, 32, <b>64</b> ]	
<i>Activation</i>	[ <b>tanh</b> , relu]	
<i>Dropout</i>	[ <b>0.0</b> , 0.1, 0.2]	<b>265.12</b>
<i>Rec. Dropout</i>	[0.0, 0.1, <b>0.2</b> ]	
<i>Learning Rate</i>	[0.01, <b>0.001</b> ]	

combinação testada, são realizadas 10 rodadas federadas de treinamento. O número de rodadas foi escolhido de modo a possibilitar a exploração de uma quantidade razoável de parâmetros, mantendo um número de rodadas suficiente para avaliar a rede. Durante os experimentos são utilizadas 20 rodadas federadas de treinamento, ao invés das 10 rodadas realizadas na hiper-parametrização. Uma exceção é feita no cenário de Final de Semana, no qual não há dados suficientes para realizar 20 rodadas com Walk-Forward Validation. Assim, nesse caso são realizadas 19 rodadas federadas de treinamento.

## 5.2 Avaliação da capacidade do RADAR em estimar o fluxo de tráfego

Nesta seção serão apresentados os resultados dos experimentos envolvendo a variação do número de participantes na rede federada. O objetivo dessa etapa da avaliação é encontrar a melhor configuração para a rede do RADAR e aferir o número ótimo de clientes participantes para a próxima etapa.

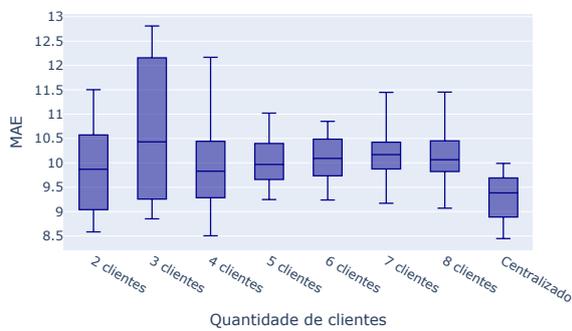
A Figura 5.2 apresenta a variação de clientes no RADAR em função das métricas MAE e RMSE para uma via com baixa e alta densidade. Além disso, os resultados do RADAR são comparados com os resultados do modelo centralizado. O objetivo é avaliar a eficiência das predições do RADAR, *i.e.* o quão próxima a predição está dos valores reais. Além disso, com o RMSE é possível observar o impacto de predições que estejam muito distantes dos valores esperados. É possível notar nos gráficos que, de modo geral, há uma tendência de diminuição na dispersão dos resultados com o aumento do número de clientes. Nas Figuras 5.2b e 5.2d, respectivamente, se observa uma redução de 82.45% e 74.44% na dispersão quando o caso de 2 clientes é comparado com o de 8 clientes. Esse efeito ocorre pois, com poucos clientes, é possível ter combinações apenas entre clientes que tenham baixo ou alto fluxo. Ao aumentar o número de clientes, reduz-se a probabilidade de um experimento conter esses tipos de arranjos, diminuindo assim a dispersão no gráfico como apresentado nas figuras Figuras 5.2b e 5.2d. Já nas Figuras 5.2a e 5.2c há um aumento na

dispersão quando o caso de 2 clientes é comparado com o de 3 clientes. Esse efeito pode ser atribuído a uma combinação de 3 clientes que gera um resultado muito discrepante dos demais. Ainda assim, novamente nas Figuras 5.2a e 5.2c a tendência de diminuição volta a ser observada com uma redução na dispersão de 78.28% e 62.18%, respectivamente, quando o caso de 3 clientes é comparado com o de 8 clientes. Portanto, a Figura 5.2 indica que os resultados são sensíveis com relação aos clientes que foram utilizados em cada experimento. Em experimentos com poucos clientes, essa sensibilidade se torna mais acentuada, enquanto em experimentos com muitos clientes ela é atenuada.

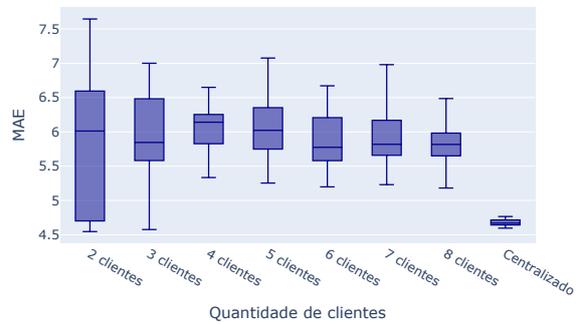
Ao considerar uma via com alta densidade, Figuras 5.2a e 5.2c, observa-se, com relação às medianas, que o modelo centralizado obteve resultados levemente melhores do que o RADAR. No melhor caso, 4 clientes, o modelo centralizado apresentou uma redução de 4.53% e 9.75% para o MAE e RMSE, respectivamente, quando comparado com o RADAR. De fato, isso está dentro do que se espera ao realizar uma comparação de um modelo FL com um modelo centralizado. O modelo centralizado utiliza dados de apenas um sensor, sendo com isso capaz de otimizar a predição especificamente para esse sensor. Por outro lado, um modelo federado utiliza dados de múltiplos sensores e precisa produzir um modelo que se adapte bem a todos os clientes da rede. Esse fato faz com que o RADAR não seja tão agressivo quanto o modelo centralizado em otimizar o modelo especificamente para uma das vias. Nesse ponto, modelos federados apresentam um *trade-off* com relação a modelos centralizados, no sentido que modelos federados possuem vantagens importantes, como no tempo de tomada de decisão, justamente por serem descentralizados. Ainda assim, os resultados do RADAR foram bons o bastante para serem competitivos, uma vez que o modelo centralizado obteve uma redução de apenas 4.53% para o MAE e 9.75% para o RMSE.

Ao observar a via com baixa densidade, Figuras 5.2b e 5.2d, nota-se que os resultados do modelo centralizado foram novamente superiores aos do RADAR. Assim como na via de alta densidade, uma vantagem do modelo centralizado é, de fato, algo esperado. Porém, no caso de baixa densidade, essa diferença se deu de maneira mais acentuada do que anteriormente, com o modelo centralizado apresentando uma redução de 19.14% e 12.77% nos resultados de MAE e RMSE, respectivamente, se comparados com o melhor caso do RADAR, utilizando 6 clientes. Isso pode indicar que a menor quantidade de observações disponíveis resultou em uma maior dificuldade do RADAR em adaptar um modelo preditivo para a via. Um outro ponto a se notar é que, ao contrário do caso de alto fluxo de veículos, a vantagem do modelo centralizado no MAE foi maior do que a no RMSE. Essa diferença pode ser atribuída ao fato do modelo centralizado ser mais agressivo em otimizar sua predição, se tornando, dessa forma, também mais suscetível a produzir erros maiores caso as características do fluxo da via, por qualquer motivo, se

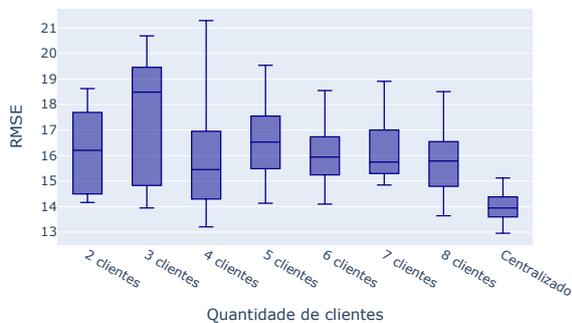
alterem para um padrão não visto anteriormente. Já no RADAR o modelo é treinado para um conjunto de clientes, portanto caso exista uma alteração anômala no fluxo de uma via é possível que algum outro cliente na rede já tenha visto algo parecido, como pode ser observado pela baixa amplitude interquartil na figura Figura 5.2. Além disso, é possível observar também que os resultados encontrados nas Figuras 5.2b e 5.2d foram consideravelmente menores do que os seus pares nas Figuras 5.2a e 5.2c. Essa diferença se dá pelo fato do fluxo de veículos em um sensor ser maior do que no outro, como pode ser observado na Tabela 4.3. Se tratando de um sensor com um menor número de veículos, é natural que o erro encontrado também seja menor relativo ao sensor com mais veículos.



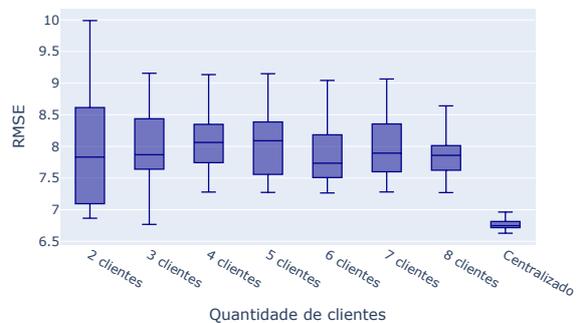
(a) MAE, alto fluxo de veículos



(b) MAE, baixo fluxo de veículos



(c) RMSE, alto fluxo de veículos



(d) RMSE, baixo fluxo de veículos

Figura 5.2: Variação dos melhores resultados obtidos em relação a quantidade de clientes utilizados

Observando esses resultados, a escolha da melhor configuração para a rede do RADAR foi realizada com base nas medianas dos valores de MAE e RMSE. Os melhores resultados encontrados foram obtidos utilizando 4 e 6 clientes para as vias de alto e baixo fluxo, respectivamente, e portanto essas foram as duas configurações utilizadas na próxima etapa. Cabe ressaltar que, de modo geral, os resultados obtidos nas outras configurações ficaram relativamente próximos dos melhores resultados. A única exceção observada é encontrada

no pior resultado de RMSE para a via de alto fluxo, com 3 clientes, que sofre de uma combinação de clientes gerando resultados muito discrepantes, ocasionando em uma diferença entre a pior e a melhor configuração de 17.87%. Por outro lado, a diferença entre a pior e a melhor configuração foi de apenas 5.94%, 6.14% e 4.48% para o MAE na via de alto fluxo e para o MAE e o RMSE na via de baixo fluxo, respectivamente. Portanto, dada a proximidade dos resultados, é possível concluir que o RADAR conseguiu adaptar a rede relativamente bem para a grande maioria das configurações, com a exceção do caso de 3 clientes para a via de alto fluxo.

A Figura 5.3 apresenta o Loss médio obtido nos experimentos ao longo das rodadas de aprendizado federado. O objetivo é avaliar a velocidade de convergência do modelo do RADAR. Como função de custo para o RADAR foi utilizada a métrica MSE. Observou-se que ao longo das rodadas o Loss foi minimizado até atingir uma convergência, independente da quantidade de clientes utilizados. No gráfico, podemos ver um grande decaimento nas primeiras 6 rodadas, o que é de se esperar já que a rede inicialmente irá partir de um ponto pouco otimizado. Em seguida há uma consolidação entre as faixas de 200 a 600, com um declínio mais leve no final do treinamento. Em modelos FL, geralmente é necessário um grande número de rodadas de treinamento para se obter resultados satisfatórios. Ainda que seja possível melhorar o resultado com mais rodadas, considerando o esperado de redes federadas, o modelo RADAR convergiu de maneira relativamente rápida. Ainda, levando em conta os resultados de MAE e RMSE vistos anteriormente, podemos dizer que a rede convergiu para resultados suficientemente bons. Ao todo foram realizadas 20 rodadas de treinamento, um valor que pode ser considerado baixo no contexto de FL. Essa rápida convergência se apresenta como um fator positivo encontrado nos experimentos. Tendo em conta que o RADAR considera sistemas com poucos recursos computacionais, realizar muitas rodadas de treinamento poderia tornar o RADAR inviável em sua proposta. Assim, podemos dizer que o modelo cumpre um dos seus objetivos ao demonstrar ser capaz de convergir rapidamente.

A Figura 5.4 apresenta os valores observados do fluxo de tráfego e a predição realizada pela rede RADAR durante um período de 7 dias. O objetivo é verificar se a predição de fato acompanha a forma da curva de fluxo observada. As Figuras 5.4a e 5.4b apresentam os valores observados e a predição de fluxo de tráfego para o caso de alto e baixo fluxo de veículos, respectivamente. No gráfico, podemos observar que durante o período considerado a predição acompanhou satisfatoriamente os valores reais observados. Como os valores de MAE e RMSE encontrados foram relativamente baixos e a predição seguiu a forma do fluxo real, podemos dizer que o RADAR fez uma previsão eficiente do fluxo de tráfego. Além disso, nota-se na Figura 5.4a que de modo geral os valores da predição não alcançam os patamares máximos e mínimos dos valores observados. Ao invés disso,

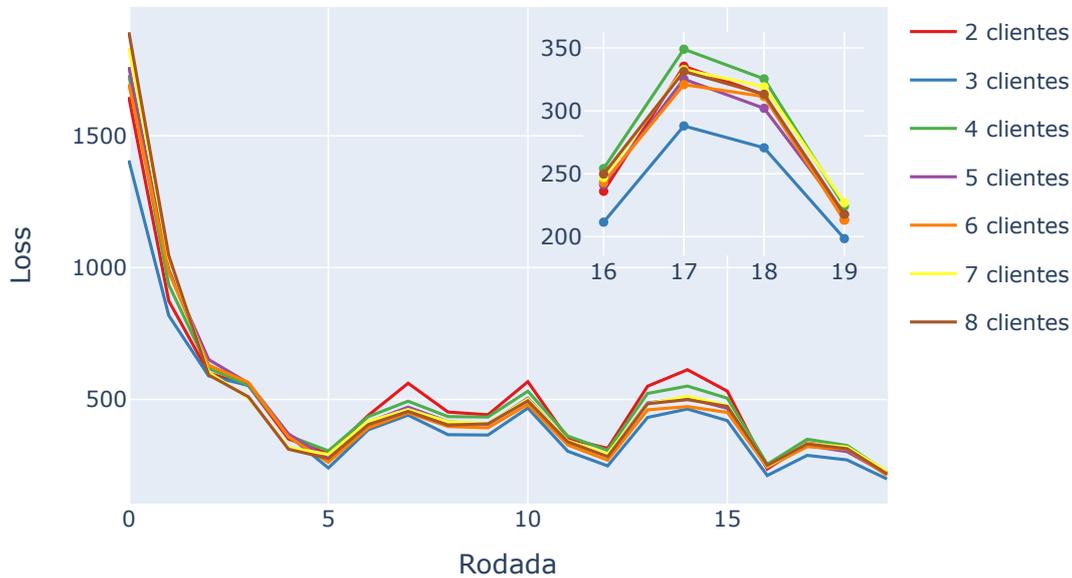
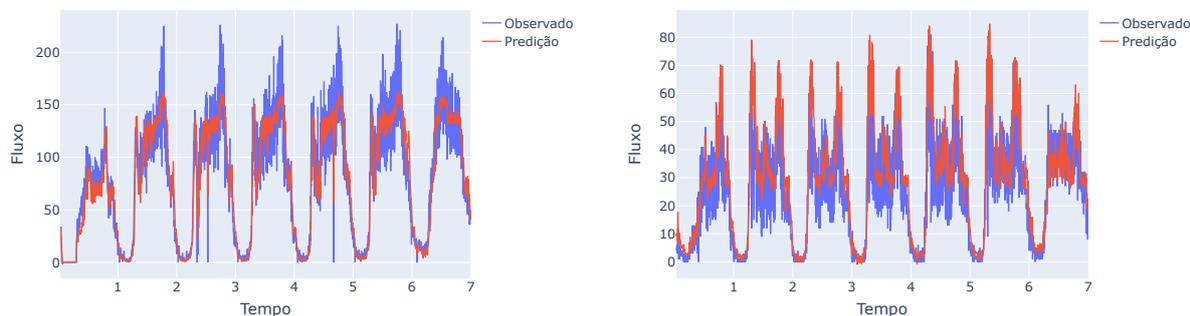


Figura 5.3: Loss médio ao longo das rodadas de treinamento

a predição parece se manter entre esses patamares, sem subir acima dos máximos nem descer abaixo dos mínimos. Isso poderia significar uma dificuldade do RADAR em lidar com valores de fluxo fora do comum. Ainda assim, em casos como o do primeiro dia nos gráficos das Figuras 5.4a e 5.4b, podemos ver que o modelo foi capaz de se sair bem em um caso no qual o fluxo de tráfego seguia uma forma pouco usual. Em outros casos, é possível ver a correta predição de picos ao fim do dia, ainda que não atinjam os picos observados. Um último ponto interessante a se comentar é a já citada diferença nos níveis de fluxo nos sensores *RSI032* e *RSI132*. Na Figura 5.4a, o *RSI032* possui picos bem elevados nos valores observados, chegando a ultrapassar o nível dos 200 veículos em 6 dos 7 dias. Por outro lado, na Figura 5.4b, os valores de pico do *RSI132* ficam sempre abaixo do nível de 90 veículos, ultrapassando o nível de 80 veículos em apenas três ocasiões. Desse modo, os picos do *RSI032* são bem maiores do que os picos do *RSI132*, o que remete à diferença nos níveis de fluxo que podem ser observados na Tabela 4.3. Ainda assim, as Figuras 5.4a e 5.4b mostram que a predição do modelo federado do RADAR se adaptou a esses diferentes níveis.



(a) Alto fluxo de veículos

(b) Baixo fluxo de veículos

Figura 5.4: Fluxo observado vs. predição em um dos experimentos

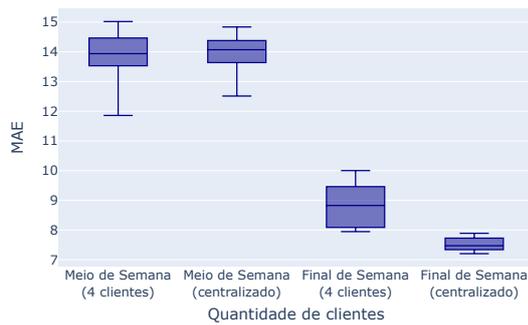
### 5.3 Análise do desempenho do RADAR em diferentes cenários de densidade

Nesta seção serão apresentados os resultados para a etapa de variação do tipo de dia da semana. O objetivo dessa etapa é avaliar o RADAR de acordo com a densidade de veículos na via. Nos experimentos o RADAR foi configurado com 4 clientes para o caso da via de alto fluxo e 6 clientes para o caso da via de baixo fluxo, conforme definido na seção anterior.

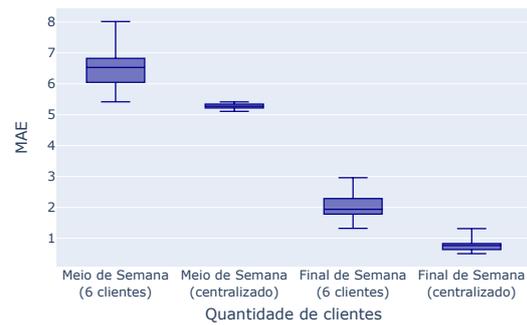
A Figura 5.5 apresenta a variação do tipo de dia da semana em função das métricas MAE e RMSE para uma via com baixa e alta densidade, comparando-o com o modelo centralizado. O objetivo é avaliar o desempenho do RADAR em diferentes contextos de densidade de fluxo de tráfego. Podemos observar que os valores encontrados nos resultados do cenário de Meio de Semana são mais altos do que os encontrados nos resultados do cenário de Final de Semana. Isso pode ser explicado devido a diferença no nível de fluxo entre esses tipos de dia. De modo geral, dias de Final de Semana possuem fluxo menor do que dias de Meio de Semana. Portanto, é natural que o erro medido pelo MAE e pelo RMSE seja menor em dias de Final de Semana. Além disso, um ponto recorrente encontrado nos resultados obtidos é a boa performance do cenário de Final de Semana com o modelo centralizado. Um dos fatores que pode ter levado o RADAR a apresentar resultados inferiores no cenário de Final de Semana em comparação ao cenário de Meio de Semana é a quantidade de dados disponíveis para treinar o modelo. Ao se usar apenas os finais de semana, joga-se fora cerca de 71% dos dados disponíveis.

As Figuras 5.5a e 5.5c apresentam os resultados de MAE e RMSE, respectivamente, para a via com alta densidade de veículos. É possível notar que o caso de Meio de Semana possui resultados mais competitivos entre o RADAR e o modelo centralizado, com

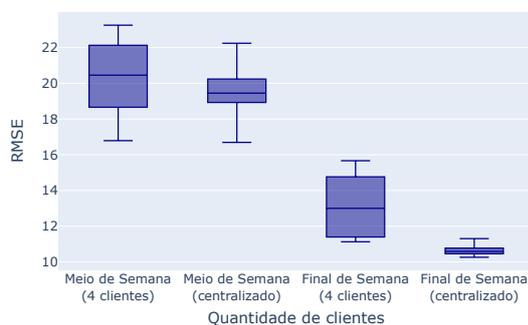
o RADAR chegando inclusive a apresentar uma leve vantagem de 0.94% no MAE. Nos outros casos, o modelo centralizado obteve um melhor resultado se comparado ao RADAR, observando 4.91%, 15.33% e 18.56% de redução para o RMSE de Meio de Semana, MAE de Final de Semana e RMSE de Final de Semana, respectivamente. Já as Figuras 5.5b e 5.5d apresentam os resultados para a via com baixa densidade de veículos. Se observa no cenário de Meio de Semana que o modelo centralizado obteve uma redução de 19.21% e 21.73% para o MAE e o RMSE, respectivamente, quando comparado aos resultados do RADAR. Já para o caso de Final de Semana, a redução observada salta para 60.46% no MAE e 48.75% no RMSE. Cabe ressaltar que, nesse caso, um dos fatores que explica a alta porcentagem de redução é o fato do erro já ser um valor relativamente baixo. Com valores menores, é natural observar uma variação percentual maior. Além disso, aqui o fator da menor quantidade de dados disponíveis para treinar o modelo é acentuado pelo fato de se tratar de uma via com baixa densidade de veículos, que naturalmente possui menos observações.



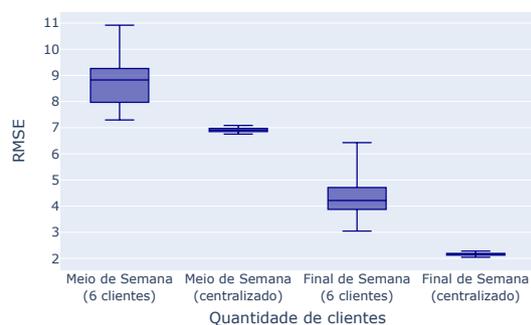
(a) MAE, alto fluxo de veículos



(b) MAE, baixo fluxo de veículos



(c) RMSE, alto fluxo de veículos

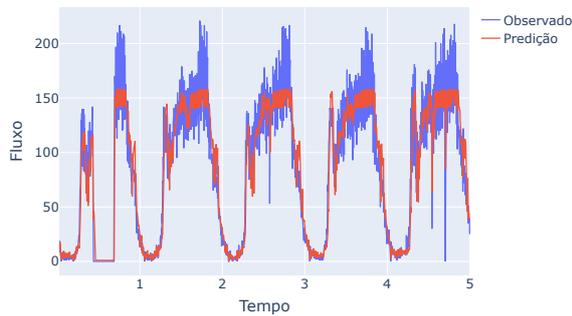


(d) RMSE, baixo fluxo de veículos

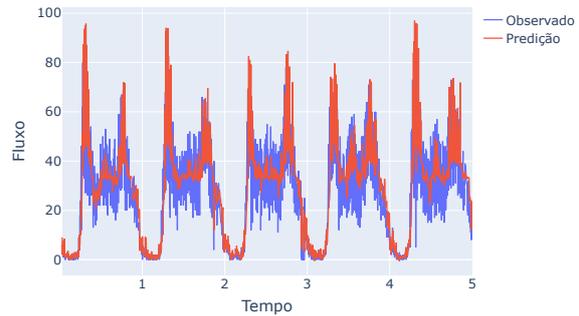
Figura 5.5: Variação dos melhores resultados obtidos em relação a quantidade de clientes utilizados

A Figura 5.6 apresenta os valores observados do fluxo de tráfego e a previsão realizada

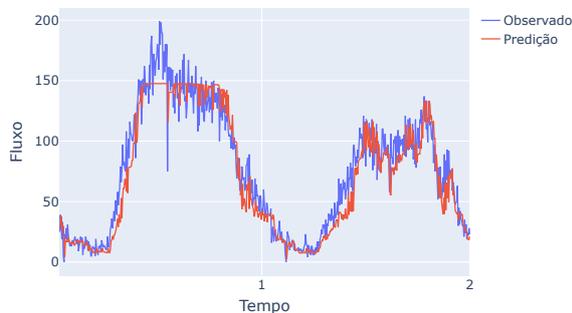
pelo RADAR nos cenários de Meio de Semana e Final de Semana. O objetivo é observar o comportamento do fluxo de veículos ao longo do tempo e verificar se a predição de fato acompanha os valores observados nos diferentes cenários de densidade considerados. Para o cenário de Meio de Semana, Figuras 5.6a e 5.6b, é considerado um período de 5 dias, *i.e.* de segunda a sexta. Já para o cenário de Final de Semana, Figuras 5.6c e 5.6d, é considerado um período de 2 dias, *i.e.* um final de semana inteiro. Em todos os casos, é possível observar que a predição do fluxo ao longo dos dias parece acompanhar a forma da curva de fluxo observada no mesmo período. É possível notar também a já referida diferença de fluxo de veículos presente nas vias em que os sensores se encontram, com a via de alto fluxo apresentando picos acima do patamar de 200 veículos, enquanto a via de baixo fluxo não ultrapassa o patamar de 100 veículos em nenhum dia. Ainda, podemos comparar os níveis de fluxo nas Figuras 5.6a e 5.6b com os níveis de fluxo das Figuras 5.6c e 5.6d para ilustrar a diferença no fluxo de veículos entre cenários de Meio de Semana e Final de Semana. De modo geral, o cenário de Final de Semana apresenta picos de fluxo menores do que o cenário de Meio de Semana.



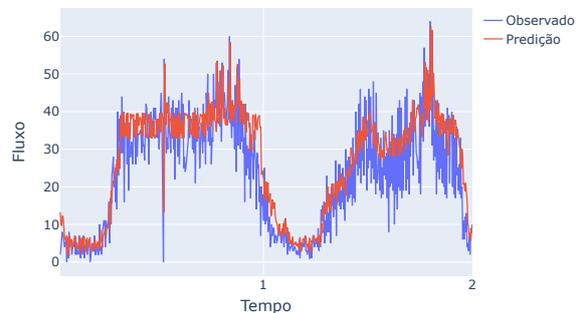
(a) Meio de Semana, via de alto fluxo de veículos



(b) Meio de Semana, via de baixo fluxo de veículos



(c) Final de Semana, via de alto fluxo de veículos



(d) Final de Semana, via de baixo fluxo de veículos

Figura 5.6: Variação dos melhores resultados obtidos em relação ao tipo de dia da semana

## 5.4 Considerações Finais

Na etapa envolvendo a variação do número de clientes na rede, foi observada uma redução da dispersão dos resultados com o aumento do número de participantes, devido a redução na variabilidade de fluxo entre diferentes clientes. Além disso, nos casos avaliados, os resultados obtidos mostraram que o RADAR consegue ser competitivo com modelos tradicionais e realizar uma predição satisfatória. Na maioria dos casos, a diferença entre o resultado do RADAR e do modelo centralizado esteve dentro do esperado, com uma vantagem relativa do modelo centralizado que pode ser explicada pelas limitações do modelo federado do RADAR. Ao longo dos experimentos foi possível observar também que a diferença do nível de fluxo da via se traduz em diferentes níveis de erro na predição. Ainda, diferentes níveis de fluxo de veículos implicam em diferentes quantidades de dados disponíveis para o treinamento do RADAR, alterando o seu desempenho quando comparado ao modelo centralizado. Nos casos em que foi observada uma acentuação da diferença entre o RADAR e o modelo centralizado, isso esteve relacionado com uma baixa quantidade de dados disponíveis para o treinamento ou, em um caso específico, uma combinação de clientes na rede federada gerando resultados discrepantes.

Na etapa envolvendo a variação do tipo de dia da semana, foi possível observar que os resultados foram relativamente mais próximos quando comparados ao modelo centralizado no caso do cenário de Meio de Semana. De fato, o RADAR chegou inclusive a ter um resultado levemente melhor para uma das métricas nesse cenário. Dado as características de um modelo FL, essa vantagem do RADAR pode ser vista como eficiente. Por outro lado, para o caso de Final de Semana, foi observada uma vantagem relativamente maior nos resultados do modelo centralizado. Desse modo, considerando a relação entre os tipos de dia e a densidade da via, os experimentos reforçam a sensibilidade do desempenho do RADAR com relação à quantidade de dados disponíveis para o treinamento. Dias de Final de Semana possuem um fluxo menor de veículos, o que implica em menos dados para o treinamento da rede. Reciprocamente, dias de Meio de Semana possuem um fluxo maior de veículos, o que implica em mais dados para o treinamento da rede. Isso é ainda amplificado pelo fato de existirem menos dias de Final de Semana do que dias de Meio de Semana, o que se converte em mais dados ainda para os dias de Meio de Semana. Assim, nos casos com mais dados o resultado do RADAR foi relativamente melhor, enquanto nos casos com menos dados o resultado foi relativamente pior.

# Capítulo 6

## Conclusão

Ao longo da elaboração desta pesquisa, foi possível compreender os avanços que os sistemas de predição de fluxo de tráfego podem trazer na impulsão de uma urbanização eficiente e sustentável. A expansão das malhas urbanas no mundo contemporâneo fez eclodir um ambiente no qual há uma grande gama de dispositivos capazes de auxiliar na resolução desse problema. Entretanto, a inclusão desses dispositivos em um único mecanismo é frequentemente limitada por fatores que impossibilitam o compartilhamento de dados entre eles, seja por questões técnicas ou burocráticas.

De modo a oferecer uma alternativa que contorne esses desafios, foi proposto o RADAR: um mecanismo federado para o problema de predição de fluxo de tráfego em vias urbanas. O RADAR funciona por meio de uma rede de sensores eletrônicos que captam seus próprios dados de tráfego e um servidor central que coordena cada sensor na realização do processo de *Federated Learning*. Os clientes, então, treinam modelos locais que são agregados em um modelo global pelo servidor central, sem que o compartilhamento dos dados locais seja necessário.

O RADAR foi validado em duas etapas. A primeira etapa consistiu de experimentos variando o número de sensores participando no mecanismo, de modo a encontrar a melhor configuração para a rede e aferir o número ótimo de clientes participantes do processo federado. Já a segunda etapa consistiu de experimentos variando o tipo de dia da semana, *i.e.* final de semana ou meio de semana, em que o RADAR é utilizado, de modo a avaliar o RADAR de acordo com a densidade de veículos na via. Como base de comparação, o desempenho do RADAR foi contrastado com modelos centralizados. Os resultados obtidos mostraram que o RADAR consegue ser competitivo com esses modelos tradicionais e realizar uma predição adequada do fluxo de veículos. Nos experimentos, os modelos centralizados apresentaram valores levemente melhores quanto às métricas de erro, o que pode ser explicado devido a limitações inerentes aos modelos *Federated Learning*. Além disso, foi detectado que o RADAR é particularmente sensível a quantidade de

dados disponíveis para o treinamento, obtendo resultados melhores na presença de uma quantidade maior de dados.

## 6.1 Trabalhos Futuros

A elaboração deste trabalho permitiu identificar evoluções que poderiam ser aplicadas em trabalhos futuros. Primeiramente, acredita-se que a utilização de uma base de dados maior e a realização de mais rodadas federadas poderia melhorar o desempenho do modelo proposto. Além disso, seria interessante trabalhar em uma personalização do modelo distribuído para cada cliente. Desse modo, os clientes continuariam aprendendo padrões uns com os outros, porém evitaria-se casos nos quais uma combinação de clientes gera resultados discrepantes dos demais, em razão das diferenças observadas nos seus padrões de fluxo. Também seria interessante tratar da inserção e remoção de clientes durante o processo federado, ao invés de ter um número fixo de clientes do início ao fim, trazendo mais dinamicidade e flexibilidade ao modelo. Outro fator que poderia ser melhorado está no desenvolvimento e na aplicação de métodos federados para o tratamento de atributos e hiper-parametrização do modelo. Devido ao campo de FL ser relativamente recente, ainda não há um corpo maduro de técnicas para lidar com esses pontos, o que os levou a serem considerados fora do escopo do trabalho. Finalmente, outras características de modelos FL poderiam ser mais exploradas no contexto da proposta, como questões relativas à privacidade dos dados e à segurança do modelo contra ataques.

# Bibliografia

- [1] Sara Paiva, Mohd Abdul Ahad, Gautami Tripathi, Noushaba Feroz e Gabriella Casalino. “Enabling technologies for urban smart mobility: Recent trends, opportunities and challenges”. Em: *Sensors* 21.6 (2021), p. 2143.
- [2] Zaheer Allam e Ayyoob Sharifi. “Research Structure and Trends of Smart Urban Mobility”. Em: *Smart Cities* 5.2 (2022), pp. 539–561.
- [3] Luke Butler, Tan Yigitcanlar e Alexander Paz. “Smart urban mobility innovations: A comprehensive review and evaluation”. Em: *Ieee Access* 8 (2020), pp. 196034–196049.
- [4] Arnav Choudhry. “Smart mobility: challenges and opportunities for the next generation of transportation”. Em: *XRDS: Crossroads, The ACM Magazine for Students* 28.3 (2022), pp. 14–19.
- [5] Attila M Nagy e Vilmos Simon. “Survey on traffic prediction in smart cities”. Em: *Pervasive and Mobile Computing* 50 (2018), pp. 148–163.
- [6] Elvira Ismagilova, Laurie Hughes, Yogesh K Dwivedi e K Ravi Raman. “Smart cities: Advances in research—An information systems perspective”. Em: *International Journal of Information Management* 47 (2019), pp. 88–100.
- [7] Zaib Ullah, Fadi Al-Turjman, Leonardo Mostarda e Roberto Gagliardi. “Applications of Artificial Intelligence and Machine learning in smart cities”. Em: *Computer Communications* 154 (2020), pp. 313–323. ISSN: 0140-3664. DOI: <https://doi.org/10.1016/j.comcom.2020.02.069>. URL: <https://www.sciencedirect.com/science/article/pii/S0140366419320821>.
- [8] Janetta Culita, Simona Iuliana Caramihai, Ioan Dumitrache, Mihnea Alexandru Moisescu e Ioan Stefan Sacala. “An hybrid approach for urban traffic prediction and control in smart cities”. Em: *Sensors* 20.24 (2020), p. 7209.
- [9] Thomas Liebig, Nico Piatkowski, Christian Bockermann e Katharina Morik. “Dynamic route planning with real-time traffic predictions”. Em: *Information Systems* 64 (2017), pp. 258–265.

- [10] Yan Kuang, Barbara TH Yen, Emiliya Suprun e Oz Sahin. “A soft traffic management approach for achieving environmentally sustainable and economically viable outcomes: An Australian case study”. Em: *Journal of environmental management* 237 (2019), pp. 379–386.
- [11] Arzoo Miglani e Neeraj Kumar. “Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges”. Em: *Vehicular Communications* 20 (2019), p. 100184.
- [12] Hoang Nguyen, Le-Minh Kieu, Tao Wen e Chen Cai. “Deep learning methods in transportation domain: a review”. Em: *IET Intelligent Transport Systems* 12.9 (2018), pp. 998–1004.
- [13] Yaguang Li e Cyrus Shahabi. “A brief overview of machine learning methods for short-term traffic forecasting and future directions”. Em: *Sigspatial Special* 10.1 (2018), pp. 3–9.
- [14] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson e Blaise Agueray Arcas. “Communication-efficient learning of deep networks from decentralized data”. Em: *Artificial intelligence and statistics*. PMLR. 2017, pp. 1273–1282.
- [15] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings et al. “Advances and open problems in federated learning”. Em: *Foundations and Trends® in Machine Learning* 14.1–2 (2021), pp. 1–210.
- [16] Tian Li, Anit Kumar Sahu, Ameet Talwalkar e Virginia Smith. “Federated learning: Challenges, methods, and future directions”. Em: *IEEE Signal Processing Magazine* 37.3 (2020), pp. 50–60.
- [17] Chen Zhang, Yu Xie, Hang Bai, Bin Yu, Weihong Li e Yuan Gao. “A survey on federated learning”. Em: *Knowledge-Based Systems* 216 (2021), p. 106775.
- [18] Mingqi Lv, Zhaoxiong Hong, Ling Chen, Tieming Chen, Tiantian Zhu e Shouling Ji. “Temporal multi-graph convolutional network for traffic flow prediction”. Em: *IEEE Transactions on Intelligent Transportation Systems* 22.6 (2020), pp. 3337–3348.
- [19] Ahmad Ali, Yanmin Zhu e Muhammad Zakarya. “Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction”. Em: *Neural Networks* 145 (2022), pp. 233–247. ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2021.10.021>. URL: <https://www.sciencedirect.com/science/article/pii/S0893608021004123>.

- [20] Haifeng Zheng, Feng Lin, Xinxin Feng e Youjia Chen. “A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction”. Em: *IEEE Transactions on Intelligent Transportation Systems* 22.11 (2020), pp. 6910–6920.
- [21] Cen Chen, Kenli Li, Sin G Teo, Xiaofeng Zou, Keqin Li e Zeng Zeng. “Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks”. Em: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14.4 (2020), pp. 1–23.
- [22] Xianglong Luo, Danyang Li, Yu Yang e Shengrui Zhang. “Spatiotemporal traffic flow prediction with KNN and LSTM”. Em: *Journal of Advanced Transportation* 2019 (2019).
- [23] Wentian Zhao, Yanyun Gao, Tingxiang Ji, Xili Wan, Feng Ye e Guangwei Bai. “Deep temporal convolutional networks for short-term traffic flow forecasting”. Em: *IEEE Access* 7 (2019), pp. 114496–114507.
- [24] Aniekan Essien, Ilias Petrounias, Pedro Sampaio e Sandra Sampaio. “A deep-learning model for urban traffic flow prediction with traffic events mined from twitter”. Em: *World Wide Web* 24.4 (2021), pp. 1345–1368.
- [25] Yi Liu, James J. Q. Yu, Jiawen Kang, Dusit Niyato e Shuyu Zhang. “Privacy-Preserving Traffic Flow Prediction: A Federated Learning Approach”. Em: *IEEE Internet of Things Journal* 7.8 (2020), pp. 7751–7763. DOI: 10.1109/JIOT.2020.2991401.
- [26] Yuanhang Qi, M Shamim Hossain, Jiangtian Nie e Xuandi Li. “Privacy-preserving blockchain-based federated learning for traffic flow prediction”. Em: *Future Generation Computer Systems* 117 (2021), pp. 328–337.
- [27] Matteus Silva, Luiz Bittencourt e Adín Rivera. “Towards Federated Learning in Edge Computing for Real-Time Traffic Estimation in Smart Cities”. Em: *Anais do IV Workshop de Computação Urbana*. Rio de Janeiro: SBC, 2020, pp. 166–177. DOI: 10.5753/courb.2020.12361. URL: <https://sol.sbc.org.br/index.php/courb/article/view/12361>.
- [28] Chenhan Zhang, Shuyu Zhang, James J. Q. Yu e Shui Yu. “FASTGNN: A Topological Information Protected Federated Learning Approach for Traffic Speed Forecasting”. Em: *IEEE Transactions on Industrial Informatics* 17.12 (2021), pp. 8464–8474. DOI: 10.1109/TII.2021.3055283.
- [29] Henry Lieu. “Traffic-flow theory”. Em: *Public Roads* 62.4 (1999).

- [30] Joost Verbraeken, Matthijs Wolting, Jonathan Katzy, Jeroen Kloppenburg, Tim Verbelen e Jan S Rellermeyer. “A survey on distributed machine learning”. Em: *Acm computing surveys (csur)* 53.2 (2020), pp. 1–33.
- [31] Iqbal H Sarker. “Machine learning: Algorithms, real-world applications and research directions”. Em: *SN Computer Science* 2.3 (2021), pp. 1–21.
- [32] Tom Mitchell, Bruce Buchanan, Gerald DeJong, Thomas Dietterich, Paul Rosenbloom e Alex Waibel. “Machine learning”. Em: *Annual review of computer science* 4.1 (1990), pp. 417–433.
- [33] A. L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. Em: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: 10.1147/rd.33.0210.
- [34] Thomas G Dietterich. “Machine learning”. Em: *Annual review of computer science* 4.1 (1990), pp. 255–306.
- [35] Ajay Shrestha e Ausif Mahmood. “Review of deep learning algorithms and architectures”. Em: *IEEE access* 7 (2019), pp. 53040–53065.
- [36] Yann LeCun, Yoshua Bengio e Geoffrey Hinton. “Deep learning”. Em: *nature* 521.7553 (2015), pp. 436–444.
- [37] Christian Janiesch, Patrick Zschech e Kai Heinrich. “Machine learning and deep learning”. Em: *Electronic Markets* 31.3 (2021), pp. 685–695.
- [38] Shi Dong, Ping Wang e Khushnood Abbas. “A survey on deep learning and its applications”. Em: *Computer Science Review* 40 (2021), p. 100379.
- [39] Alex Sherstinsky. “Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network”. Em: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306.
- [40] Sepp Hochreiter e Jürgen Schmidhuber. “Long Short-Term Memory”. Em: *Neural Comput.* 9.8 (nov. de 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [41] Greg Van Houdt, Carlos Mosquera e Gonzalo Nápoles. “A review on the long short-term memory model”. Em: *Artificial Intelligence Review* 53.8 (2020), pp. 5929–5955.
- [42] Ralf C Staudemeyer e Eric Rothstein Morris. “Understanding LSTM—a tutorial into long short-term memory recurrent neural networks”. Em: *arXiv preprint arXiv:1909.09586* (2019).

- [43] Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong e Dejing Dou. “From distributed machine learning to federated learning: A survey”. Em: *Knowledge and Information Systems* (2022), pp. 1–33.
- [44] Omar Abdel Wahab, Azzam Mourad, Hadi Otrok e Tarik Taleb. “Federated machine learning: Survey, multi-level classification, desirable criteria and future directions in communication and networking systems”. Em: *IEEE Communications Surveys & Tutorials* 23.2 (2021), pp. 1342–1397.
- [45] Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konečn, Stefano Mazzocchi, Brendan McMahan et al. “Towards federated learning at scale: System design”. Em: *Proceedings of Machine Learning and Systems* 1 (2019), pp. 374–388.
- [46] John Glen Wardrop. “Road paper. some theoretical aspects of road traffic research.” Em: *Proceedings of the institution of civil engineers* 1.3 (1952), pp. 325–362.
- [47] Ji Chu Jiang, Burak Kantarci, Sema Oktug e Tolga Soyata. “Federated Learning in Smart City Sensing: Challenges and Opportunities”. Em: *Sensors* 20.21 (2020). ISSN: 1424-8220. DOI: 10.3390/s20216230. URL: <https://www.mdpi.com/1424-8220/20/21/6230>.
- [48] Xavier Glorot e Yoshua Bengio. “Understanding the difficulty of training deep feed-forward neural networks”. Em: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. por Yee Whye Teh e Mike Titterton. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May de 2010, pp. 249–256. URL: <https://proceedings.mlr.press/v9/glorot10a.html>.
- [49] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson e Blaise Agüera y Arcas. “Communication-Efficient Learning of Deep Networks from Decentralized Data”. Em: (2016). DOI: 10.48550/ARXIV.1602.05629. URL: <https://arxiv.org/abs/1602.05629>.
- [50] Jason Brownlee. *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. Machine Learning Mastery, 2017.
- [51] Frank Hutter, Lars Kotthoff e Joaquin Vanschoren. *Automated machine learning: methods, systems, challenges*. Springer Nature, 2019.