Monitoramento remoto de áreas utilizando VANTs: uma abordagem baseada na Transferência do Aprendizado e Aprendizado Federado

Mizael Pereira Gonçalves[1], **Allan M. de Souza**[2], **José Rodrigues Torres Neto**[1]

[1]Departamento de Computação – Universidade Federal do Piauí (UFPI)
Teresina – PI – Brasil

[2]Instituto de Computação – Universidade Estadual de Campinas (UNICAMP)
Campinas – PI – Brasil

{mizaelgoncalves,jtorres}@ufpi.edu.br, allanms@ic.unicamp.br

Abstract. *Remote terrain monitoring using unmanned aerial vehicles (UAVs) has gained prominence in fields such as precision agriculture, environmental mapping, and urban management, enabling large-scale and real-time data collection. This work presents a comparative analysis between centralized and federated machine learning, leveraging transfer learning for the classification of aerial images captured by UAVs. The MobileNetV3Small architecture was employed as the foundation for developing a model capable of maximizing computational efficiency on edge devices with processing and energy constraints, while the EuroSAT_RGB dataset served as the evaluation basis. In the federated scenario, the DEEV (Devices, I Choose You) technique was employed to reduce the communication rate between devices and the central server, promoting greater bandwidth efficiency and data privacy. The results indicate that federated learning, combined with transfer learning, can achieve accuracy levels comparable to centralized learning, with significant advantages in terms of energy consumption and communication savings, demonstrating its feasibility for distributed monitoring applications.*

Resumo. *O monitoramento remoto de terrenos com veículos aéreos não tripulados (VANTs) tem ganhado destaque em áreas como agricultura de precisão, mapeamento ambiental e gestão urbana, permitindo a coleta de dados em grande escala e em tempo real. Este trabalho realiza uma análise comparativa entre o aprendizado de máquina centralizado e federado, utilizando aprendizado por transferência na tarefa de classificação de imagens aéreas capturadas por VANTs. A arquitetura MobileNetV3Small foi empregada como base para o desenvolvimento de um modelo capaz de maximizar a eficiência computacional em dispositivos de borda com limitações de processamento e energia, enquanto o dataset EuroSAT_RGB serviu como base de avaliação. No cenário federado, foi utilizada a técnica DEEV (Dispositivos, Eu Escolho Vocês) para reduzir a taxa de comunicação entre os dispositivos e o servidor central, promovendo maior economia de banda e preservação de privacidade. Os resultados indicam que o aprendizado federado, aliado ao transfer learning, pode atingir níveis de acurácia comparáveis ao aprendizado centralizado, com vantagens significativas em termos de consumo de energia e economia de comunicação, demonstrando sua viabilidade para aplicações de monitoramento distribuído.*

1. Introdução

Com o uso de Veículos Aéreos Não Tripulados (VANTs), grandes volumes de dados de imagem podem ser coletados em tempo real, possibilitando a análise de terrenos e a detecção de padrões, especialmente em locais de difícil acesso e onde o monitoramento contínuo é necessário Neto et al. (2017). No entanto, processar essas imagens de forma eficiente apresenta desafios relevantes, sobretudo quando há limitações de largura de banda na comunicação, questões de privacidade e restrições de recursos computacionais nos dispositivos de captura Gonçalves et al. (2024).

As redes neurais convolucionais (*Convolutional Neural Networks* - CNN) emergiram como a principal abordagem para tarefas de classificação e análise de imagens [Zou et al. 2015]. Contudo, o treinamento redes em larga escala nem sempre é viável por ser um processo computacionalmente custoso. Nesse contexto, o aprendizado por transferência (transfer learning) se apresenta como uma solução favorável, pois permite usar redes previamente treinadas em grandes bases de dados para acelerar o treinamento em novas tarefas, utilizando menos dados e demandando menor capacidade computacional [Bengio 2012].

Apesar da eficiência no treinamento, a transferência de dados brutos para um servidor central traz desafios quanto à privacidade dos dados e ao alto custo de comunicação. Neste sentido, o aprendizado federado (*Federated Learning* - FL) tem se destacado como uma alternativa promissora. O FL permite que o treinamento de modelos de aprendizado de máquina ocorra localmente nos dispositivos distribuídos, sem a necessidade de compartilhar os dados originais [McMahan et al. 2017]. Essa abordagem é particularmente relevante em cenários de monitoramento remoto com VANTs, onde os dados capturados podem ser sensíveis e a largura de banda de comunicação com o servidor central é limitada.

No entanto, o aprendizado federado enfrenta desafios críticos relacionados à comunicação entre dispositivos de borda e o servidor central. Diversas técnicas têm sido propostas para reduzir o número de rodadas de comunicação e o volume de parâmetros trocados na comunicação de clientes no Aprendizado Federado. Neste trabalho, propomos uma análise de desempenho do aprendizado federado aplicado à tarefa de classificação de imagens aéreas capturadas por VANTs, utilizando aprendizado por transferência com a arquitetura MobileNetV3Small.

2. Trabalhos Relacionados

Hu et al. (2015) destacaram que o treinamento de CNN pode ser desafiador e demorado, especialmente quando os conjuntos de dados são limitados. Como alternativa, o aprendizado por transferência permite que modelos pré-treinados em grandes *datasets* sejam aplicados a novas tarefas, transferindo o conhecimento acumulado para cenários com menos dados e reduzindo o tempo de treinamento. Em CNN, estudos como os de Yin et al. (2017) e Yosinski et al. (2014) evidenciaram que as camadas iniciais do modelo geralmente aprendem características genéricas, aplicáveis a uma ampla gama de imagens. Por outro lado, as camadas finais tendem a se especializar em padrões específicos do conjunto de dados utilizado durante o treinamento. Esse comportamento torna o aprendizado por transferência altamente eficaz, permitindo o reaproveitamento das camadas iniciais

pré-treinadas enquanto as camadas finais são ajustadas para atender aos requisitos de uma nova tarefa.

Em uma análise abrangente, Dastour e Hassan (2023) investigaram o desempenho de trinta e nove modelos de aprendizado por transferência profundo na classificação de LULC, incluindo arquiteturas como ResNet50, EfficientNetV2B0 e ResNet152, avaliadas com base em métricas de acurácia e f1-score. O modelo ResNet50 obteve a maior precisão no conjunto de teste, enquanto EfficientNetV2B0 apresentou um equilíbrio favorável entre acurácia e tempo de treinamento. Este estudo destaca o valor do aprendizado por transferência em cenários com dados limitados, permitindo que modelos aproveitem conhecimento prévio e melhorem a precisão na classificação de imagens de satélite.

Naushad, Kaur e Ghaderpour (2021) exploraram o uso do aprendizado por transferência na classificação de imagens de alta resolução do *dataset* EuroSAT, utilizando as arquiteturas VGG16 e Wide ResNet-50. Para otimizar os resultados, os autores empregaram técnicas como aumento de dados, ajuste da taxa de aprendizado e *early stopping*, abordando o problema da escassez de dados para treinamento. O modelo Wide ResNet-50 alcançou uma precisão de 99,17%, superando tanto o VGG16 quanto modelos anteriores, além de demonstrar maior eficiência computacional e menor tempo de treinamento por época. Essa pesquisa evidencia que arquiteturas mais profundas, quando adequadamente ajustadas, podem alcançar alta precisão e eficiência na classificação de imagens de LULC, especialmente com dados aumentados e técnicas de regularização.

O desafio de treinar redes neurais profundas com recursos computacionais limitados é outro ponto destacado por Bichri et al. (2023), que demonstraram como o aprendizado por transferência permite reutilizar pesos de modelos pré-treinados, como VGG19, ResNet50 e MobileNet V2, para classificar objetos específicos com menos dados de treino. O estudo evidenciou que, embora o VGG19 alcance a melhor precisão de 95%, o MobileNet V2 oferece uma alternativa mais eficiente em termos de tempo de execução, reforçando a eficácia do reuso de modelos em contextos com restrição de dados, fundamental para aplicações de monitoramento remoto.

Além dos ganhos de eficiência, outro aspecto fundamental no aprendizado por transferência é a seleção estratégica do *dataset* inicial para pré-treinamento. Hernandez-Sequeira et al. (2022) demonstraram que essa escolha pode afetar diretamente o desempenho em domínios específicos, como o sensoriamento remoto. Em seu estudo, avaliaram a performance do modelo ResNet50 pré-treinado em diferentes bases de dados, incluindo RESISC-45, ImageNet e BigEarthNet, ao classificar o EuroSAT. Observou-se que o modelo pré-treinado no RESISC-45 — um *dataset* de temática semelhante — alcançou a melhor precisão, com 97,23%, superando tanto o ImageNet (95,94%) quanto o BigEarthNet (95,93%). Esse resultado indica que o uso de bases de dados alinhadas ao contexto final do modelo favorece o aprendizado por transferência, proporcionando ganhos de precisão em comparação com conjuntos de dados mais gerais.

O presente trabalho, portanto, propõe uma abordagem que combina aprendizado por transferência e aprendizado federado colaborativo, aplicando a arquitetura compacta MobileNetV3Small para melhor adequação a dispositivos de baixa capacidade computacional e energética, como os VANTs, no monitoramento remoto de áreas. Além disso, integramos o algoritmo DEEV (Dispositivos, Eu Escolho Vocês) [Souza et al. 2023] para

otimizar a comunicação federada, reduzindo o consumo de banda e preservando a privacidade dos dados. Essa abordagem permite avaliar a viabilidade e as vantagens da combinação dessas técnicas para redes de monitoramento remoto distribuídas, com foco em dispositivos de recursos limitados.

3. Aprendizado Federado em Redes de Monitoramento Remoto

3.1. Redes de Monitoramento Remoto com VANTs

As redes de monitoramento remoto baseadas em VANTs são amplamente utilizadas para coletar informações geográficas e ambientais de áreas extensas ou de difícil acesso. Cada VANT é equipado com sensores, como câmeras e dispositivos de medição, que permitem a captura de dados relevantes para aplicações como agricultura de precisão, gestão de desastres, monitoramento ambiental e inspeção de infraestrutura. Esses dispositivos são programados para operar de forma autônoma ou semiautônoma, seguindo rotas pré-estabelecidas ou ajustando suas trajetórias com base em eventos específicos detectados [Osco et al. 2021].

A colaboração entre os VANTs é um aspecto central para a eficiência dessas redes. Em cenários operacionais, os dispositivos frequentemente compartilham informações uns com os outros para otimizar a cobertura da área monitorada. Conforme abordado por Bailon-Ruiz e Lacroix (2020), alguns VANTs possuem a capacidade de realizar processamento on-board, ou seja, de analisar e processar os dados coletados localmente, sem a necessidade de envio imediato para servidores distantes. Isso permite maior eficiência na tomada de decisões em tempo real, além de possibilitar ajustes nas operações localmente, reduzindo a dependência de conectividade constante e a necessidade de aguardar orientações de uma base central.

A Figura 1 ilustra a diferença entre o aprendizado centralizado e o aprendizado federado, aplicados ao monitoramento remoto com VANTs. No aprendizado centralizado (Figura 1a), os VANTs transmitem os dados coletados diretamente para um servidor central, que processa essas informações, realiza o treinamento do modelo de aprendizado de máquina e, posteriormente, distribui o modelo treinado para que os clientes (VANTs) o utilizem em inferências.

No aprendizado federado (Figura 1b), por outro lado, o processo é distribuído. O servidor central envia uma cópia inicial do modelo de aprendizado para os VANTs, que treinam o modelo localmente utilizando seus próprios dados, sem transferi-los ao servidor, garantindo a preservação da privacidade. Após o treinamento local, os dispositivos enviam os parâmetros atualizados do modelo para o servidor, que realiza a agregação desses parâmetros para atualizar o modelo global. Esse processo, conhecido como rodada de treinamento federado, é repetido iterativamente até que o modelo global atinja convergência ou um desempenho satisfatório.

Esse processo distribuído não apenas preserva a privacidade dos dados, mas também evidencia as vantagens do aprendizado federado em cenários como o monitoramento remoto. A redução na transferência de grandes volumes de dados torna-se especialmente relevante para VANTs que operam em regiões remotas ou enfrentam conectividade limitada. Contudo, esse cenário traz desafios significativos devido à heterogeneidade dos dados coletados pelos diferentes dispositivos, frequentemente distribuídos de forma não

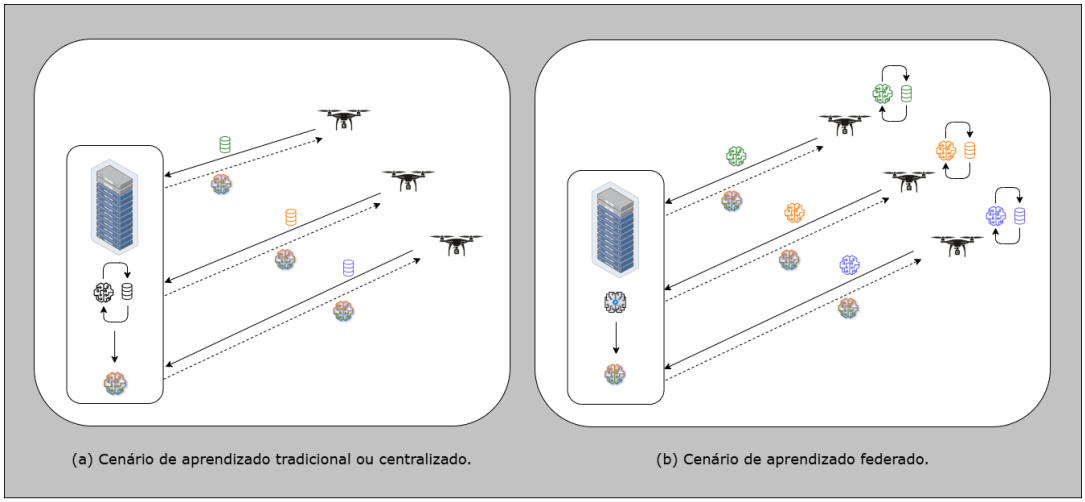

Figura 1. Cenários de treinamento. (a) Setas preenchidas indicam o envio de dados ao servidor, enquanto setas tracejadas representam o envio dos modelos treinados de volta aos clientes (VANTs). (b) Setas tracejadas mostram a inicialização dos pesos e a distribuição do modelo pelo servidor. Setas em loop representam o treinamento dos clientes com seus dados locais, e setas preenchidas indicam a devolução dos modelos ao servidor para agregação.

independente e não idêntica (non-IID). Essas diferenças podem comprometer a qualidade do modelo global, tornando indispensável o uso de técnicas de agregação robustas para assegurar uma convergência eficiente [McMahan et al. 2017].

Diante das limitações observadas em métodos tradicionais como o FedAvg(2017), surgiram técnicas mais avançadas que buscam otimizar a comunicação e a agregação, como o algoritmo DEEV (Dispositivos, Eu Escolho Vocês), proposto por Souza et al. (2023). O DEEV adota uma estratégia de seleção adaptativa, priorizando os clientes com menor desempenho local em cada rodada de treinamento. Em contraste com o FedAvg, que geralmente seleciona clientes de forma aleatória ou com base no tamanho do conjunto de dados de cada cliente, o DEEV identifica os clientes que mais necessitam de ajustes no modelo e os inclui na rodada de comunicação, enquanto aqueles com desempenho satisfatório são temporariamente excluídos das rodadas, reduzindo a sobrecarga na comnunicação entre clientes e servidor central.

4. Metodologia e Cenário Experimental

Nesta seção, apresentamos o conjunto de dados utilizado, as configurações do modelo base com aprendizado por transferência e os cenários de aprendizado centralizado e federado contruídos para comparar o desempenho em um ambiente de monitoramento remoto. Além disso, discutimos os detalhes da técnica de seleção adaptativa e as métricas de avaliação.

4.1. Conjunto de Dados

Os experimentos foram realizados com o *dataset* EuroSAT_RGB, composto por 27.000 imagens capturadas via satélite e rotuladas em dez categorias relacionadas ao uso e cobertura do solo: *Residential, Industrial, Highway, River, Forest, Pasture, Permanent Crop, Herbaceous Vegetation, Sea/Lake e Annual Crop*. Cada imagem original possui resolução

de 64x64 pixels e três canais RGB. Para adaptar as imagens ao formato de entrada (*input shape*) exigido pelo modelo base na transferência de aprendizado, foi realizado um pré-processamento que incluiu o redimensionamento para 224x224 pixels e a normalização dos valores de pixel, seguindo o padrão do modelo base utilizado.

4.2. Modelo Base e Aprendizado por Transferência

Para otimizar o treinamento do modelo e reduzir a demanda por grandes quantidades de dados, foi utilizada a técnica de aprendizado por transferência (transfer learning) com a arquitetura MobileNetV3Small, uma rede convolucional compacta e projetada para o uso em dispositivos de baixa capacidade computacional e restrições de energia, como VANTs [Howard et al. 2019]. Essa arquitetura utiliza blocos de convolução com profundidade separável e camadas de ativação com função *hard-swish*, que proporcionam um equilíbrio ideal entre precisão e eficiência computacional.

Tabela 1. Comparação entre o modelo base (MobileNetV3Small) e o novo modelo criado para classificação de uso e cobertura do solo.

Modelo	Total de Parâmetros	Treináveis	Não-Treináveis
MobileNetV3Small	441,000 (1.68 MB)	428,888 (1.64 MB)	12,112 (47.31 KB)
Novo Modelo	459,922 (1.75 MB)	18,858 (73.66 KB)	441,064 (1.68 MB)

Neste estudo, foi desenvolvido um novo modelo utilizando a rede MobileNetV3Small como base, previamente treinada no ImageNet, um extenso conjunto de dados amplamente utilizado no treinamento de CNN. A estratégia adotada foi congelar as camadas do modelo base, o que significa que os pesos dessas camadas não são atualizados durante o treinamento. Isso ajudou a preservar as características gerais aprendidas pela MobileNetV3Small, que são relevantes para a tarefa de classificação de imagens. Para adaptar o modelo à tarefa específica de classificação de imagens de uso e cobertura do solo no EuroSAT_RGB, foram anexadas camadas adicionais: uma camada de *pooling* global (*GlobalAveragePooling2D*), uma camada densa com 32 neurônios, uma camada de normalização (*BatchNormalization*), uma camada de *dropout* e, finalmente, uma camada densa com 10 neurônios para a classificação final. Como resultado, o modelo final apresenta 459.922 parâmetros, representando um aumento de 18.922 parâmetros em comparação com o modelo base.

4.3. Configuração dos Cenários Centralizado e Federado

Para avaliar o desempenho do aprendizado centralizado em comparação ao aprendizado federado, foram estabelecidos dois cenários experimentais:

- **Cenário Centralizado:** No cenário centralizado, todas as imagens do EuroSAT_RGB foram carregadas em um servidor central, onde o modelo adaptado foi treinado globalmente. Isso significa que o modelo teve acesso a todo o conjunto de dados desde o início, permitindo otimizar os parâmetros de forma contínua e uniforme. Essa abordagem centralizada serviu como base comparativa, principalmente para avaliar a acurácia e o tempo de treinamento.

- **Cenário Federado:** No cenário federado, o treinamento do modelo foi realizado de forma distribuída entre múltiplos clientes, simulando dispositivos de borda (como VANTs) que possuem acesso apenas a subsets específicos dos dados. Cada cliente treinou o modelo localmente em seus dados, mantendo-os armazenados localmente e transmitindo apenas as atualizações dos pesos para um servidor central, onde essas atualizações eram agregadas periodicamente. Essa configuração preserva a privacidade dos dados e reflete um ambiente realista de monitoramento remoto distribuído, onde os dispositivos capturam dados de diferentes regiões e condições de terreno.

Para otimizar a comunicação no cenário federado, foi adotado o algoritmo DEEV (Dispositivos, Eu Escolho Vocês), que emprega uma estratégia de seleção adaptativa baseada no desempenho local dos clientes. A solução FedAvg foi definida como *baseline*, com a premissa de que 100% dos clientes seriam selecionados em cada rodada de comunicação. No caso do DEEV, apenas os clientes com menor acurácia no treinamento local continuavam a participar, enquanto os demais aguardavam até que seus modelos se aproximassem, em termos de acurácia, do modelo global. Proposta por Souza et al. (2023), essa técnica visa reduzir o overhead de comunicação e aumentar a eficiência do treinamento distribuído, sem comprometer a qualidade final do modelo. Em nossos experimentos, o DEEV foi implementado sem o uso do fator de decaimento gradual de clientes, devido ao número reduzido de participantes. Embora essa simplificação tenha tornado o processo de seleção mais direto, a eficácia da técnica foi mantida, garantindo resultados consistentes.

Um ambiente baseado em contêineres (dockerizado) foi criado para simular o cenário federado, com o objetivo de isolar as aplicações servidor e cliente em ambientes distintos. Dessa forma, tanto o servidor quanto os clientes são implantados como contêineres. Um contêiner manager executa a aplicação do servidor, enquanto os contêineres workers executam uma ou mais instâncias da aplicação cliente.

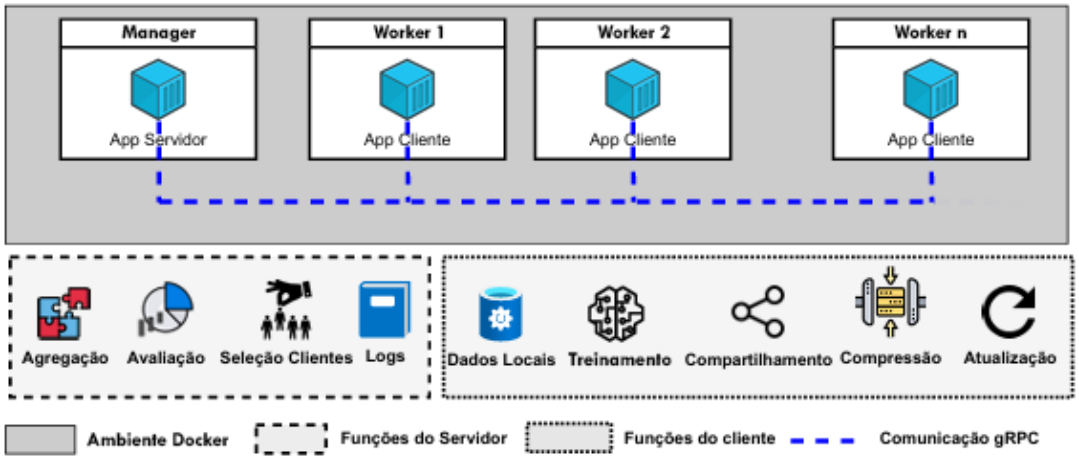

Figura 2. Ambiente dockerizado descrevendo a organização dos contêineres e as tarefas executadas pelo servidor e clientes no processo de aprendizado federado.

Como ilustrado, os contêineres clientes são responsáveis por armazenar dados locais, treinar o modelo localmente, compactar o modelo para melhorar a eficiência da comunicação, compartilhar o modelo treinado com o servidor para promover um aprendizado colaborativo e, finalmente, atualizar o modelo após a agregação realizada pelo servidor. A aplicação servidor, por outro lado, é responsável por agregar os modelos recebidos, avaliar o desempenho geral, selecionar um conjunto de clientes para a próxima rodada de treinamento e fornecer logs de comunicação, uso de recursos e métricas de desempenho. Esses logs são obtidos através da API do mecanismo Docker. A comunicação entre os clientes e o servidor é realizada via gRPC, utilizando o *framework* Flower [2].

O ambiente distribuído é configurado por meio de um arquivo docker-compose, permitindo a passagem de diferentes parâmetros para cada contêiner por meio de variáveis de ambiente. Esses parâmetros incluem a abordagem de agregação, o limiar de transmissão e a alocação de recursos, como limites de CPU e memória RAM. Dessa forma, é possível definir clientes heterogêneos ao personalizar as restrições de recursos para cada contêiner worker.

4.4. Configurações Experimentais e Hiperparâmetros

O experimento inicial foi realizado em um cenário centralizado como baseline, com o objetivo de estabelecer uma referência para comparação com o aprendizado federado. Para isso, foi utilizado um mecanismo de parada antecipada (*early stopping*) em conjunto com o *ModelCheckpoint*, que permitiram monitorar o desempenho do modelo e salvar automaticamente a melhor configuração obtida durante o treinamento. Essas estratégias ajudaram a prevenir o sobreajuste e a otimizar o tempo de execução. Diferentemente de abordagens tradicionais que utilizam um conjunto de teste independente, neste trabalho optou-se por utilizar apenas os conjuntos de treino e validação, divididos em 80% e 20%, respectivamente. Essa decisão foi motivada pela necessidade de manter consistência metodológica entre os cenários centralizado e federado, já que, neste último, é comum utilizar apenas o conjunto de validação para acompanhar o desempenho durante o treinamento.

Embora essa escolha limite a avaliação direta da generalização final do modelo em dados completamente inéditos, ela é suficiente para os objetivos do trabalho, que se concentram na comparação do desempenho relativo entre os dois cenários. No cenário centralizado, a acurácia obtida na validação, próxima de 95%, foi considerada como uma referência ou limite superior (“teto”) para avaliar a convergência no aprendizado federado. Assim, o foco está em analisar as métricas de acurácia e eficiência entre as abordagens, com a consciência de que os resultados refletem principalmente o comportamento durante o treinamento e validação.

Para ambos os cenários, os experimentos foram realizados utilizando o otimizador *Adam* com uma taxa de aprendizado $\mu = 0{,}0001$ e a função de perda *Sparse Categorical Crossentropy*, que é amplamente utilizada em tarefas de classificação com múltiplas classes. A taxa de aprendizado foi selecionada com base em testes preliminares para garantir uma convergência estável tanto no aprendizado centralizado quanto no federado.

No cenário federado, os dados foram distribuídos de maneira a criar um cenário não independente e não idêntico (non-IID), utilizando a distribuição *Dirichlet*. Essa abor-

[1] Imagem adaptada de Souza et al. (2023).

[2] Disponível em `https://flower.ai/`

dagem, frequentemente adotada na literatura, é utilizada para simular cenários de desbalanceamento típicos em dados de monitoramento remoto, onde cada cliente recebe amostras predominantemente de algumas classes, neste caso, de terreno específicas. Para garantir a reprodutibilidade das distribuições geradas, foram definidas *seeds* fixas durante o processo de particionamento dos dados.

Para controlar a variabilidade e o desbalanceamento entre as distribuições dos dados de cada cliente, a distribuição *Dirichlet* conta com o parâmetro *alpha* (α). Valores baixos de α (próximos de 0) indicam uma forte desigualdade na distribuição dos dados entre os clientes, ou seja, cada cliente receberá dados de um número muito restrito de classes, com pouca sobreposição. Valores médios de α (em torno de 1) resultam em distribuições mais equilibradas, com os clientes recebendo uma diversidade moderada de classes. Já valores altos de α (próximos ou acima de 10) indicam que as distribuições dos dados entre os clientes são mais semelhantes entre si, com mais classes compartilhadas e uma distribuição mais equilibrada de amostras. Essa estratégia de distribuição reflete a heterogeneidade característica de sistemas de monitoramento com VANTs, nos quais os dados coletados de diferentes regiões geográficas apresentam variações tanto na proporção das classes quanto nas características dos dados.

Os treinamentos federados foram configurados com 20 rodadas ($T = 20$) e tamanho do *batch* local $B = 32$. O número de épocas de treinamento local por cliente e o número de clientes K serão definidos como variáveis de teste e avaliados na seção de resultados. Para simular falhas de comunicação, foi definido um limiar de transmissão de 20%, indicando que cada cliente, com base em um número aleatório, possui 20% de probabilidade de não enviar suas atualizações de volta ao servidor após o treinamento local. Além disso, foi estabelecido um limite de CPU uniforme entre os clientes, de modo que clientes com o mesmo ID mantivessem as mesmas limitações de capacidade ao longo de todas as rodadas.

Após cada rodada de treinamento, a avaliação no cenário federado foi realizada em todos os clientes disponíveis, independentemente de terem participado do treinamento naquela rodada, contribuindo para uma análise abrangente do desempenho global do modelo. No treinamento centralizado, utilizou-se o mesmo tamanho de *batch* ($B = 32$), buscando manter consistência nas configurações.

Todos os experimentos foram realizados em um notebook equipado com processador Core i5-8250U 1.80GHz, 20 GB de memória RAM e placa gráfica NVIDIA GeForce MX150 com 2 GB de memória dedicada, que oferece suporte a CUDA para aceleração de treinamento. O arcabouço foi desenvolvido na linguagem Python, utilizando o *framework* Flower, previamente mencionado, em conjunto com a biblioteca TensorFlow[3]. O código está disponibilizado em um repositório[4] no GitHub, com o objetivo de assegurar a reprodutibilidade dos experimentos.

Para a análise comparativa entre os cenários, foram consideradas três métricas principais, utilizadas para avaliar tanto o desempenho do modelo quanto a eficiência do treinamento nos contextos centralizado e federado: Acurácia, Tempo de Treinamento, Custo de Comunicação.

[3]Disponível em `https://www.tensorflow.org/`

[4]Disponível em `https://github.com/mizaelgoonfs/Federated-Learning-Transfer`

5. Resultados

Nesta seção, apresentamos os resultados obtidos nos cenários de aprendizado centralizado e federado, utilizando o modelo de rede neural desenvolvido com transferência de aprendizado.

5.1. Cenário Centralizado

No treinamento centralizado, a solução foi executada em um ambiente virtual monolítico, no qual o modelo teve acesso a todo o conjunto de dados EuroSAT_RGB de forma integral. O treinamento foi configurado com lotes (*batches*) de tamanho fixo, e as condições de parada foram controladas utilizando *early stopping*, com uma paciência de 5 épocas, resultando em um tempo de treinamento total de aproximadamente 23 minutos. A Figura 3 apresenta as curvas de aprendizado e *loss* ao longo das épocas.

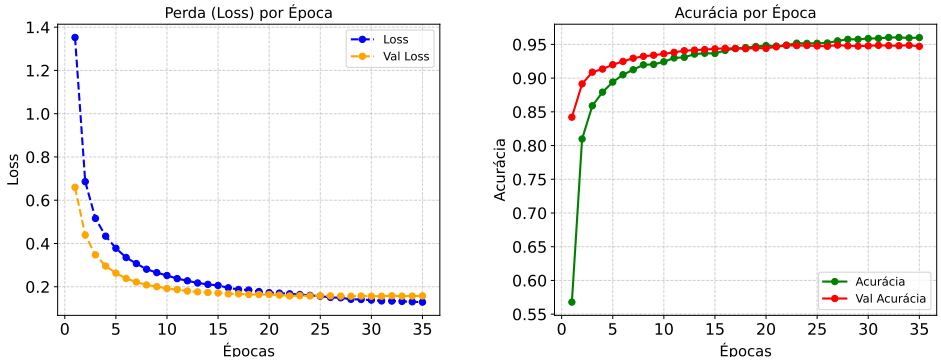

Figura 3. Curvas de aprendizado e *loss* ao longo das épocas, no cenário de aprendizado centralizado

Observa-se que os valores de acurácia já iniciam em níveis elevados, especialmente no conjunto de validação, evidenciando o impacto positivo da transferência de aprendizado. Além disso, o modelo apresentou aprendizado consistente e progressivo, atingindo estabilidade nos valores de *loss* e acurácia após cerca de 30 épocas. Esses resultados estabelecem a linha de base para comparação com os cenários federados.

5.2. Cenário Federado

No contexto de aprendizado federado, o conjunto de dados foi particionado em um cenário *non-IID*, utilizando uma distribuição de Dirichlet com parâmetro $\alpha = 1.0$, simulando um cenário moderadamente heterogêneo, no qual os dados de cada cliente apresentam similaridades limitadas. Dois algoritmos principais foram avaliados: FedAvg, amplamente reconhecido como método clássico no aprendizado federado, e DEEV, uma abordagem alternativa projetada para lidar com cenários desafiadores.

As simulações contemplaram diferentes configurações de número de clientes ($K = 4$ e $K = 8$) e épocas locais ($E = 1$ e $E = 4$). Para cada configuração, as curvas de aprendizado dos algoritmos foram registradas, com a linha de base da acurácia centralizada (95%) sendo exibida como referência nos gráficos apresentados na Figura 4. Além disso, foi monitorado o tempo total de execução em cada caso, conforme detalhado na Tabela 2.

Tabela 2. Tempo total de execução (em minutos) para diferentes configurações de clientes (K) e épocas locais (E), utilizando os algoritmos FedAvg e DEEV.

Algoritmo	K (Clientes)	E (Épocas Locais)	Tempo Total (minutos)
FedAvg	4	1	16.66
FedAvg	4	4	44.67
FedAvg	8	1	25.05
FedAvg	8	4	50.71
DEEV	4	1	14.90
DEEV	4	4	45.46
DEEV	8	1	17.44
DEEV	8	4	31.77

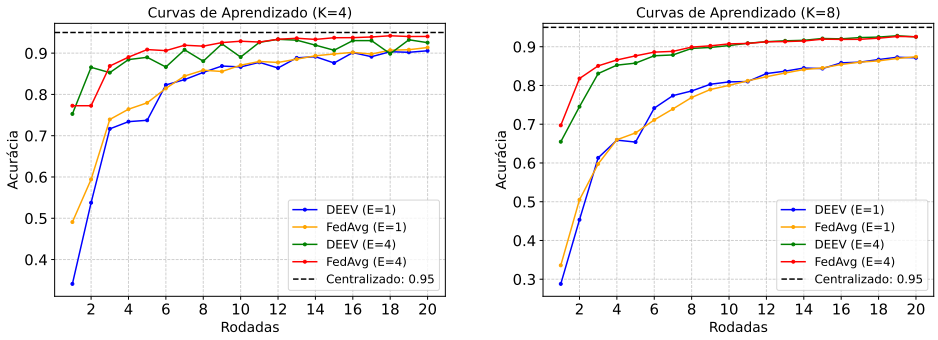

Figura 4. Convergência do modelo de aprendizado federado, validado no servidor de agregação para 4 e 8 clientes variando o número de épocas locais em 1 e 4.

No cenário com $K = 4$, os resultados indicam que o algoritmo FedAvg apresentou melhor desempenho inicial em $E = 1$, superando o DEEV nas primeiras rodadas de treinamento. Contudo, ambos os algoritmos convergem de maneira semelhante nas iterações subsequentes, aproximando-se da acurácia centralizada, especialmente na configuração $E = 4$. Esse comportamento deve-se ao maior refinamento permitido pelas épocas locais adicionais, que aceleram a convergência para uma acurácia elevada. Contudo, o DEEV apresentou maior variação nas curvas de aprendizado nessa configuração com $E = 4$, sugerindo maior sensibilidade à variação dos dados nesse cenário com menos clientes.

Entretanto, ao correlacionar esses resultados com o tempo de execução, nota-se que as estratégias consumiram tempos bastantes próximos. A pouca diferença temporal sugere que o custo computacional adicional de FedAvg é compensado por sua eficiência na agregação de pesos com poucos clientes, resultando em uma convergência ligeiramente mais rápida e com menor oscilação em relação ao DEEV.

Ao correlacionar com os tempos de execução, observa-se que, para $K = 4$, FedAvg consumiu 16,66 minutos em $E = 1$ e 44,67 minutos em $E = 4$, enquanto DEEV levou 14,90 e 45,46 minutos, respectivamente. A proximidade temporal entre as estratégias

pode ser atribuída à menor quantidade de clientes, que reduz tanto a heterogeneidade dos dados quanto o custo de agregação no servidor, limitando o impacto das diferenças entre os algoritmos. Além disso, o maior número de épocas locais ($E = 4$) amplifica o tempo de treinamento nos clientes, tornando o tempo de agregação uma parcela menos significativa no tempo total. Esses fatores contribuem para que o desempenho temporal dos algoritmos se mostre comparável nesse cenário.

Com $K = 8$, a maior heterogeneidade dos dados aumenta a complexidade do treinamento. Nesse contexto, FedAvg manteve superioridade inicial em $E = 4$, mas o DEEV, embora apresente um aprendizado inicial mais lento em $E = 1$, demonstra clara vantagem a partir da quinta rodada, alcançando uma curva de aprendizado mais aguda e convergindo de forma estável para valores próximos à acurácia centralizada. Quando analisamos os tempos de execução, DEEV com $E = 4$ consumiu 31,77 minutos, enquanto FedAvg levou 50,71 minutos, demonstrando uma clara vantagem de eficiência temporal do DEEV. Em $E = 1$, o tempo de DEEV foi de 17,44 minutos, ligeiramente menor que os 25,05 minutos de FedAvg.

Os resultados gerais destacam que o DEEV apresenta maior eficiência temporal em quase todas as configurações, mantendo curvas de aprendizado competitivas, especialmente em cenários com maior número de clientes ($K = 8$). Observa-se que, para configurações com uma única época local ($E = 1$), os tempos totais de execução das abordagens federadas se aproximam do tempo da abordagem centralizada (23 minutos), a qual processa todo o *dataset* em cada época. Essa proximidade sugere uma boa aplicação da distribuição de carga nos ambientes distribuídos, garantindo que o treinamento federado, mesmo em um cenário particionado, mantenha eficiência comparável ao treinamento centralizado.

A configuração com $E = 4$ mostrou-se superior em termos de convergência para ambos os algoritmos, embora implique maior custo computacional. No entanto, esse custo é justificado pela aproximação mais rápida ao baseline centralizado, reduzindo as rodadas necessárias para alcançar acurácias elevadas. Esses resultados demonstram a capacidade dos algoritmos de lidar com ambientes federados heterogêneos, com destaque para o DEEV, que traz equilíbrio entre desempenho e custo computacional de maneira eficiente.

5.3. Seleção de Clientes

A abordagem de seleção adaptativa de clientes no treinamento federado é fundamental para equilibrar a eficiência computacional e a eficácia do modelo. Nos experimentos apresentados, envolvendo 8 clientes e diferentes épocas locais (E=1 e E=4), os gráficos da Figura 4 demonstram como a quantidade de clientes selecionados por rodada influencia a acurácia agregada e sua interação com o número de épocas locais.

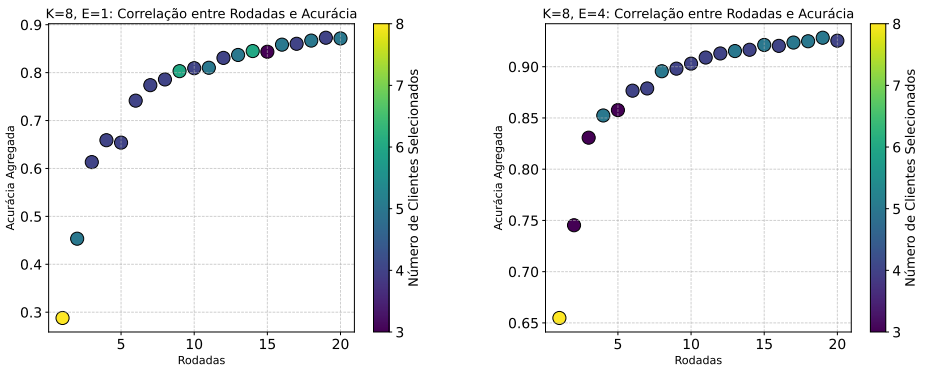

Figura 5. Análise do impacto da Seleção Adaptativa de Clientes na acurácia agregada: comparação entre 1 e 4 Épocas Locais (E=1 e E=4) com 8 Clientes (K=8)

Para $E = 1$, a seleção adaptativa resultou em maior variação na quantidade de clientes selecionados por rodada, o que ocasionou flutuações na acurácia agregada. Em contraste, com $E = 4$, a estabilidade na seleção de clientes promoveu uma curva de acurácia mais consistente. Destaca-se que, em $E = 4$, o DEEV selecionou menos clientes nas rodadas iniciais, mantendo, contudo, o ganho de aprendizado. Esse comportamento sugere que um maior número de épocas locais pode compensar uma menor participação de clientes, reduzindo o custo computacional sem comprometer, e até mesmo melhorando, a acurácia.

O FedAvg, configurado para selecionar 100% dos clientes em todas as rodadas, foi excluído desta comparação, pois não incorpora mecanismos de seleção adaptativa de clientes, o que é o foco central desta análise.

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou uma análise comparativa entre os paradigmas de aprendizado de máquina centralizado e federado aplicados à tarefa de classificação de imagens aéreas capturadas por VANTs. A utilização da arquitetura MobileNetV3Small, combinada com aprendizado por transferência, possibilitou o desenvolvimento de um modelo eficiente para dispositivos de borda com restrições de processamento e energia. No cenário federado, a técnica DEEV demonstrou-se eficaz na redução do custo de comunicação, promovendo economia de banda e preservação de privacidade, enquanto os resultados obtidos revelaram que o aprendizado federado, em geral, pode alcançar níveis de acurácia comparáveis ao aprendizado centralizado, mesmo em cenários de dados não independentes e não idênticos (non-IID).

Como trabalhos futuros, pretende-se explorar a heterogeneidade dos dispositivos e dos dados com outro dataset. Além disso, será feita uma análise de desempenho considerando outros pararâmetros de rede para avaliar a eficiência energética dos dispositivos durante a comunicação.

Referências

Bailon-Ruiz, R. and Lacroix, S. (2020). Wildfire remote sensing with uavs: A review from the autonomy point of view. pages 412–420.

Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. volume 27, pages 17–36. PMLR.

Bichri, H., Chergui, A., and Mustapha, H. (2023). Image classification with transfer learning using a custom dataset: Comparative study. *Procedia Computer Science*, 220:48–54.

Dastour, H. and Hassan, Q. K. (2023). A comparison of deep transfer learning methods for land use and land cover classification. *Sustainability*, 15.

Gonçalves, A., Castro, A., Evilly, B., Leão, E., Neto, J. T., Silva, R., Filho, A. C., and Rabelo, R. (2024). Simulador de aplicações de inteligência artificial das coisas para monitoramento em tempo real. In *Anais da XII Escola Regional de Computação do Ceará, Maranhão e Piauí*, pages 159–168, Porto Alegre, RS, Brasil. SBC.

Hernandez-Sequeira, I., Fernandez-Beltran, R., and Pla, F. (2022). Transfer deep learning for remote sensing datasets: A comparison study. pages 3207–3210.

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., and Adam, H. (2019). Searching for mobilenetv3. *CoRR*, abs/1905.02244.

Hu, F., Xia, G.-S., Hu, J., and Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7:14680–14707.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Arcas, B. A. y. (2017). Communication-efficient learning of deep networks from decentralized data. volume 54, pages 1273–1282. PMLR.

Naushad, R., Kaur, T., and Ghaderpour, E. (2021). Deep transfer learning for land use and land cover classification: A comparative study. *Sensors*, 21.

Neto, J. R., Boukerche, A., Yokoyama, R. S., Guidoni, D. L., Meneguette, R. I., Ueyama, J., and Villas, L. A. (2017). Performance evaluation of unmanned aerial vehicles in automatic power meter readings. *Ad Hoc Networks*, 60:11–25.

Osco, L. P., Junior, J. M., Ramos, A. P. M., de Castro Jorge, L. A., Fatholahi, S. N., de Andrade Silva, J., Matsubara, E. T., Pistori, H., Gonçalves, W. N., and Li, J. (2021). A review on deep learning in uav remote sensing. *International Journal of Applied Earth Observation and Geoinformation*, 102:102456.

Souza, A., Bittencourt, L., Cerqueira, E., Loureiro, A., and Villas, L. (2023). Dispositivos, eu escolho vocês: Seleção de clientes adaptativa para comunicação eficiente em aprendizado federado. pages 1–14. SBC.

Yin, X., Chen, W., Wu, X., and Yue, H. (2017). Fine-tuning and visualization of convolutional neural networks. pages 1310–1315.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. (2014). How transferable are features in deep neural networks?

Zou, Q., Ni, L., Zhang, T., and Wang, Q. (2015). Deep learning based feature selection for remote sensing scene classification. *IEEE Geoscience and Remote Sensing Letters*, 12:2321–2325.